Louie Balderrama
Springboard Data Science Career Track, January 2019 cohort

Capstone Project II – Milestone Report 1

Problem Statement: Classifying Amazon reviews based on customer ratings using NLP

Impact

Reviews provide objective feedback to a product and are therefore inherently useful for consumers. These ratings are often summarized by a numerical rating, or the number of stars. Of course there is more value in the actual text itself than the quantified stars. And at times, the given rating does not truly convey the experience of the product – the heart of the feedback is actually in the text itself. The goal therefore is to build a classifier that would understand the essence of a piece of review and assign it the most appropriate rating based on the meaning of the text.

Background

Though product ratings on Amazon are aggregated from all the reviews by every customer, each individual rating is actually only an integer that ranges from one star to five stars. This reduces our predictions to discrete classes totaling five possibilities. Therefore what we'll have is a supervised, multi-class classifier with the actual review text as the core predictor.

This study is an exploration of Natural Language Processing (NLP). The goal of predicting the star rating given a piece of text will take on different NLP topics including word embedding, topic modeling, and dimension reduction. From there, we'll arrive at a final dataframe and we'll be employing different machine learning techniques in order to come up with the best approach (i.e. most accurate estimator) for our classifier.

Datasets

The **Amazon dataset** contains the customer reviews for all listed *Electronics* products spanning from May 1996 up to July 2014. There are a total of 1,689,188 reviews by a total of 192,403 customers on 63,001 unique products. The data dictionary is as follows:

- **asin** - Unique ID of the product being reviewed, *string*
- **helpful** - A list with two elements: the number of users that voted *helpful*, and the total number of users that voted on the review (including the *not helpful* votes), *list*
- **overall** - The reviewer's rating of the product, *int64*
- **reviewText** - The review text itself, *string*
- **reviewerID** - Unique ID of the reviewer, *string*
- **reviewerName** - Specified name of the reviewer, *string*
- **summary** - Headline summary of the review, *string*
- **unixReviewTime** - Unix Time of when the review was posted, *string*

Data Wrangling

The `df` is created from the Amazon dataset. If the file has been downloaded then the dataset is loaded from the local file. Otherwise the file is accessed and extracted directly from the repository.

	asin	helpful	overall	reviewText	reviewTime	reviewerID	reviewerName	summary	unixReviewTime
0	0528881469	[0, 0]	5	We got this GPS for my husband who is an (OTR)...	06 2, 2013	AO94DHGC771SJ	amazdnu	Gotta have GPS!	1370131200
1	0528881469	[12, 15]	1	I'm a professional OTR truck driver, and I bou...	11 25, 2010	AMO214LNFCEI4	Amazon Customer	Very Disappointed	1290643200
2	0528881469	[43, 45]	3	Well, what can I say. I've had this unit in m...	09 9, 2010	A3N7T0DY83Y4IG	C. A. Freeman	1st impression	1283990400
3	0528881469	[9, 10]	2	Not going to write a long review, even thought...	11 24, 2010	A1H8PY3QHMQQA0	Dave M. Shaw "mack dave"	Great grafics, POOR GPS	1290556800
4	0528881469	[0, 0]	1	I've had mine for a year and here's what we go...	09 29, 2011	A24EV6RXELQZ63	Wayne Smith	Major issues, only excuses for support	1317254400
5	0594451647	[3, 3]	5	I am using this with a Nook HD+. It works as d...	01 3, 2014	A2JXAZZI9PHK9Z	Billy G. Noland "Bill Noland"	HDMI Nook adapter cable	1388707200
6	0594451647	[0, 0]	2	The cable is very wobbly and sometimes disconn...	04 27, 2014	A2P5U7BDKKT7FW	Christian	Cheap proprietary scam	1398556800
7	0594451647	[0, 0]	5	This adaptor is real easy to setup and use rig...	05 4, 2014	AAZ084UMH8VZ2	D. L. Brown "A Knower Of Good Things"	A Perfdect Nook HD+ hook up	1399161600
8	0594451647	[0, 0]	4	This adapter easily connects my Nook HD 7"...	07 11, 2014	AEZ3CR6BKIROJ	Mark Dietter	A nice easy to use accessory.	1405036800
9	0594451647	[3, 3]	5	This product really works great but I found th...	01 20, 2014	A3BY5KCNQZXV5U	Matenai	This works great but read the details...	1390176000

Only the `overall` and the `unixReviewTime` series are stored as integers. The rest are interpreted as strings (objects).

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1689188 entries, 0 to 1689187
Data columns (total 9 columns):
asin              1689188 non-null object
helpful           1689188 non-null object
overall           1689188 non-null int64
reviewText        1689188 non-null object
reviewTime        1689188 non-null object
reviewerID        1689188 non-null object
reviewerName      1664458 non-null object
summary           1689188 non-null object
unixReviewTime    1689188 non-null int64
dtypes: int64(2), object(7)
memory usage: 116.0+ MB
```

The `unixReviewTime` is converted from Unix time to the more intuitive `datetime` datatype.

```
from datetime import datetime

condition = lambda row: datetime.fromtimestamp(row).strftime("%m-%d-%Y")
df["unixReviewTime"] = df["unixReviewTime"].apply(condition)
```

The `reviewTime` is dropped since the `unixReviewTime` series more accurately describes the time when each review was posted.

	asin	helpful	overall	reviewText	reviewerID	reviewerName	summary	unixReviewTime
0	0528881469	[0, 0]	5	We got this GPS for my husband who is an (OTR)...	AO94DHGC771SJ	amazdnu	Gotta have GPS!	06-01-2013
1	0528881469	[12, 15]	1	I'm a professional OTR truck driver, and I bou...	AMO214LNFCEI4	Amazon Customer	Very Disappointed	11-24-2010
2	0528881469	[43, 45]	3	Well, what can I say. I've had this unit in m...	A3N7T0DY83Y4IG	C. A. Freeman	1st impression	09-08-2010
3	0528881469	[9, 10]	2	Not going to write a long review, even thought...	A1H8PY3QHMQQA0	Dave M. Shaw "mack dave"	Great grafics, POOR GPS	11-23-2010
4	0528881469	[0, 0]	1	I've had mine for a year and here's what we go...	A24EV6RXELQZ63	Wayne Smith	Major issues, only excuses for support	09-28-2011

Each review is stored as string in the `reviewText` series. A sample product review is below:

```
I&#8217;m a big fan of the Brainwavz S1 (actually all of their headphones &#8211; have yet to be disappoin
ted with any of their products). The S1 has been my main set for active use (e.g., workouts, runs, etc.) s
ince the flat cable is very durable and resistant to tangles. The S5 keeps all the good features of the S1
and adds to it &#8211; the sound quality is richer and better defined.That&#8217;s not to say the S1 sound
s poor &#8211; they are quite good, in fact. But the S5 are better. The highs are better defined and the m
idrange has more punch to it. The bass comes through clearly without moving into the harsh territory when
the volume is pushed (as the S1s can do). The overall sound quality is very pleasing.The build quality see
ms solid &#8211; as solid as the S1 or better. I love the flat cable! I know that&#8217;s something that i
s not appreciated by everyone, but for me it&#8217;s been working out wonderfully. Although this (as most
other Brainwavz headsets) comes with an excellent hard shell case, I usually tote my earbuds wrapped aroun
d my mp3 player in my pocket. Easy to carry; very stressful on the cables and can lead to tangles with rou
nd wires. Flat wires, especially those with a thick jacket such as these, survive that abuse with zero pro
blems.The earbuds themselves are more sleekly shaped than the &#8220;can&#8221; style of the S1.Comfort is
in line with the customary Brainwavz style, which is to say it&#8217;s outstanding. It comes with a wide r
ange of tips to fit pretty much any ear, plus the Comply foam tips (which are my favorite). If fitted prop
erly you end up with zero ear irritation plus excellent sound isolation and bass response.These are an ove
r-the-ear design much like the S1. I never used that design prior to the S1 and it did take me a little ti
me to get accustomed to it. It became second nature quickly, and the design is a lot more stable when exer
cising than the conventional in-ear design.These are more expensive than the S1, but you can hear the diff
erence in price. Still, if you&#8217;re looking to keep the cost down a bit, the S1 is an excellent perfor
mer as well.Great sound, great comfort, wonderful cable design, and it comes with a solidly made case and
lots of eartips. Highly recommend.[Sample provided for review]
```

Each review is associated with a rating stored under the `overall` field. This serves as the quantified summary of a given review and will thus be used as the ground truth labels for the model.

NLP Pre-Processing

We'll work with `reviewText` to prepare our model's final dataframe. The goal is to produce tokens for every document (i.e. every review). These documents will make up our corpora where we'll draw our vocabulary from.

The following is a sample text in its original form. This is the same as what was inspected in the previous section.


