extended_case_5

May 29, 2020

1 Generating useful features for further analysis on Amazon reviews

Business Context. You are a business consultant with new clients that are interested in analyzing reviews of their products on Amazon (as opposed to Yelp). They want to answer business questions like: "What are the most important factors driving negative reviews?", "Have there been any large changes to customer satisfaction/reviews over time?", etc.

Business Problem. Your main task is to **explore the given data and use the results of your investigation to engineer relevant features that could facilitate subsequent analysis and model-building**.

Analytical Context. The dataset provided is a large body of reviews related to movies and television left on Amazon between 1996 and 2014. When exploring our dataset, we will quickly encounter a familiar problem we discussed in the previous case: the word "good" is one of the most important words in both positive *and* negative reviews. Thus, we must develop methods to put "good" in the appropriate context.

1.1 Loading the data

We use a dataset of around 37,000 video reviews from Amazon Instant Video and 1,700,000 movie and TV reviews, all obtained from the website: http://jmcauley.ucsd.edu/data/amazon/. Note that there are much larger datasets available at the same site. We can expect better results on larger datasets (such as book reviews).

We begin by loading the dataset below:

[23]:
```
import nltk # imports the natural language toolkit
nltk.download('punkt')
import pandas as pd
import string
import gzip
import plotly

def parse(path):
  g = gzip.open(path, 'rb')
  for l in g:
    yield eval(l)
```

```
def getDF(path):
  i = 0
  df = {}
  for d in parse(path):
    df[i] = d
    i += 1
  return pd.DataFrame.from_dict(df, orient='index')

AIV = getDF('reviews_Amazon_Instant_Video_5.json.gz')
TVM = getDF('reviews_Movies_and_TV_5.json.gz')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\haris\AppData\Roaming\nltk_data…
[nltk_data]   Package punkt is already up-to-date!
```

[24]:
```
print(len(AIV))
print(AIV.head(10))
```

```
37126
       reviewerID        asin                    reviewerName   helpful  \
0  A11N155CW1UV02  B000H00VBQ                        AdrianaM    [0, 0]
1  A3BC8O2KCL29V2  B000H00VBQ                         Carol T    [0, 0]
2   A60D5HQFOTSOM  B000H00VBQ  Daniel Cooper "dancoopermedia"    [0, 1]
3  A1RJPIGRSNX4PW  B000H00VBQ                  J. Kaplan "JJ"    [0, 0]
4  A16XRPF40679KG  B000H00VBQ                   Michael Dobey    [1, 1]
5  A1POFVVXUZR3IQ  B000H00VBQ                         Z Hayes  [12, 12]
6  A1PG2VV4W1WRPL  B000H0X79O  Jimmy C. Saunders "Papa Smurf"    [0, 0]
7   ATASGS8HZHGIB  B000H0X79O                        JohnnyC    [0, 0]
8  A3RXD7Z44T9DHW  B000H0X79O                         Kansas    [0, 0]
9   AUX8EUBNTHIIU  B000H0X79O            Louis V. Borsellino    [0, 0]

                                          reviewText  overall  \
0  I had big expectations because I love English …      2.0
1  I highly recommend this series. It is a must f…      5.0
2  This one is a real snoozer. Don't believe anyt…      1.0
3  Mysteries are interesting.  The tension betwee…      4.0
4  This show always is excellent, as far as briti…      5.0
5  I discovered this series quite by accident. Ha…      5.0
6  It beats watching a blank screen. However, I j…      3.0
7  There are many episodes in this series, so I p…      3.0
8  This is the best of the best comedy Stand-up. …      5.0
9  Not bad.  Didn't know any of the comedians but…      3.0

                                    summary  unixReviewTime  \
0                 A little bit boring for me      1399075200
1                      Excellent Grown Up TV      1346630400
2                      Way too boring for me      1381881600
```

```
3                       Robson Green is mesmerizing      1383091200
4                     Robson green and great writing      1234310400
5  I purchased the series via streaming and loved…      1318291200
6                              It takes up your time.      1381795200
7             A reasonable way to kill a few minutes      1388275200
8                                          kansas001      1393372800
9                                Entertaining Comedy      1396396800

    reviewTime
0   05 3, 2014
1   09 3, 2012
2  10 16, 2013
3  10 30, 2013
4  02 11, 2009
5  10 11, 2011
6  10 15, 2013
7  12 29, 2013
8  02 26, 2014
9   04 2, 2014
```

[25]:
```
print(len(TVM))
print(TVM.head(10))
```

```
1697533
       reviewerID        asin                         reviewerName helpful  \
0   ADZPIG9QOCDG5  0005019281  Alice L. Larson "alice-loves-books"  [0, 0]
1  A35947ZP82G7JH  0005019281                         Amarah Strack  [0, 0]
2  A3UORV8A9D5L2E  0005019281                      Amazon Customer  [0, 0]
3  A1VKW06X1O2X7V  0005019281           Amazon Customer "Softmill"  [0, 0]
4  A3R27T4HADWFFJ  0005019281                                 BABE  [0, 0]
5  A2L0G56BNOTX6S  0005019281                    barbara whapeles  [0, 0]
6   A5NYUBEKXFLX5  0005019281                  B. Babb "kites0852"  [1, 1]
7  A2DJ8B8GE4V2VD  0005019281                       Berl S. Meyer  [0, 0]
8   AWF2S3UNW9UA0  0005019281                         beth holman  [0, 0]
9  A3O4UUT83DG3OU  0005019281                   Bettylou Sperling  [0, 0]

                                        reviewText  overall  \
0  This is a charming version of the classic Dick…      4.0
1  It was good but not as emotionally moving as t…      3.0
2  Don't get me wrong, Winkler is a wonderful cha…      3.0
3  Henry Winkler is very good in this twist on th…      5.0
4  This is one of the best Scrooge movies out.  H…      4.0
5  This has been a favorite movie of mine for a l…      5.0
6  This is the American adaptation of the Charles…      5.0
7  Glad that this american classic came out on dv…      5.0
8  A good Christmas carol dhenry winkler one duri…      5.0
9  How a bitter old man comes to know the true me…      5.0
```

```
                                              summary  unixReviewTime  \
0                           good version of a classic      1203984000
1                              Good but not as moving      1388361600
2               Winkler's Performance was ok at best!      1388361600
3       It's an enjoyable twist on the classic story      1202860800
4                                    Best Scrooge yet      1387670400
5                                    Dickens updated.      1383696000
6  A MUST-HAVE FOR ANY VIDEO CHRISTMAS COLLECTION!!      1230595200
7                          An American Christmas Carol      1260835200
8                          an american christmas carol      1386201600
9                                        Fantastic!      1379721600

    reviewTime
0  02 26, 2008
1  12 30, 2013
2  12 30, 2013
3  02 13, 2008
4  12 22, 2013
5   11 6, 2013
6  12 30, 2008
7  12 15, 2009
8   12 5, 2013
9  09 21, 2013
```