```
I&#8217;m a big fan of the Brainwavz S1 (actually all of their headphones &#8211; have yet to be disappoin
ted with any of their products). The S1 has been my main set for active use (e.g., workouts, runs, etc.) s
ince the flat cable is very durable and resistant to tangles. The S5 keeps all the good features of the S1
and adds to it &#8211; the sound quality is richer and better defined.That&#8217;s not to say the S1 sound
s poor &#8211; they are quite good, in fact. But the S5 are better. The highs are better defined and the m
idrange has more punch to it. The bass comes through clearly without moving into the harsh territory when
the volume is pushed (as the S1s can do). The overall sound quality is very pleasing.The build quality see
ms solid &#8211; as solid as the S1 or better. I love the flat cable! I know that&#8217;s something that i
s not appreciated by everyone, but for me it&#8217;s been working out wonderfully. Although this (as most
other Brainwavz headsets) comes with an excellent hard shell case, I usually tote my earbuds wrapped aroun
d my mp3 player in my pocket. Easy to carry; very stressful on the cables and can lead to tangles with rou
nd wires. Flat wires, especially those with a thick jacket such as these, survive that abuse with zero pro
blems.The earbuds themselves are more sleekly shaped than the &#8220;can&#8221; style of the S1.Comfort is
in line with the customary Brainwavz style, which is to say it&#8217;s outstanding. It comes with a wide r
ange of tips to fit pretty much any ear, plus the Comply foam tips (which are my favorite). If fitted prop
erly you end up with zero ear irritation plus excellent sound isolation and bass response.These are an ove
r-the-ear design much like the S1. I never used that design prior to the S1 and it did take me a little ti
me to get accustomed to it. It became second nature quickly, and the design is a lot more stable when exer
cising than the conventional in-ear design.These are more expensive than the S1, but you can hear the diff
erence in price. Still, if you&#8217;re looking to keep the cost down a bit, the S1 is an excellent perfor
mer as well.Great sound, great comfort, wonderful cable design, and it comes with a solidly made case and
lots of eartips. Highly recommend.[Sample provided for review]
```

HTML Entities

Some special characters like the apostrophe (’) and the en dash (–) are expressed as a set of numbers prefixed by `&#` and suffixed by `;`. This is because the dataset was scraped from an HTML parser, and the dataset itself includes data that predated the universal UTF-8 standard.

These *HTML Entities* can be decoded by importing the `html` library.

```
I’m a big fan of the Brainwavz S1 (actually all of their headphones – have yet to be disappointed with any
of their products). The S1 has been my main set for active use (e.g., workouts, runs, etc.) since the flat
cable is very durable and resistant to tangles. The S5 keeps all the good features of the S1 and adds to i
t – the sound quality is richer and better defined.That’s not to say the S1 sounds poor – they are quite g
ood, in fact. But the S5 are better. The highs are better defined and the midrange has more punch to it. T
he bass comes through clearly without moving into the harsh territory when the volume is pushed (as the S1
s can do). The overall sound quality is very pleasing.The build quality seems solid – as solid as the S1 o
r better. I love the flat cable! I know that’s something that is not appreciated by everyone, but for me i
t’s been working out wonderfully. Although this (as most other Brainwavz headsets) comes with an excellent
hard shell case, I usually tote my earbuds wrapped around my mp3 player in my pocket. Easy to carry; very
stressful on the cables and can lead to tangles with round wires. Flat wires, especially those with a thic
k jacket such as these, survive that abuse with zero problems.The earbuds themselves are more sleekly shap
ed than the “can” style of the S1.Comfort is in line with the customary Brainwavz style, which is to say i
t’s outstanding. It comes with a wide range of tips to fit pretty much any ear, plus the Comply foam tips
(which are my favorite). If fitted properly you end up with zero ear irritation plus excellent sound isola
tion and bass response.These are an over-the-ear design much like the S1. I never used that design prior t
o the S1 and it did take me a little time to get accustomed to it. It became second nature quickly, and th
e design is a lot more stable when exercising than the conventional in-ear design.These are more expensive
than the S1, but you can hear the difference in price. Still, if you’re looking to keep the cost down a bi
t, the S1 is an excellent performer as well.Great sound, great comfort, wonderful cable design, and it com
es with a solidly made case and lots of eartips. Highly recommend.[Sample provided for review]
```

Since punctuation marks do not add value in the way we'll perform NLP, all the HTML entities in the review texts can be dropped. The output series `preprocessed` is our `reviewText` but without the special characters.

```
pattern = r"\&\#[0-9]+\;"

df["preprocessed"] = df["reviewText"].str.replace(pat=pattern, repl="", regex=True)

print(df["preprocessed"].iloc[1689185])
```

```
Im a big fan of the Brainwavz S1 (actually all of their headphones  have yet to be disappointed with any o
f their products). The S1 has been my main set for active use (e.g., workouts, runs, etc.) since the flat
cable is very durable and resistant to tangles. The S5 keeps all the good features of the S1 and adds to i
t  the sound quality is richer and better defined.Thats not to say the S1 sounds poor  they are quite goo
d, in fact. But the S5 are better. The highs are better defined and the midrange has more punch to it. The
bass comes through clearly without moving into the harsh territory when the volume is pushed (as the S1s c
an do). The overall sound quality is very pleasing.The build quality seems solid  as solid as the S1 or be
tter. I love the flat cable! I know thats something that is not appreciated by everyone, but for me its be
en working out wonderfully. Although this (as most other Brainwavz headsets) comes with an excellent hard
shell case, I usually tote my earbuds wrapped around my mp3 player in my pocket. Easy to carry; very stres
sful on the cables and can lead to tangles with round wires. Flat wires, especially those with a thick jac
ket such as these, survive that abuse with zero problems.The earbuds themselves are more sleekly shaped th
an the can style of the S1.Comfort is in line with the customary Brainwavz style, which is to say its outs
tanding. It comes with a wide range of tips to fit pretty much any ear, plus the Comply foam tips (which a
re my favorite). If fitted properly you end up with zero ear irritation plus excellent sound isolation and
bass response.These are an over-the-ear design much like the S1. I never used that design prior to the S1
and it did take me a little time to get accustomed to it. It became second nature quickly, and the design
is a lot more stable when exercising than the conventional in-ear design.These are more expensive than the
S1, but you can hear the difference in price. Still, if youre looking to keep the cost down a bit, the S1
is an excellent performer as well.Great sound, great comfort, wonderful cable design, and it comes with a
solidly made case and lots of eartips. Highly recommend.[Sample provided for review]
```

Extracting the Root Word

How often a word is used is key information in natural language processing. It is therefore important to reduce words to their root form. An example would be the usage of the word "*learn*". If we differentiate this base form from a modified version like "*learning*" then we might lose relational context between two documents that have used either word.

We'll be using Lemmatization to reduce tokens to their base word. This technique takes into account context similarity according to part-of-speech anatomy. Stemming is another common approach, although stemming only performs truncation and would not be able to reduce "*taught*" to "*teach*".

We will be using the *WordNetLemmatizer* from the Natural Language Toolkit (or *NLTK*). Lemmatization only applies to each word but it is dependent on sentence structure to understand context. We therefore need to have part-of-speech tags associated with each word. Our output is derived from applying the `lemmatize_doc` function to our `preprocessed` column.

The lemmatize_doc works as follows:

- Each review is broken down into a list of sentences
- Punctuations that only group words or separate sentences (hyphens therefore are excluded) are removed (replaced by whitespace) using RegEx
- Every sentence is further broken down into words (tokens)

Each of the sentences then becomes an ordered bag of words. Every word is then *tagged* to a part-of-speech. This word-tag tuple pair is then fed one at a time to the `lemmatize_word` function, which works as follows:

- Only modifiable words – nouns, verbs, adjectives, and adverbs – can be reduced to roots
- These words are lemmatized and appended to the `root` list
- Words that are not modifiable are added as they are to the `root` list

The output lists are linked together as a string using whitespace. In the end, each `preprocessed` review will retain its text form but with each word simplified as much as possible.

```python
import re
import nltk

from nltk import word_tokenize, pos_tag
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import sent_tokenize
from nltk.corpus import wordnet

#import nltk resources
resources = ["wordnet", "stopwords", "punkt", \
             "averaged_perceptron_tagger", "maxent_treebank_pos_tagger"]

for resource in resources:
    try:
        nltk.data.find("tokenizers/" + resource)
    except LookupError:
        nltk.download(resource)

#create Lemmatizer object
lemma = WordNetLemmatizer()

def lemmatize_word(tagged_token):
    """ Returns lemmatized word given its tag"""
    root = []
    for token in tagged_token:
        tag = token[1][0]
        word = token[0]
        if tag.startswith('J'):
            root.append(lemma.lemmatize(word, wordnet.ADJ))
        elif tag.startswith('V'):
            root.append(lemma.lemmatize(word, wordnet.VERB))
        elif tag.startswith('N'):
            root.append(lemma.lemmatize(word, wordnet.NOUN))
        elif tag.startswith('R'):
            root.append(lemma.lemmatize(word, wordnet.ADV))
        else:
            root.append(word)
    return root

def lemmatize_doc(document):
    """ Tags words then returns sentence with lemmatized words"""
    lemmatized_list = []
    tokenized_sent = sent_tokenize(document)
    for sentence in tokenized_sent:
        no_punctuation = re.sub(r"[`'\",.!?()]", " ", sentence)
        tokenized_word = word_tokenize(no_punctuation)
        tagged_token = pos_tag(tokenized_word)
        lemmatized = lemmatize_word(tagged_token)
        lemmatized_list.extend(lemmatized)
    return " ".join(lemmatized_list)

#apply our functions
df["preprocessed"] = df["preprocessed"].apply(lambda row: lemmatize_doc(row))

print(df["preprocessed"].iloc[1689185])
```

```
Im a big fan of the Brainwavz S1 actually all of their headphone have yet to be disappoint with any of the
ir product The S1 have be my main set for active use e g workouts run etc since the flat cable be very dur
able and resistant to tangle The S5 keep all the good feature of the S1 and add to it the sound quality be
rich and well define Thats not to say the S1 sound poor they be quite good in fact But the S5 be well The
high be well define and the midrange have more punch to it The bass come through clearly without move into
the harsh territory when the volume be push as the S1s can do The overall sound quality be very please The
build quality seem solid as solid as the S1 or good I love the flat cable I know thats something that be n
ot appreciate by everyone but for me its be work out wonderfully Although this as most other Brainwavz hea
dset come with an excellent hard shell case I usually tote my earbuds wrap around my mp3 player in my pock
et Easy to carry ; very stressful on the cable and can lead to tangle with round wire Flat wire especially
those with a thick jacket such as these survive that abuse with zero problem The earbuds themselves be mor
e sleekly shape than the can style of the S1 Comfort be in line with the customary Brainwavz style which b
e to say its outstanding It come with a wide range of tip to fit pretty much any ear plus the Comply foam
tip which be my favorite If fit properly you end up with zero ear irritation plus excellent sound isolatio
n and bass response These be an over-the-ear design much like the S1 I never use that design prior to the
S1 and it do take me a little time to get accustom to it It become second nature quickly and the design be
a lot more stable when exercise than the conventional in-ear design These be more expensive than the S1 bu
t you can hear the difference in price Still if youre look to keep the cost down a bit the S1 be an excell
ent performer as well Great sound great comfort wonderful cable design and it come with a solidly make cas
e and lot of eartips Highly recommend [ Sample provide for review ]
```

Removing Accents

Each review is normalized from longform UTF-8 to ASCII encoding. This will remove accents in characters and ensure that words like "*naïve*" will simply be interpreted as (and therefore not differentiated from) "*naive*".

Removing Punctuations

The `preprocessed` reviews are further cleaned by dropping punctuations. Using regular expressions, only whitespaces and alphanumeric characters are kept.

```
Im a big fan of the Brainwavz S1 actually all of their headphone have yet to be disappoint with any of the
ir product The S1 have be my main set for active use e g workouts run etc since the flat cable be very dur
able and resistant to tangle The S5 keep all the good feature of the S1 and add to it the sound quality be
rich and well define Thats not to say the S1 sound poor they be quite good in fact But the S5 be well The
high be well define and the midrange have more punch to it The bass come through clearly without move into
the harsh territory when the volume be push as the S1s can do The overall sound quality be very please The
build quality seem solid as solid as the S1 or good I love the flat cable I know thats something that be n
ot appreciate by everyone but for me its be work out wonderfully Although this as most other Brainwavz hea
dset come with an excellent hard shell case I usually tote my earbuds wrap around my mp3 player in my pock
et Easy to carry   very stressful on the cable and can lead to tangle with round wire Flat wire especially
those with a thick jacket such as these survive that abuse with zero problem The earbuds themselves be mor
e sleekly shape than the can style of the S1 Comfort be in line with the customary Brainwavz style which b
e to say its outstanding It come with a wide range of tip to fit pretty much any ear plus the Comply foam
tip which be my favorite If fit properly you end up with zero ear irritation plus excellent sound isolatio
n and bass response These be an over the ear design much like the S1 I never use that design prior to the
S1 and it do take me a little time to get accustom to it It become second nature quickly and the design be
a lot more stable when exercise than the conventional in ear design These be more expensive than the S1 bu
t you can hear the difference in price Still if youre look to keep the cost down a bit the S1 be an excell
ent performer as well Great sound great comfort wonderful cable design and it come with a solidly make cas
e and lot of eartips Highly recommend   Sample provide for review
```

Converting to Lowercase

Every letter is also converted to lowercase. This makes it so that "*iPhone*" will not be distinguishable from "*iphone*".