We notice that TVM is extremely long, and several columns seem uninteresting or hard to work with (e.g. `reviewerID`, `asin`, `reviewername`, `reviewtime`). We drop some information to make the following run more quickly:

[26]:
```
# TVM was annoyingly long while writing this - shorten it for now.
TVM = TVM.head(100000)
TVM = TVM.drop(columns = ['reviewerID', 'asin', 'reviewerName', 'reviewTime'])
AIV = AIV.drop(columns = ['reviewerID', 'asin', 'reviewerName', 'reviewTime'])
TVM.head(10)
```

[26]:
```
   helpful                                         reviewText  overall  \
0   [0, 0]  This is a charming version of the classic Dick…      4.0
1   [0, 0]  It was good but not as emotionally moving as t…      3.0
2   [0, 0]  Don't get me wrong, Winkler is a wonderful cha…      3.0
3   [0, 0]  Henry Winkler is very good in this twist on th…      5.0
4   [0, 0]  This is one of the best Scrooge movies out.  H…      4.0
5   [0, 0]  This has been a favorite movie of mine for a l…      5.0
6   [1, 1]  This is the American adaptation of the Charles…      5.0
7   [0, 0]  Glad that this american classic came out on dv…      5.0
8   [0, 0]  A good Christmas carol dhenry winkler one duri…      5.0
9   [0, 0]  How a bitter old man comes to know the true me…      5.0

                                              summary  unixReviewTime
0                           good version of a classic      1203984000
```

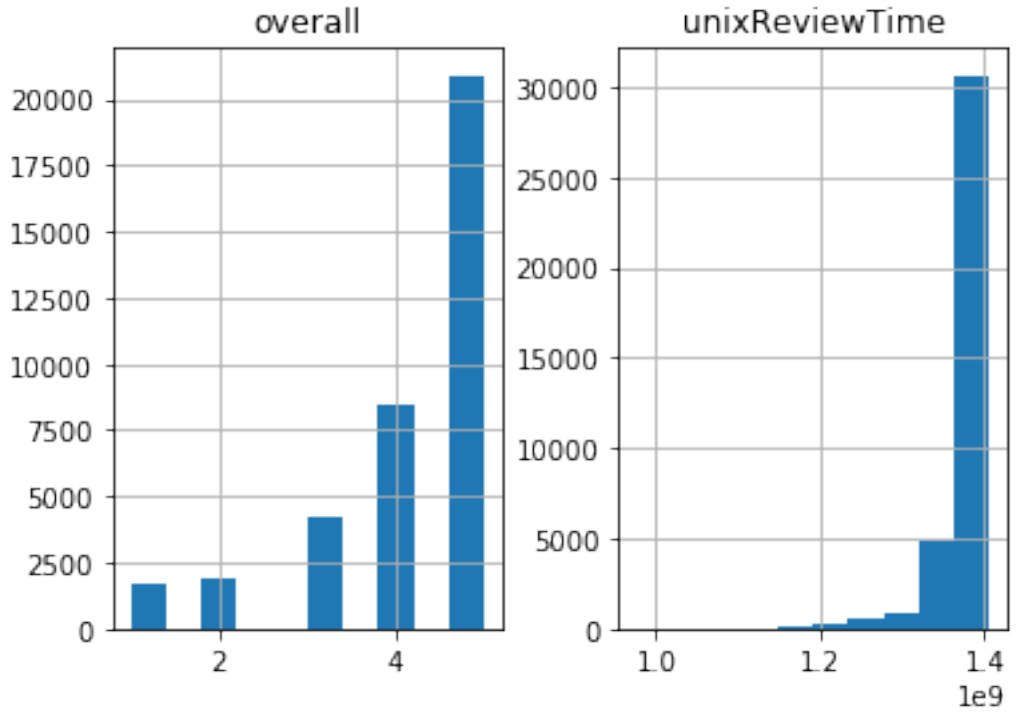
1	Good but not as moving	1388361600
2	Winkler's Performance was ok at best!	1388361600
3	It's an enjoyable twist on the classic story	1202860800
4	Best Scrooge yet	1387670400
5	Dickens updated.	1383696000
6	A MUST-HAVE FOR ANY VIDEO CHRISTMAS COLLECTION!!	1230595200
7	An American Christmas Carol	1260835200
8	an american christmas carol	1386201600
9	Fantastic!	1379721600

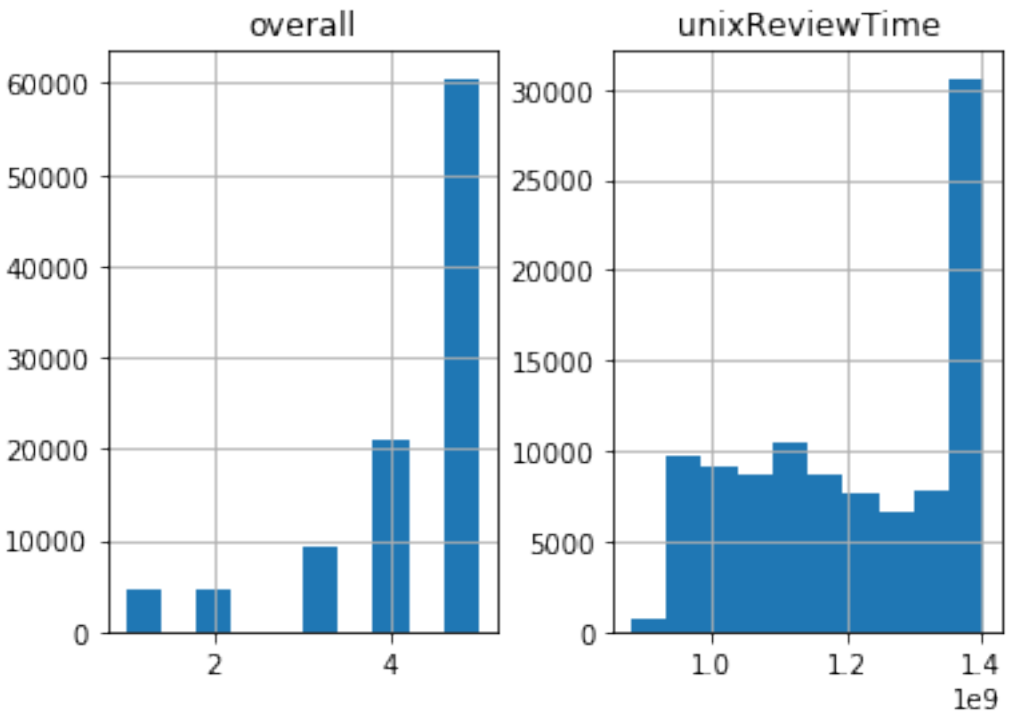
1.1.1 Exercise 1:

Plot histograms of all numeric quantities and compute pairwise correlations between them. Do you notice anything interesting?