```
im a big fan of the brainwavz s1 actually all of their headphone have yet to be disappoint with any of the
ir product the s1 have be my main set for active use e g workouts run etc since the flat cable be very dur
able and resistant to tangle the s5 keep all the good feature of the s1 and add to it the sound quality be
rich and well define thats not to say the s1 sound poor they be quite good in fact but the s5 be well the
high be well define and the midrange have more punch to it the bass come through clearly without move into
the harsh territory when the volume be push as the s1s can do the overall sound quality be very please the
build quality seem solid as solid as the s1 or good i love the flat cable i know thats something that be n
ot appreciate by everyone but for me its be work out wonderfully although this as most other brainwavz hea
dset come with an excellent hard shell case i usually tote my earbuds wrap around my mp3 player in my pock
et easy to carry   very stressful on the cable and can lead to tangle with round wire flat wire especially
those with a thick jacket such as these survive that abuse with zero problem the earbuds themselves be mor
e sleekly shape than the can style of the s1 comfort be in line with the customary brainwavz style which b
e to say its outstanding it come with a wide range of tip to fit pretty much any ear plus the comply foam
tip which be my favorite if fit properly you end up with zero ear irritation plus excellent sound isolatio
n and bass response these be an over the ear design much like the s1 i never use that design prior to the
s1 and it do take me a little time to get accustom to it it become second nature quickly and the design be
a lot more stable when exercise than the conventional in ear design these be more expensive than the s1 bu
t you can hear the difference in price still if youre look to keep the cost down a bit the s1 be an excell
ent performer as well great sound great comfort wonderful cable design and it come with a solidly make cas
e and lot of eartips highly recommend   sample provide for review
```

Removing Stop Words

Stop words consist of the most commonly used words that include pronouns (e.g. *us*, *she*, *their*), articles (e.g. *the*), and prepositions (e.g. *under*, *from*, *off*). These words are not helpful in distinguishing a document from another and are therefore dropped.

Note that the `stop_words` were stripped of punctuations just as what we have done to our dataset.

```
sample stop words: ['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', 'youre', 'youve',
'youll', 'youd', 'your', 'yours']
```

```
im big fan brainwavz s1 actually headphone yet disappoint product s1 main set active use e g workouts run
etc since flat cable durable resistant tangle s5 keep good feature s1 add sound quality rich well define t
hats say s1 sound poor quite good fact s5 well high well define midrange punch bass come clearly without m
ove harsh territory volume push s1s overall sound quality please build quality seem solid solid s1 good lo
ve flat cable know thats something appreciate everyone work wonderfully although brainwavz headset come ex
cellent hard shell case usually tote earbuds wrap around mp3 player pocket easy carry   stressful cable le
ad tangle round wire flat wire especially thick jacket survive abuse zero problem earbuds sleekly shape st
yle s1 comfort line customary brainwavz style say outstanding come wide range tip fit pretty much ear plus
comply foam tip favorite fit properly end zero ear irritation plus excellent sound isolation bass response
ear design much like s1 never use design prior s1 take little time get accustom become second nature quick
ly design lot stable exercise conventional ear design expensive s1 hear difference price still look keep c
ost bit s1 excellent performer well great sound great comfort wonderful cable design come solidly make cas
e lot eartips highly recommend   sample provide review
```

Removing Extra Spaces

Again, we make use of regular expressions to ensure we never get more than a single whitespace to separate words in our sentences.

```
im big fan brainwavz s1 actually headphone yet disappoint product s1 main set active use e g workouts run
etc since flat cable durable resistant tangle s5 keep good feature s1 add sound quality rich well define t
hats say s1 sound poor quite good fact s5 well high well define midrange punch bass come clearly without m
ove harsh territory volume push s1s overall sound quality please build quality seem solid solid s1 good lo
ve flat cable know thats something appreciate everyone work wonderfully although brainwavz headset come ex
cellent hard shell case usually tote earbuds wrap around mp3 player pocket easy carry stressful cable lead
tangle round wire flat wire especially thick jacket survive abuse zero problem earbuds sleekly shape style
s1 comfort line customary brainwavz style say outstanding come wide range tip fit pretty much ear plus com
ply foam tip favorite fit properly end zero ear irritation plus excellent sound isolation bass response ea
r design much like s1 never use design prior s1 take little time get accustom become second nature quickly
design lot stable exercise conventional ear design expensive s1 hear difference price still look keep cost
bit s1 excellent performer well great sound great comfort wonderful cable design come solidly make case lo
t eartips highly recommend sample provide review
```

Tokenization

The entries for the `preprocessed` column are extracted to make up our *corpora*, which is simply a collection of all our documents. Each review is then transformed into an ordered list of words. This is the process of *tokenization* – the document is broken down into individual words or tokens.

Our tokenized sample review is below:

```
corpora = df["preprocessed"].values
tokenized = [corpus.split(" ") for corpus in corpora]

print(tokenized[1689185])
```

```
['im', 'big', 'fan', 'brainwavz', 's1', 'actually', 'headphone', 'yet', 'disappoint', 'product', 's1', 'ma
in', 'set', 'active', 'use', 'e', 'g', 'workouts', 'run', 'etc', 'since', 'flat', 'cable', 'durable', 'res
istant', 'tangle', 's5', 'keep', 'good', 'feature', 's1', 'add', 'sound', 'quality', 'rich', 'well', 'defi
ne', 'thats', 'say', 's1', 'sound', 'poor', 'quite', 'good', 'fact', 's5', 'well', 'high', 'well', 'defin
e', 'midrange', 'punch', 'bass', 'come', 'clearly', 'without', 'move', 'harsh', 'territory', 'volume', 'pu
sh', 's1s', 'overall', 'sound', 'quality', 'please', 'build', 'quality', 'seem', 'solid', 'solid', 's1',
'good', 'love', 'flat', 'cable', 'know', 'thats', 'something', 'appreciate', 'everyone', 'work', 'wonderfu
lly', 'although', 'brainwavz', 'headset', 'come', 'excellent', 'hard', 'shell', 'case', 'usually', 'tote',
'earbuds', 'wrap', 'around', 'mp3', 'player', 'pocket', 'easy', 'carry', 'stressful', 'cable', 'lead', 'ta
ngle', 'round', 'wire', 'flat', 'wire', 'especially', 'thick', 'jacket', 'survive', 'abuse', 'zero', 'prob
lem', 'earbuds', 'sleekly', 'shape', 'style', 's1', 'comfort', 'line', 'customary', 'brainwavz', 'style',
'say', 'outstanding', 'come', 'wide', 'range', 'tip', 'fit', 'pretty', 'much', 'ear', 'plus', 'comply', 'f
oam', 'tip', 'favorite', 'fit', 'properly', 'end', 'zero', 'ear', 'irritation', 'plus', 'excellent', 'soun
d', 'isolation', 'bass', 'response', 'ear', 'design', 'much', 'like', 's1', 'never', 'use', 'design', 'pri
or', 's1', 'take', 'little', 'time', 'get', 'accustom', 'become', 'second', 'nature', 'quickly', 'design',
'lot', 'stable', 'exercise', 'conventional', 'ear', 'design', 'expensive', 's1', 'hear', 'difference', 'pr
ice', 'still', 'look', 'keep', 'cost', 'bit', 's1', 'excellent', 'performer', 'well', 'great', 'sound', 'g
reat', 'comfort', 'wonderful', 'cable', 'design', 'come', 'solidly', 'make', 'case', 'lot', 'eartips', 'hi
ghly', 'recommend', 'sample', 'provide', 'review', '']
```

Phrase Modeling

Since order of words matter in most NLP models, it is often helpful to group neighboring words that appear to convey one meaning as though they are a single word, like *smart TV*.

To be considered a *phrase*, the number of times that two words should appear next to each other is set to at least `300`. The *threshold* then takes that minimum and compares it to the total number of token instances in the corpora. The higher the threshold, the more often two words must appear adjacent to be grouped into a phrase.

```
from gensim.models import Phrases
from gensim.models.phrases import Phraser

bi_gram = Phrases(tokenized, min_count=300, threshold=50)

tri_gram = Phrases(bi_gram[tokenized], min_count=300, threshold=50)
```

Bigrams

Bigrams are generated from using the *gensim* phraser. Only those that pass the `bi_gram` criteria are considered. Sample bigrams below:

```
['2_00', 'make_convenient', 'matter_fact', 'actually_see', 'sure_problem', 'problem_design', 'work_everyth
ing', 'standard_camera', '1080p_120hz', 'set_ipad', 'control_cable', 'nikon_brand', 'tiny_size', 'tiny_cam
era', 'use_default', 'plug_network', 'light_fit', 'button_click', '4kb_qd', 'wheel_click', 'hold_device',
'ipod_phone', 'might_break', 'big_small', 'noise_ratio', 'less_200', 'design_camera', 'camera_function']
```

Trigrams

Trigrams are generated by applying another *gensim* phraser on top of a bigram phraser. Take for example the tokens *sd* and *card*. Because they appear often together enough, they become linked together as *sd_card*. In turn, if *sd_card* appears adjacent to the token *reader* in enough instances, then the `tri_gram` model would link them together as well to tokenize *sd_card_reader*. Sample trigrams below:

```
['play_blu_ray', 'samsung_galaxy_s4', 'old_macbook_pro', 'quality_top_notch', 'b_w_filter', 'one_living_ro
om', 'mac_os_x', 'far_exceed_expectation', 'nexus_7_2013', 'cell_phone_use', 'customer_service_great', '5d
_mark_iii', 'cell_phone_camera', 'macbook_pro_work', 'first_blu_ray', 'case_nexus_7', 'double_sided_tape',
'price_highly_recommended', 'almost_non_existent', '2_4ghz_5ghz', 'macbook_pro_13', 'customer_service_re
p', 'samsung_840_pro', 'blu_ray_disk', 'use_third_party', 'n_uuml_vi', 'home_theater_pc', 'complete_waste_
money', 'small_form_factor', 'use_home_theater', 'fast_forward_rewind', 'wi_fi_connection', 'amazon_return
_policy', 'new_kindle_fire', '192_168_1', 'aps_c_sensor', 'ear_bud_come', 'mp3_player_work', 'mp3_player_u
se', 'use_macbook_pro', 'run_os_x', 'canon_5d_mark', 'blu_ray_movie', 'western_digital_passport', 'dd_wrt_
firmware', 'inch_macbook_pro', 'heart_rate_monitor', 'great_mp3_player', 'kindle_fire_hd', 'samsung_galaxy
_tab']
```

The `tri_gram` and `bi_gram` phrasers are applied to our `tokenized` corpora.

```
tokenized = [Phraser(tri_gram)[Phraser(bi_gram)[i]] for i in tokenized]
```

Single-character tokens are removed from every tokenized document. Our tokenized review, in its final form, is below.

```
['im', 'big', 'fan', 'brainwavz', 's1', 'actually', 'headphone', 'yet', 'disappoint', 'product', 's1', 'ma
in', 'set', 'active', 'use', 'workouts', 'run', 'etc', 'since', 'flat', 'cable', 'durable', 'resistant',
'tangle', 's5', 'keep', 'good', 'feature', 's1', 'add', 'sound', 'quality', 'rich', 'well', 'define', 'tha
ts', 'say', 's1', 'sound', 'poor', 'quite', 'good', 'fact', 's5', 'well', 'high', 'well', 'define', 'midra
nge', 'punch', 'bass', 'come', 'clearly', 'without', 'move', 'harsh', 'territory', 'volume', 'push', 's1
s', 'overall', 'sound', 'quality', 'please', 'build', 'quality', 'seem', 'solid', 'solid', 's1', 'good',
'love', 'flat', 'cable', 'know', 'thats', 'something', 'appreciate', 'everyone', 'work', 'wonderfully', 'a
lthough', 'brainwavz', 'headset', 'come', 'excellent', 'hard', 'shell', 'case', 'usually', 'tote', 'earbud
s', 'wrap', 'around', 'mp3', 'player', 'pocket', 'easy', 'carry', 'stressful', 'cable', 'lead', 'tangle',
'round', 'wire', 'flat', 'wire', 'especially', 'thick', 'jacket', 'survive', 'abuse', 'zero', 'problem',
'earbuds', 'sleekly', 'shape', 'style', 's1', 'comfort', 'line', 'customary', 'brainwavz', 'style', 'say',
'outstanding', 'come', 'wide', 'range', 'tip', 'fit', 'pretty', 'much', 'ear', 'plus', 'comply', 'foam',
'tip', 'favorite', 'fit', 'properly', 'end', 'zero', 'ear', 'irritation', 'plus', 'excellent', 'sound', 'i
solation', 'bass', 'response', 'ear', 'design', 'much', 'like', 's1', 'never', 'use', 'design', 'prior',
's1', 'take', 'little', 'time', 'get', 'accustom', 'become', 'second', 'nature', 'quickly', 'design', 'lo
t', 'stable', 'exercise', 'conventional', 'ear', 'design', 'expensive', 's1', 'hear', 'difference', 'pric
e', 'still', 'look', 'keep', 'cost', 'bit', 's1', 'excellent', 'performer', 'well', 'great', 'sound', 'gre
at', 'comfort', 'wonderful', 'cable', 'design', 'come', 'solidly', 'make', 'case', 'lot', 'eartips', 'high
ly', 'recommend', 'sample', 'provide', 'review']
```

Creating the Vocabulary

The `vocabulary` is the key-value pairs of all the unique tokens from every product review. Each token is assigned a lookup ID. The first 10 words in our dictionary are as follows:

```
from gensim.corpora.dictionary import Dictionary

vocabulary = Dictionary(tokenized)

vocabulary_keys = list(vocabulary.token2id)[0:10]

for key in vocabulary_keys:
    print(f"ID: {vocabulary.token2id[key]}, Token: {key}")
```

```
ID: 0, Token: address
ID: 1, Token: around
ID: 2, Token: arrive
ID: 3, Token: back
ID: 4, Token: bad
ID: 5, Token: big
ID: 6, Token: come
ID: 7, Token: contact
ID: 8, Token: could
ID: 9, Token: day
```

Count-based Feature Engineering

In order for a machine learning model to work with text input, the document must first be *vectorized*. This simply means that the input has to be converted into containers of numerical values.

Bag of Words Model

The classical approach in expressing text as a set of features is getting the token frequency. Each entry to the dataframe is a document while each column corresponds to every unique token in the entire corpora. The row will identify how many times a word appears in the document. The `bow` model for the sample review is below:

```python
bow = [vocabulary.doc2bow(doc) for doc in tokenized]

for idx, freq in bow[0]:
    print(f"Word: {vocabulary.get(idx)}, Frequency: {freq}")
```

```
Word: address, Frequency: 1
Word: around, Frequency: 1
Word: arrive, Frequency: 1
Word: back, Frequency: 1
Word: bad, Frequency: 1
Word: big, Frequency: 2
Word: come, Frequency: 1
Word: contact, Frequency: 1
Word: could, Frequency: 1
Word: day, Frequency: 1
Word: earlier, Frequency: 1
Word: ease, Frequency: 2
Word: ect, Frequency: 1
Word: email, Frequency: 2
Word: exception, Frequency: 1
Word: exchange, Frequency: 1
Word: expect, Frequency: 1
Word: freeze, Frequency: 2
```

TF-IDF Model

The Term Frequency-Inverse Document Frequency (*TF-IDF*) approach assigns continuous values instead of simple integers for the token frequency. Words that appear frequently overall tend to not establish saliency in a document, and are thus weighted lower. Words that are unique to some documents tend to help distinguish it from the rest and are thus weighted higher. The `tfidf` weighting is based on our `bow` variable.