Answer. One possible solution is shown below:

[27]:
```
# Histograms
AIV.hist();
TVM.hist();
```





We note that:

1. Most reviews are good - in fact, more than half are perfect in both datasets!
2. Almost all AIV reviews are extremely recent. It will be very difficult to detect trends without correcting for this. The TVM reviews are substantially more spread out (though they also include a recent spike).

[28]:
```
# All-pairs correlations
TVM.corr()
AIV.corr()
```

[28]:

	overall	unixReviewTime
overall	1.000000	0.009314
unixReviewTime	0.009314	1.000000

We observe that reviews become substantially shorter and more positive over time.

1.1.2 Exercise 2:

Find ten most frequently occuring non-stop words acrooss: (i) all reviews, (ii) positive reviews, (iii) negative reviews. Do the results surprise you? Why or why not?

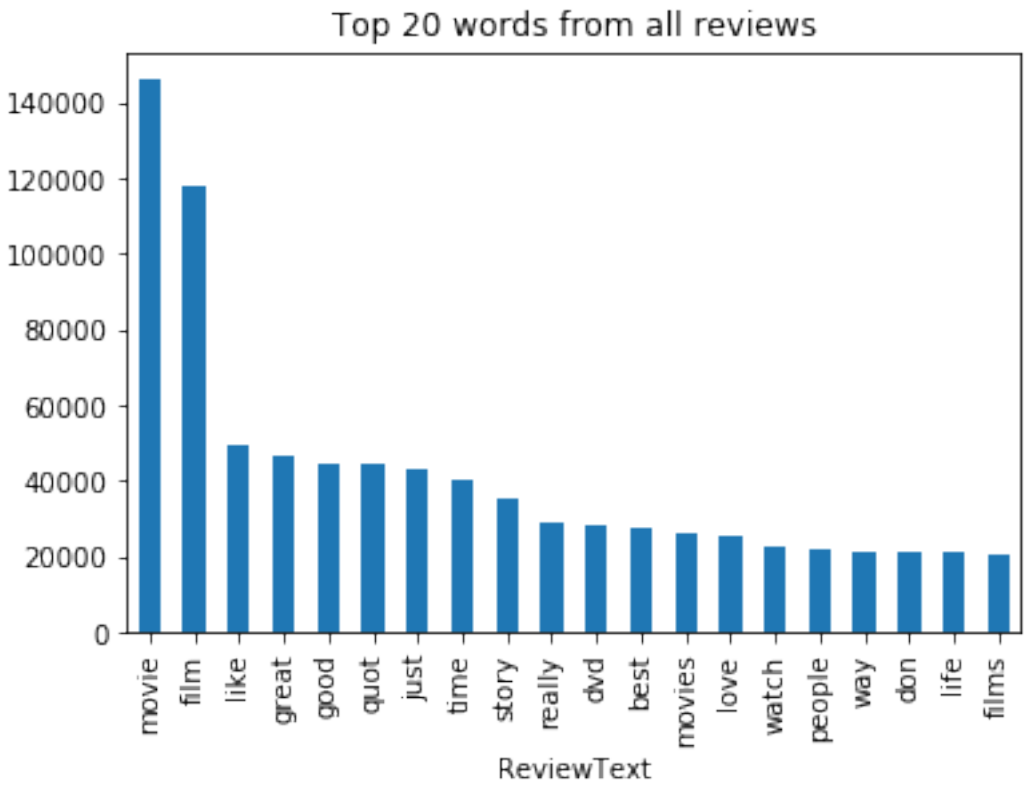
Answer. One possible solution is given below:

[29]:
```
from sklearn.feature_extraction.text import CountVectorizer

def get_top_n_words(corpus, n=1,k=1):
    vec = CountVectorizer(ngram_range=(k,k),stop_words = 'english').fit(corpus)
    bag_of_words = vec.transform(corpus)
    sum_words = bag_of_words.sum(axis=0)
    words_freq = [(word, sum_words[0, idx]) for word, idx in vec.vocabulary_.
 ↪items()]
    words_freq =sorted(words_freq, key = lambda x: x[1], reverse=True)
    return words_freq[:n]
```

[30]:
```
# All reviews
common_words = get_top_n_words(TVM['reviewText'], 20,1)
for word, freq in common_words:
    print(word, freq)

df = pd.DataFrame(common_words, columns = ['ReviewText' , 'count'])
df.groupby('ReviewText').sum()['count'].sort_values(ascending=False).plot(
    kind='bar',  title='Top 20 words from all reviews')
```

```
movie 146139
film 117896
like 49747
great 46391
good 44771
quot 44715
just 43336
time 39939
story 35359
really 29202
dvd 28395
best 27908
movies 25999
love 25779
watch 22436
people 21950
way 21529
don 21374
life 20965
films 20363
```