```python
from gensim.models.tfidfmodel import TfidfModel

tfidf = TfidfModel(bow)

for idx, weight in tfidf[bow[0]]:
    print(f"Word: {vocabulary.get(idx)}, Weight: {weight:.3f}")
```

```
Word: address, Weight: 0.113
Word: around, Weight: 0.060
Word: arrive, Weight: 0.093
Word: back, Weight: 0.051
Word: bad, Weight: 0.068
Word: big, Weight: 0.126
Word: come, Weight: 0.046
Word: contact, Weight: 0.103
Word: could, Weight: 0.054
Word: day, Weight: 0.061
Word: earlier, Weight: 0.141
Word: ease, Weight: 0.220
Word: ect, Weight: 0.181
Word: email, Weight: 0.213
Word: exception, Weight: 0.131
Word: exchange, Weight: 0.132
Word: expect, Weight: 0.067
Word: freeze, Weight: 0.259
```

Word Embedding for Feature Engineering

The downside of count-based techniques is that without regard to word sequence and sentence structure, the semantics get lost. The *Word2Vec* technique, on the other hand, actually embeds meaning in vectors by quantifying how often a word appears within the vicinity of a given set of other words.

A context window the span of `context_size` slides across every document one token at a time. In each step, the center word is described by its adjacent words and the probability that the token appears together with the others is expressed in `feature_size` dimensions. Since the minimum word requirement is set to `1`, every token in the corpora is embedded in the *Word2Vec* model.

```
import numpy as np

from gensim.models import word2vec

np.set_printoptions(suppress=True)

feature_size = 100
context_size = 20
min_word = 1

word_vec= word2vec.Word2Vec(tokenized, size=feature_size, \
                            window=context_size, min_count=min_word, \
                            iter=50, seed=42)
```

Final Dataframe

The goal is to have a dataframe with observations corresponding to the product reviews. The `word_vec` model is used to gather all the unique tokens in the corpora. This enables us to generate the `word_vec_df` which makes use of the dimensions as the features of every word.

```
word_vec_unpack = [(word, idx.index) for word, idx in \
                   word_vec.wv.vocab.items()]

tokens, indexes = zip(*word_vec_unpack)

word_vec_df = pd.DataFrame(word_vec.wv.syn0[indexes, :], index=tokens)

display(word_vec_df.head())
```

	0	1	2	3	4	5	...	95	96	97	98	99	96	97	98	99
get	2.027105	2.285539	0.325559	5.013640	-0.886071	-2.239007	...	-4.002328	5.016023	1.077161	-0.641248	1.685374	5.016023	1.077161	-0.641248	1.685374
gps	4.430076	-1.912336	9.017261	3.536343	-9.298578	-3.119642	...	-0.979287	6.612934	-4.329911	5.106474	7.303112	6.612934	-4.329911	5.106474	7.303112
husband	1.160196	4.993967	1.216444	2.476656	-3.268333	1.352225	...	-2.101081	5.262798	0.258067	-0.073419	-2.713029	5.262798	0.258067	-0.073419	-2.713029
otr	1.308931	-1.700580	2.168358	2.334967	0.032168	0.996840	...	0.947201	-0.885584	-0.979156	0.709682	1.676071	-0.885584	-0.979156	0.709682	1.676071
road	3.249146	2.961001	10.251733	3.529565	-7.407037	-1.892380	...	0.373690	10.218236	-3.727395	-3.895083	6.060423	10.218236	-3.727395	-3.895083	6.060423

5 rows × 100 columns

The `word_vec_df` is sliced by the words that appear in a given `tokenized` review and the mean along every dimension is taken. The resulting `model_array` shape is therefore the word count on *axis 0* and the number of dimensions on *axis 1*. This singularizes multiple word embeddings into one observation for each review.

If multiple occurrences of a word occurs in a review, then this only emphasizes the token since the row is pulled towards the values of the vectors of that word.

```
tokenized_array = np.array(tokenized)

model_array = np.array([word_vec_df.loc[doc].mean(axis=0) for doc in tokenized_array])
```

Every document is provided the ground truth label by imposing its `overall` rating. This completes our finalized `model_df` dataframe.

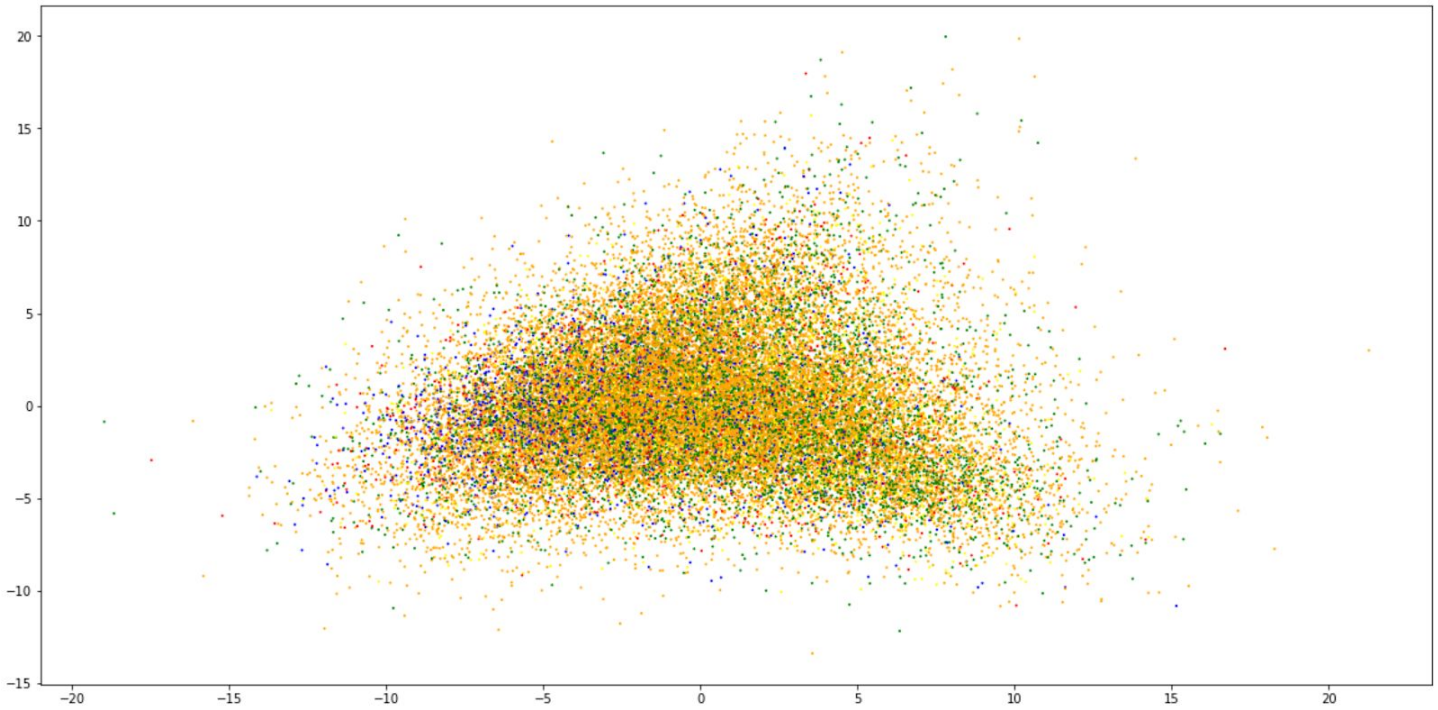
	0	1	2	3	...	96	97	98	99	label
0	0.456036	-0.340525	-0.423433	1.381710	...	1.881948	-1.110850	-0.336875	0.206793	5
1	-0.127207	0.409159	0.727547	0.947930	...	1.551120	-0.726942	-0.235786	1.039625	1
2	-0.782381	-0.635467	0.310187	1.470468	...	1.439608	-1.896147	-0.506262	0.904398	3
3	0.079711	-0.142391	0.528292	1.845955	...	1.987752	-1.498181	-0.840509	0.573334	2
4	0.446526	-0.167763	0.217617	0.934713	...	2.020145	-0.689082	0.123893	1.976123	1

5 rows × 101 columns

Principal Component Analysis

Principal Component Analysis (*PCA*) is a dimensionality reduction technique that we can use on our `model_df` to reduce its 100 dimensions to just two dimensions. This will help visualize if there is a clear decision boundary along the five `overall` rating classifications. The more datapoints belonging to the same class are clustered together, the higher the likelihood that our machine learning model is simpler and more effective.

```
import matplotlib.pyplot as plt

from sklearn.decomposition import PCA

#sampling the model_df population
pca_df = model_df.reset_index()
pca_df = model_df.dropna(axis=0).iloc[:,1:]
pca_df = pca_df.iloc[::50]

#setting up PCA
pca = PCA(n_components=2, random_state=42)
pca = pca.fit_transform(pca_df.iloc[:, :-1])
labels = pca_df["label"]

#setting up plot components
x_axis = pca[:,0]
y_axis = pca[:,1]
color_map = pca_df["label"].map({1:"blue", \
                                 2:"red", \
                                 3:"yellow", \
                                 4:"green", \
                                 5:"orange"})

#plotting PCA
f, axes = plt.subplots(figsize=(20,10))
plt.scatter(x_axis, y_axis, color=color_map, s=1)
plt.show()
```



Exploratory Data Analysis

We'll implement several interesting Natural Language Processing techniques in order to explore our Amazon dataset.

More on Word2Vec

To better appreciate the concept of word embeddings, we take five common words in our corpora and derive their five most related words using our `word_vec` model. The similarity comes from how often these tokens appear in the same window of words as their `word_bank` counterpart.

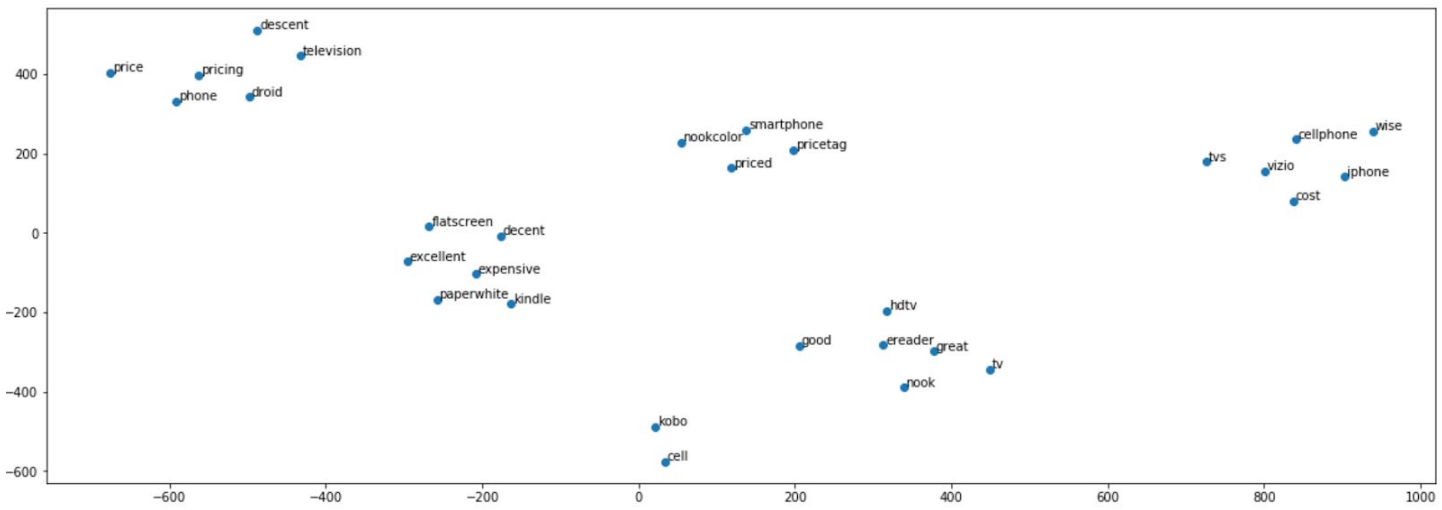
```
word_bank = ["nook", "phone", "tv", "good", "price"]

for word in word_bank[:]:
    related_vec = word_vec.wv.most_similar(word, topn=5)
    related_words = np.array(related_vec)[:,0]
    word_bank.extend(related_words)
    print(f"{word}: {related_words}")
```

```
nook: ['kindle' 'ereader' 'nookcolor' 'kobo' 'paperwhite']
phone: ['cellphone' 'smartphone' 'cell' 'droid' 'iphone']
tv: ['television' 'hdtv' 'tvs' 'vizio' 'flatscreen']
good: ['decent' 'great' 'wise' 'excellent' 'descent']
price: ['pricing' 'cost' 'priced' 'pricetag' 'expensive']
```

t-SNE

Like PCA, the t-Distributed Stochastic Neighbor Embedding (*t-SNE*) is another dimensionality reduction technique to assist in visualizing high-dimensional datasets. To perceive the similarity between the related words in terms of spatial distance, t-SNE provided the coordinates of each word in a 2D scatterplot plane.

```
from sklearn.manifold import TSNE

tsne = TSNE(n_components=2, perplexity=5, n_iter=1000, random_state=42)

sample_vecs = word_vec.wv[set(word_bank)]
sample_tsne = tsne.fit_transform(sample_vecs)
tsne_x = sample_tsne[:, 0]
tsne_y = sample_tsne[:, 1]

f, axes = plt.subplots(figsize=(20,7))
ax = plt.scatter(x=tsne_x, y=tsne_y)

for label, x, y in zip(word_bank, tsne_x, tsne_y):
    plt.annotate(label, xy=(x+3, y+3))

plt.show()
```



Word Algebra

Since *Word2Vec* characterizes words into quantified tokens, we can consequently add or subtract word vectors together. To add is to combine the meaning of the components and to subtract is to take out the context of one token from another. The following are examples of this vector algebra and their similarity scores:

Books + Touchscreen

```
word_vec.wv.most_similar(positive=["books", "touchscreen"], \
                         negative=[], topn=1)
```

```
[('ebooks', 0.6847387552261353)]
```

Cheap – Quality

```
word_vec.wv.most_similar(positive=["cheap"], \
                         negative=["quality"], topn=1)
```

```
[('ebay', 0.4894425570964813)]
```

Tablet – Phone

```
word_vec.wv.most_similar(positive=["tablet"], \
                         negative=["phone"], topn=1)
```

```
[('netbooks', 0.5097050666809082)]
```

Named Entity Recognition

We've seen *gensim* perform word tagging to identify part-of-speech. Now we use *spaCy* to go further and identify what nouns in the documents refer to. Some Named-Entity Recognition (*NER*) classification tags include distinguishing persons, organizations, products, places, dates, etc.

In exploring *spaCy*, we'll be using the `most_helpful_text`, which is the highest-rated product review by Amazon users. The `helpful` series from the `df` dataframe is actually a list with its first element storing the number of *helpful* votes a review received, and the second element containing the total number of *helpful* and *not helpful* review votes.