[30]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x185887a5f60>
```

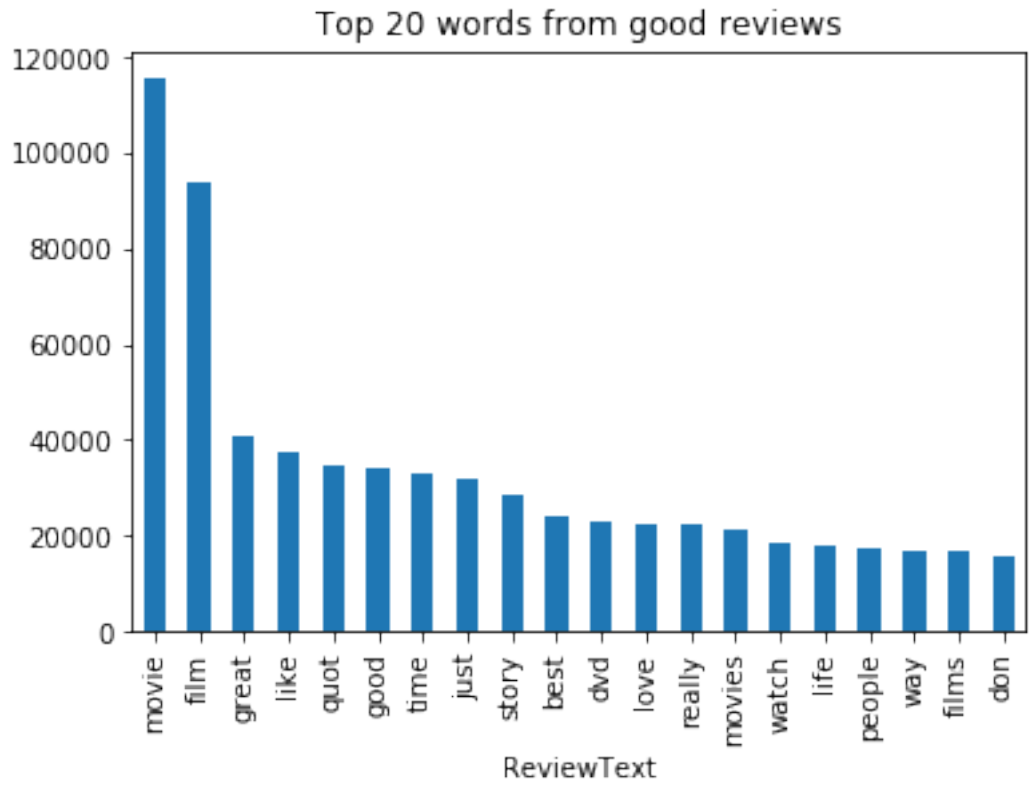


[31]:
```
# Positive reviews
GoodInd = TVM['overall'] >3.1
GoodRev = TVM[GoodInd]
BadInd = TVM['overall'] <2.1
BadRev = TVM[BadInd]

common_words = get_top_n_words(GoodRev['reviewText'], 20)
for word, freq in common_words:
    print(word, freq)
df = pd.DataFrame(common_words, columns = ['ReviewText' , 'count'])
df.groupby('ReviewText').sum()['count'].sort_values(ascending=False).plot(
    kind='bar', title='Top 20 words from good reviews')
```

```
movie 115580
film 93835
great 41013
like 37190
quot 34366
good 34320
time 32859
just 31716
story 28559
```

```
best 24257
dvd 23076
love 22405
really 22185
movies 21329
watch 18387
life 17972
people 17203
way 16891
films 16620
don 15604
```

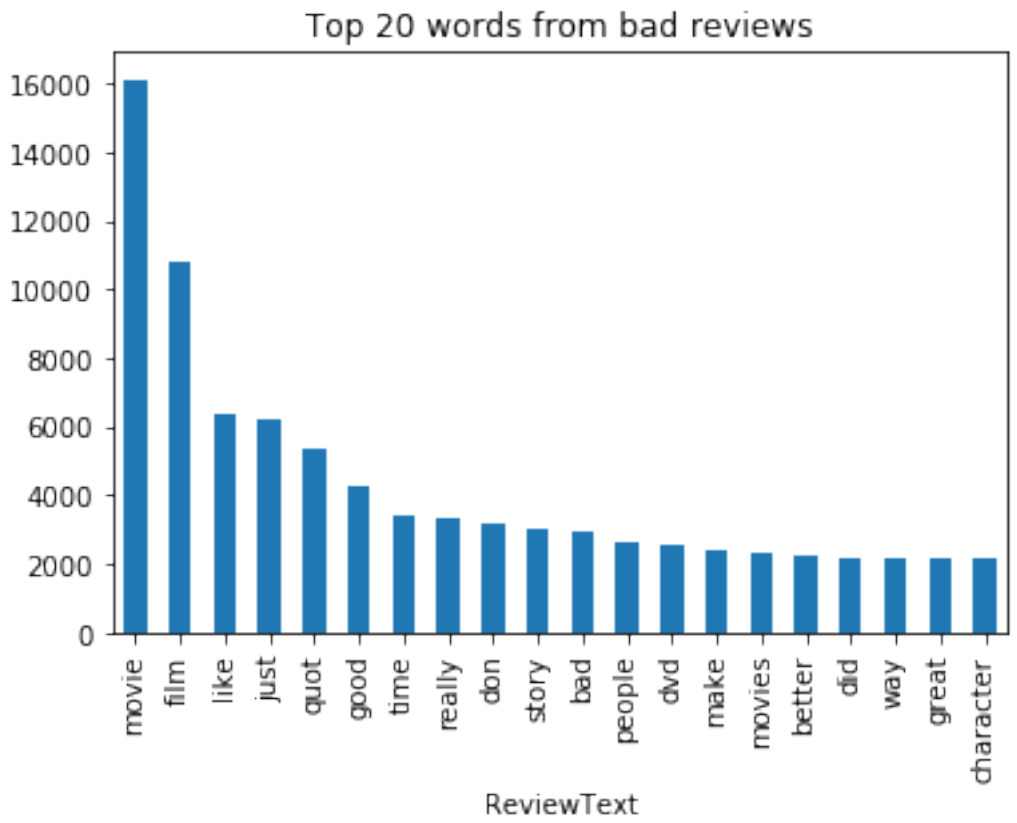
[31]: <matplotlib.axes._subplots.AxesSubplot at 0x18583a38940>



[32]:
```
# Negative reviews
common_words = get_top_n_words(BadRev['reviewText'], 20)
for word, freq in common_words:
    print(word, freq)
df = pd.DataFrame(common_words, columns = ['ReviewText' , 'count'])
df.groupby('ReviewText').sum()['count'].sort_values(ascending=False).plot(
    kind='bar', title='Top 20 words from bad reviews')
```

```
movie 16098
film 10802
like 6400
just 6185
quot 5348
good 4302
time 3401
really 3311
don 3164
story 3005
bad 2919
people 2603
dvd 2572
make 2366
movies 2350
better 2255
did 2202
way 2198
great 2185
character 2158
```

[32]: <matplotlib.axes._subplots.AxesSubplot at 0x185857b5cc0>



Note that the results are almost identical across all three groups of reviews: almost all of the most important words are very common generic words about movies (i.e. "movie" itself). As we saw in the previous case, this is not that surprising - given that we are dealing with data in a movie specific context, and stop words will only account for generic words that appear across many English contexts, the most common non-stop words will be basic vocabulary related to movies, not necessarily more precise words that give more color to the reviews themselves.

1.1.3 Exercise 3:

Manually inspect the first 10 negative reviews containing the word "good". What do you notice? How does this suggest we ought to proceed next?

Answer. One possible solution is given below:

[33]:
```
VBadInd = TVM['overall'] <1.1
VBadRev = TVM[VBadInd]

GoodWordBadRev = [rev for rev in VBadRev['reviewText'] if 'good' in rev]

print(len(GoodWordBadRev))
print(GoodWordBadRev[0:10])
```

```
1179
['The acting was good, the story was correct, but naked backsides was not
necessary.  Also the Potiphers wife did not need to be so very seductive of a
naked Joseph. Not a Christian movie.  I felt that even though it was probably
portrayed truthfully, Christans do not need to see so explicit a scene.  This
should be rated PG 13 at least.', 'The DVD transfer is pretty good but the scene
with Yukon discovering peppermint has been cut.It seems pretty odd to sell a DVD
as a restored classic only to leave out the main thing that was to be
restored!', 'why would mel gibson make a movie that would promote alot of
controversy? hmm, do you think maybe it might be because ALOT OF PEOPLE WOULD
WANT TO SEE IT THEN? i cant believe mel made this movie in honor of god, because
if he did then he would of at least made it about an interesting story with a
good plot and character development…i agree with the guy who said there should
be a sequel. but instead of making jesus a race car driver let him come back
from heaven decked out in all sorts of crazy gun crap like rambo. "Jesus is
back, and he is pissed", follow jesus as he goes back in time to save john
conner from the evil robots, and then when thats done jesus will master the
magic dragon and go save the elven maidens on mount gundor! we can make it a
coming of age story about how jesus is still learning about emotional
development or something stupid like that, and in the end can become a pirate
and sail into the sunset with a eye patch and pegged leg, or we could just
crucify him again, and make sure the nails stay in place', "Viewed: 5/14Rate:
15/14: The Passion of the Christ garnered a lot of attention in 2004, but I
didn't get caught up in the hype or follow it for the slightest bit. Fast
```

forward to ten years later, I was browsing through the library's shelf of movies that might strike my interest, and I saw The Passion of the Christ sitting there. So, I figured that I might give it a try. And I now wish that I never did and have regretted it. The thing with me, I don't watch movies made after the year of 2000 because almost everything I've seen thus far does not resemble good cinema in any shape, form, or manner. So, I gave up and declared that I'll only watch movies made before the end of the year 2000. There is a strong reason in my declaration, and The Passion of the Christ reaffirms my stance. The film brings out the worst of everything. It's the fake acting. It's the constant overt melodrama in everything that the characters do, feel, think, or speak. It's the free-all ultra-graphic torture. It's the unrealistic depiction of what a human being is capable of. It's the impossibility of physics. It's the unmerciful pacing that moves slow like molasses in January. It's the vapid storyline that seems to convey nothing of substance or significance. Truly, The Passion of the Christ is a waste of my time. By the way, I saw Braveheart and thought of it a terrific entertainment, and thenceforth, I was so annoyed by the many borrowed techniques from it into The Passion of the Christ. All in all, what I saw and felt is The Passion of the Torture on both sides: mine and the film's.", "The movie itself is beautifully filmed. The acting is quite good and as a history freak I found it interesting to hear Latin and Aramaic. I did not find it anti-semitic, but then I can understand why Jewish folk might be suspicious, given the beliefs espoused by Gibsons' family, and never actually disavowed by Gibson himself.I am not a Christian, so it struck me as odd that Christians would like this film from a theological standpoint. Perhaps I am just not getting it. I always understood that the most central point of the Jesus story was that he arose from the dead, not that he was tortured to death. So it suprises me that the film would concentrate almost entirely on this one aspect of the story, rather than on the resurrection part. That it does not, smacks of violence-as-a-marketing-tool to me. When all the hoopla has died down, it will be interesting to see how much money Gibson will have made off of the suffering of his savior.I have to say that I am more than a little shocked at the fawning tone I have seen in these reviews, particularly from Christians. The enthusiastic way in which they are taking joy in watching someone being slowly tortured to death is sickening; and exactly the same sort of bloodthirsty eagerness that the ancient Romans displayed when they watched animals and people (including Christians) being executed in the arena. The fact that there is a religious purpose to the film does not excuse this. I wonder if this movie were made with the exact same types of scenes, but without the religious content, how many of reviewers would still sing its praises. I guess Jesus' message has not really changed the human heart, eh? Shame on all of you.If you don't think about the movie's religious implcations and view it as a horror flick in period costume, the movie comes off just fine. However I suspect that this was not Gibson's intent.Not being a slasher movie fan, I probably won't see this movie again. Or even bother renting it, much less buying it. But if you are a modern day Savonarola, and believe that pain and suffering is the most important aspect of your religion, you will get off on this film.", "PATHETIC. SICK…truly SICK. This film has been wrongly categorized and is really nothing but a HORROR / SLASHER FILM, incorporating elements of the supernatural, but mostly just long

lasting sadistic torture…all somehow presumably presented for our entertainment. OH, and it's in LATIN. Yeah. Fun. If you love watching someone being tortured to death over a two hour period, then this film is for you.This film has nothing entertaining in it (I don't find torture entertaining) and provides nothing that makes us feel good or is uplifting or is of any benefit, except for maybe the some very few interesting costumes and some good visual effects, editing and camera work. This film does not teach any moral lesson or show any examples of doing good for others or how to better one's life or the lives of others. There are no spiritual lessons or examples or guides to spirituality. There is nothing that inspires any human to be a better human being.Mel gibson's use of eccentric and malformed persons laughing and heckling during one early scene somehow used to convey evil by their malformed appearance is particularly offensive to me as persons who in real life are malformed or suffer other natural disabilities are not intrinsically evil and such malformations are not caused out of some evil act. I expect the musical score was inspired, or more accurately copied, from Martin Scorsese's Last Temptation of Christ. There is nothing creative here.Mel Gibson seems to be a deeply troubled person with a particular penchant for rewriting history and attacking people, beliefs and ways of life that he finds weird or different or that he simply does not relate to. I wish he would have just continued acting and have kept his mouth shut; he would have been much easier to like.I find this film's level of outrage just below that of Pier Paolo Passolini's Salo; I own a copy of that film for some very specific reasons, mostly out of my high regard for free speech, rarely watch it due to its potency and perversity, and would never consider owning a copy of this pathetic piece of crap as it seems to have no redeeming value.I imagine Alex, the lead character in A Clockwork Orange, would likely find this entertaining…Alex in his natural, pre-corrected state.SICK SICK SICK …and just a HORROR FILM.", "I was really excited to see this movie but unfortuntly it was a big disappointment. I thought I was going to be watching a really spiritual, inspiring film but what I got instead was a big gorey bloodbath. I honestly don't understand how anyone could have a positive spiritual experience from watching this film. I found the violence to be extremely alienating and after awhile it even got a bit tedious. This is just like a big action movie, lots of blood and gore and people fighting and running around screaming. Mel seemed more interested with the violence and blood than he did with the message. Watching a person being tortured for over two hours isn't very inspiring, even if that person is Christ. The flashbacks were good and I'd rather have watched them than watched the overly violent scenes.To say some positive things about this movie, it is very well acted and the direction is just beautiful. Many of the scenes look like paintings. And I didn't see one speck of anti-semeticism in the movie. The worst characters in this film were the Roman soilders. Even though it was well acted and well directed I still don't like this movie.I don't understand why everybody is saying that this film will make people treat other people nicer after they've watched it. In this movie the violence and gore is considered more important than Jesus' teachings. We get a (very) little bit of the Sermon on the Mount about loving your enemy and that's about it.", "OK, I have to first admit that I'm anti - religious and I've always held the opinion that a religion-based nation is one that hasn't

```
realized that we're living in the 21st century and not the 16th.  But that
really didn't affect my reaction to this film, which is a brutal, trashy piece
of Christ-propaganda.  Christ was not the only person of that time period who
suffered through this much torture and brutality.  It was commonplace for people
who had differences with those in authority to be beaten and executed.  Mel
fails to explain what singled out Christ as a symbol of good because a)he is a
shallow and poor filmmaker and b)he doesn't care, he just wants an excuse to
show people lots of sickening violence to spark their guilt.  Don't allow
yourself to be roped in by this exploitive nonsense; Jesus Christ himself would
have taken offense to it.", 'I like blood and guts just as much as the next guy
(probably more so) but this film just has no entertainment (much less redeeming
value) whatsover. Based on what is assumed to be a true story to many (though
credible the evidence is certainly not) Mel Gibson basically beats the audience
over the head with excessive violence and gore in order to get them to embrace
his warped view of the world.So-called Chrisitans take note - you are endorsing
the same torture porn that you so readily abuse hollywood for (in such movies as
"Hostel" and "Saw 1-4). I believe last time I was unfortunate enough to hear a
sermon hypocrisy was still a sin.Cinematograpy-wise the film impresses, and you
have to be impressed by Jim Cavisiel\'s performance (or endurance really) but
this is a sick, twisted, nasty squalid little film that if was about Harry down
the street rather than Jebus the Christians would have condemned it
outright.That being said, an honest film made about god\'s killing and maiming
in the OT would also be just as repugnant - but at least it would make a more
interesting story than watching this guy getting beating (and no-one could
endure the pain).They say it\'s good to show the extent of the sacrifice that
Jesus apparantly made, but what\'s the big sacrifice anyway when he knew he\'d
be back new and improved in 3 days time anyway and would get to fly off to
heaven and come back a bit later to slaughter the rest of humanity (if they
don\'t believe).Yuck, left a foul taste in my mouth this one - really Mel I wish
you\'d made Lethal Weapon 5 instead.', 'Roger Ebert, whom I respect, gave this
film a good review when it was released, and I went to see it based on his
recomendation.Well, I\'ve disagreed with him before.Nothing in this movie
explains what Christ was about. Nothing about the Sermon on the Mount, nothing
about what he taught at all. It was all about how he died a gruesome death, an
obsession that Gibson puts above everything else.I noticed in the negative
reviews of "The Last Temptation of Christ" that many reviewers expressed
enthusiasm for Gibson\'s film, but hated Scorsese\'s, which I found a moving
examination of what it must have been like for Christ to struggle between the
human and divine sides of his nature. (An account based not on the Gospels, as
many believed, but upon a thoughtful writer\'s imagination.) How, after all, can
Christ be an inspiring example if he was never tempted by the things all men and
women are tempted by?The "Passion" is perverse.(Just returned to this review and
tried to take out all the extra spaces -- hope it worked.)']
```

We see that ”good” is used in three very different ways in negative reviews:

1. Sometimes the reviewer is pointing out something that is actually good, before saying that this doesn’t make up for more important problems.
2. Sometimes the reviewer is saying that something good is not present, as in ”almost everything

I've seen thus far does not resemble good cinema in any shape, form, or manner."
3. Occasionally, the word "good" is referring to morality.

How do these observations influence our future NLP efforts? Going in reverse order:

(3) Skimming through further reviews, we see that "good" is not used to refer to morality very often. This sense of the word is over-represented in our small sample above because "The Passion of the Christ" appears a few times amongst our first bad reviews. So we will put this off to the side in terms of guiding our next efforts for now.

(4) This is what we are most interested in - the word "good" used to reference *important* things that are not present. This is common throughout the dataset.

(5) This may be challenging for us to handle when we build models - it signals *unimportant* things that are good. Unfortunately, it is also quite common throughout the dataset!

These observations have a big impact on how we will process reviews. There are at least two distinct ways that negative reviews use the word "good" in a way that we care about, and a good analysis will not confuse them. There is no easy way to distinguish them by simple use of regular expressions. Instead, a more *global* analysis of the review is required. In future cases, you will learn about the concept of **polarity** which can assist with this.

1.1.4 Exercise 4:

Go through the list of bad reviews containing the word "good" that we found in the last question. For each review, extract the following:

1. The first word after "good"
2. The first word after "good" that is a noun or cardinal
3. The last word before "good" that is a noun or cardinal

Answer. One possible solution is shown below:

[34]:
```
# Answer to (i)
# We write a short function to extract the first word after "good" in a sentence

import re

def next_word(sentence):
    post = re.findall(r'good.*', sentence)
    if (len(post) > 0):
        temp = re.split(r'\s',post[0])
        if (len(temp) > 1):
            return(temp[1])
        else:
            return('')
    else:
        return('')

# We then use this to extract the relevant words
```

```
post_good = []
for sentence in GoodWordBadRev:
    temp = next_word(sentence)
    post_good.append(temp)

post_good = [i for i in post_good if i]
print(post_good[0:10])
```

```
['the', 'but', 'plot', 'cinema', 'and', 'or', 'and', 'because', 'to', 'review']
```

[36]:
```
# Answer to (ii)
# First, a function that extracts only "interesting" parts of speech
nltk.download('averaged_perceptron_tagger')
good_pos = ['CD','FW','NN','NNS','NNP','NNPS']

def ExtractInteresting(sentence, good):
    words = nltk.word_tokenize(sentence)
    interesting = [k for k,v in nltk.pos_tag(words) if v in good]
    return(interesting)

# We use this to define a new function to extract the next "interesting" word
def next_word2(sentence):
    post = re.findall(r'good.*', sentence)
    if (len(post) > 0):
        temp = ExtractInteresting(post[0],good_pos)
        if (len(temp) > 0):
            return(temp[0])
        else:
            return('')
    else:
        return('')

# Finally, we actually find our list
post_good2 = []

for sentence in GoodWordBadRev:
    temp = next_word2(sentence)
    post_good2.append(temp)

post_good2 = [i for i in post_good2 if i]
print(post_good2[0:10])
```

```
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\haris\AppData\Roaming\nltk_data…
[nltk_data]   Unzipping taggers\averaged_perceptron_tagger.zip.
```

```
['story', 'scene', 'plot', 'cinema', 'history', 'benefit', 'scenes.To',
'filmmaker', 'extent', 'review']
```

[37]:
```
# Answer to (iii)
# We define a function to extract all words *before* the first appearance of␣
 ↪"good"

def prev_word(sentence):
    post = re.findall(r'.*?good', sentence)
    if (len(post) > 0):
        temp = ExtractInteresting(post[0],good_pos)
        if (len(temp) > 0):
            return(temp[len(temp)-1])
        else:
            return('')
    else:
        return('')

# Finally, we actually find our list
pre_good = []

for sentence in GoodWordBadRev:
    temp = prev_word(sentence)
    pre_good.append(temp)

pre_good = [i for i in pre_good if i]
print(pre_good[0:10])
```

```
['acting', 'transfer', 'story', 'everything', 'acting', 'nothing', 'flashbacks',
'symbol', '.They', 'film']
```

1.1.5 Exercise 5:

We have seen that individual words are not always very informative. Look for the most informative bigrams and trigrams, in both, positive and negative reviews.

Answer. One possible solution is shown below:

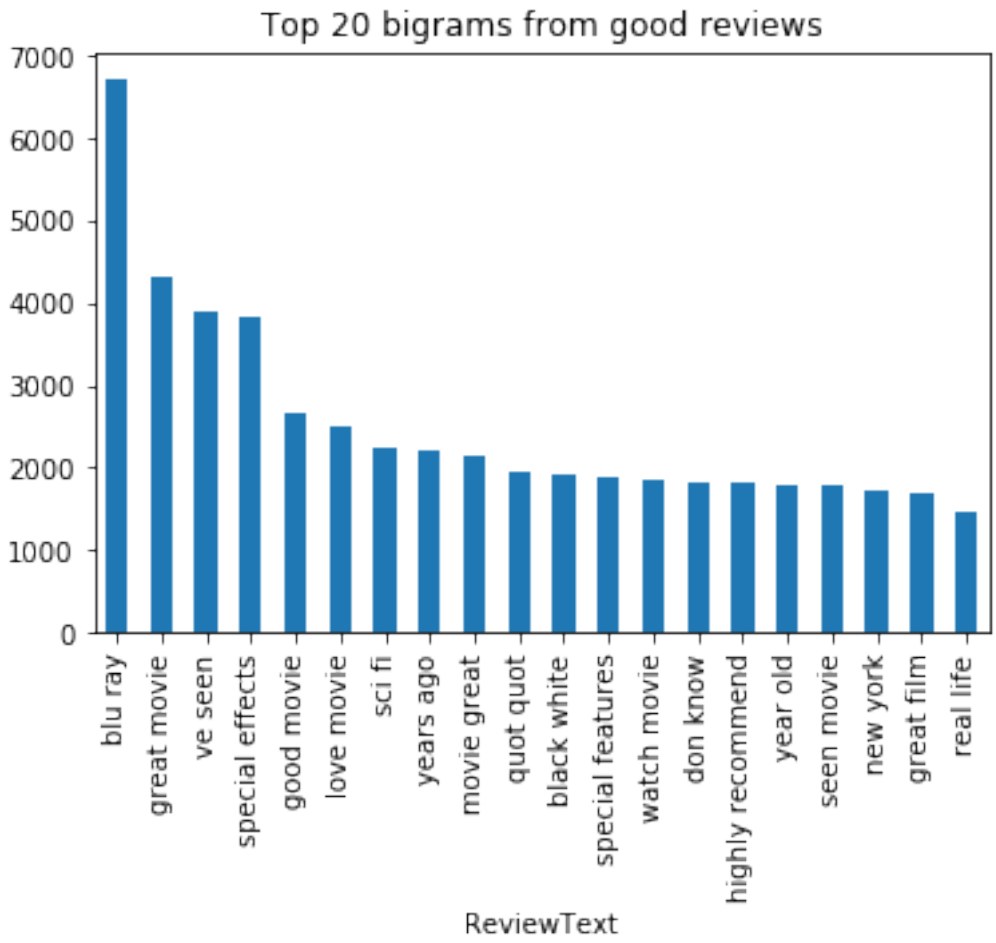
[38]:
```
# We recall the following code for extracting the top k n-grams from a corpus
def get_top_n_words(corpus, n=1,k=1):
    vec = CountVectorizer(ngram_range=(k,k),stop_words = 'english').fit(corpus)
    bag_of_words = vec.transform(corpus)
    sum_words = bag_of_words.sum(axis=0)
    words_freq = [(word, sum_words[0, idx]) for word, idx in vec.vocabulary_.
 ↪items()]
    words_freq =sorted(words_freq, key = lambda x: x[1], reverse=True)
    return words_freq[:n]
```

[39]:
```
# We apply the function "get_top_n_words"  to the lists "GoodRev" and "BadRev"
# First, bigrams from good reviews
```

```
common_words = get_top_n_words(GoodRev['reviewText'], 20,2)
for word, freq in common_words:
    print(word, freq)
df = pd.DataFrame(common_words, columns = ['ReviewText' , 'count'])
df.groupby('ReviewText').sum()['count'].sort_values(ascending=False).plot(
    kind='bar', title='Top 20 bigrams from good reviews')
```

```
blu ray 6714
great movie 4300
ve seen 3895
special effects 3830
good movie 2669
love movie 2509
sci fi 2246
years ago 2191
movie great 2152
quot quot 1935
black white 1927
special features 1874
watch movie 1853
don know 1824
highly recommend 1812
year old 1799
seen movie 1774
new york 1720
great film 1683
real life 1471
```

[39]: <matplotlib.axes._subplots.AxesSubplot at 0x185836e07b8>

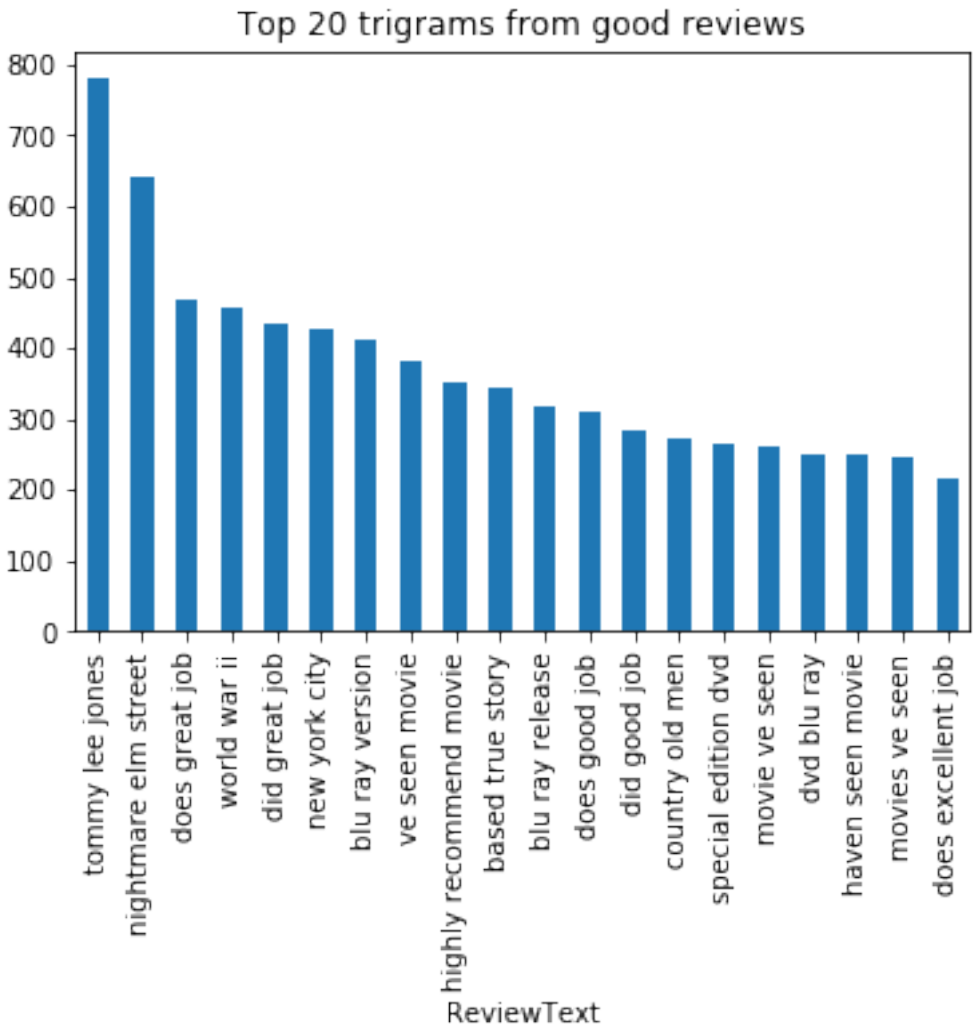


[40]:
```
# Next, trigrams from good reviews
common_words = get_top_n_words(GoodRev['reviewText'], 20,3)
for word, freq in common_words:
    print(word, freq)
df = pd.DataFrame(common_words, columns = ['ReviewText' , 'count'])
df.groupby('ReviewText').sum()['count'].sort_values(ascending=False).plot(
    kind='bar', title='Top 20 trigrams from good reviews')
```

```
tommy lee jones 779
nightmare elm street 640
does great job 467
world war ii 456
did great job 433
new york city 427
blu ray version 412
ve seen movie 383
highly recommend movie 350
based true story 344
```

```
blu ray release 316
does good job 310
did good job 283
country old men 273
special edition dvd 265
movie ve seen 262
dvd blu ray 251
haven seen movie 250
movies ve seen 247
does excellent job 216
```

[40]: <matplotlib.axes._subplots.AxesSubplot at 0x186589422e8>



1.1.6 Exercise 6:

Throughout the above search for informative words, we have seen that unigrams are not enough, but important words (such as "good") are not always next to the informative words that they describe. Devise a method to extract these informative words.

Answer. Important words like "good" (as well as those appearing in the other n-grams above) tend to be adjectives. Adjectives describe nouns, more specifically, almost always the noun following it that is closest to it in a sentence. This suggests that we can use POS tagging to extract the locations of the adjectives and nouns, then iterate through all adjectives and count forward from each until we reach the next noun. THese are our informative words.

1.1.7 Exercise 7:

Write a function(s) that transforms a sentence into a new text list by iteratively pairing each adjective in the sentence with the next noun that follows it in the sentence. For example, the text "That was a good, long movie" should return `["good movie", "long movie"]`.

[41]:
```
# We make a function that grabs the first adjective and its partner noun, then␣
 ↪returns (i) the pair and (ii) a shorter sentence.

def GrabFirstPair(sentence):
    words = nltk.word_tokenize(sentence)
    adjectives = [k for k,v in nltk.pos_tag(words) if v == 'JJ']
    if(len(adjectives) > 0):
        shorter = re.findall(adjectives[0] +'.*',sentence)[0]
        words2 = nltk.word_tokenize(shorter)
        nouns = [k for k,v in nltk.pos_tag(words) if v == 'NN']
        if(len(nouns) > 0):
            shorter = shorter.split(' ', 1)[1]
            return(adjectives[0], nouns[0],shorter)
    return('','','')

# We make a function that uses the above to iterate through a sentence
def GrabAllPairs(sentence):
    Pairs = []
    noun = ''
    adj =''
    while(len(sentence) > 0):
        adj,noun,sentence = GrabFirstPair(sentence)
        if(adj != ''):
            Pairs.append(adj + ' ' + noun)
    return(Pairs)

# We make a function that iterates through many sentences after tokenizing
def MakePairList(corpus):
    Sentences = nltk.sent_tokenize(BigRev)
```

```
    Pairs = []
    for sentence in Sentences:
        Pairs.extend(GrabAllPairs(sentence))
```

[ ]:
```
# We illustrate this
sentence = "The big black dog scared the red cat."
print(GrabFirstPair(sentence))
print(GrabAllPairs(sentence))
```