```
helpful = df["helpful"].tolist()
most_helpful = max(helpful, key=lambda x: x[0])

most_helpful_idx = df["helpful"].astype(str) == str(most_helpful)
most_helpful_idx = df[most_helpful_idx].index

most_helpful_text = df["reviewText"].iloc[most_helpful_idx].values[0]

print(most_helpful_text)
```

```
I've been an iPad user since the original came out. I also have an iPad 3. I have worked in IT for the pas
t few years so I would say I am pretty good with technology and fancy new devices. With that introduction
out of the way, I will be reviewing key points that I have seen touched upon in other reviews. Here goe
s...BUILDThe device feels nice and solid. I'm a little surprised at how heavy it is, but that's not necess
arily a bad thing. The rubberized backing is always nice for added grip. It's not as nice as say unibody a
luminum, but it's not $500 either.SCREENThe screen is fantastic. But my problem is the same as when iPad g
ot Retina Display, other than the OS, most apps look rather pixelated. A lot of the games I tried are not
high definition, at least not high enough to look smooth on this screen. Hopefully apps get updated to hig
her resolutions.LOCK SCREEN ADSYeah there are ads on my lock screen. I'm not sure why this is such a big d
eal. How much time do people really spend looking at the lock screen? The first thing I thought when I saw
the ads is WOW the pictures are really crisp! The ads are there to subsidize some of the $200 price tag. I
might pay the $15 to get rid of them so I can customize it, but I might not. I feel like this has been blo
wn out of proportion by other customers.SOUNDThe sound from the speakers is great. Much better than you wo
uld get from more expensive devices, very crisp and clean. I have the official Amazon case on and it has n
ot affected the sound at all. Nothing much else to say, I doubt anyone will complain about this.CRASHING
I've had two apps crash on opening. I don't know if it is the app or the OS. It's probably somewhere in th
e middle. Again, not a big deal for me. If it crashes, then I just tap it again and it works...
```

We use `ner_dict`, a dictionary initialized as a list, to segregate the nouns in the `most_helpful_text` into the NER tags.

```
import spacy

from collections import defaultdict

ner = spacy.load("en")

ner_helpful = ner(most_helpful_text)

ner_dict = defaultdict(list)
for entity in ner_helpful.ents:
    ner_dict[entity.label_].append(entity)

for NER, name in ner_dict.items():
    print(f"{NER}:\n{name}\n")
```

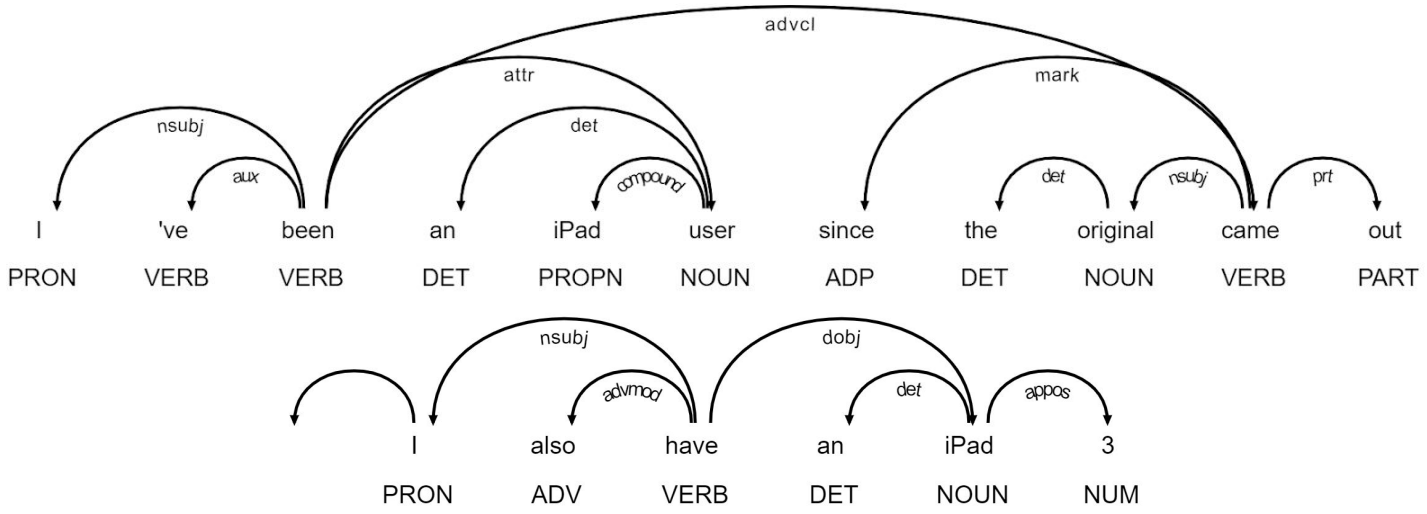
```
ORG:
[iPad, iPad, OS, LOCK SCREEN, Amazon, OS, Amazon, iPad, iPad, Amazon, iPad, iPad, Power/Volume Buttons, iP
ad, Seek & Find, Amazon, iPad, MUSICI, Samsung Galaxy S3, iSyncr, Amazon, Amazon Customer Service, Amazon,
Amazon, Apple, Amazon, Amazon, HDMI CONNECTIONI, HDMI, Samsung, Amazon, OS, OS, LEFT HAND MODEI, Avia Medi
a Player, GOOGLE, Google]

DATE:
[the past few years, the past 24, 9/20/12Two days later, an appointment days later, this past weekend]

MONEY:
[500, 200, 15, another $200+, 200.UPDATE, 3]

PERSON:
[Retina Display, Netflix, to.-, Screen Glare, Cloud Player, Battery HD, DID, Home, APPI]

ORDINAL:
[first]

CARDINAL:
[two, 46, two, 5, 4]

GPE:
[Caltrain, Palo Alto, iTunes, Cloud, Cloud, Kindle, Kindle, Bluray, Kindle, mp4, 11/15/12Still, Solitaire,
LA, SF, Kindle]

NORP:
[Enigmatis]

TIME:
[about an hour and a half, 5 minute]
```

We use `displaCy` to visualize the tags in the review.

I've been an iPad ORG user since the original came out. I also have an iPad 3. I have worked in IT for the past few years DATE so I would say I am pretty good with technology and fancy new devices. With that introduction out of the way, I will be reviewing key points that I have seen touched upon in other reviews. Here goes...BUILDThe device feels nice and solid. I'm a little surprised at how heavy it is, but that's not necessarily a bad thing. The rubberized backing is always nice for added grip. It's not as nice as say unibody aluminum, but it's not $ 500 MONEY either.SCREENThe screen is fantastic. But my problem is the same as when iPad ORG got Retina Display PERSON , other than the OS ORG , most apps look rather pixelated. A lot of the games I tried are not high definition, at least not high enough to look smooth on this screen. Hopefully apps get updated to higher resolutions. LOCK SCREEN ORG ADSYeah there are ads on my lock screen. I'm not sure why this is such a big deal. How much time do people really spend looking at the lock screen? The first ORDINAL thing I thought when I saw the ads is WOW the pictures are really crisp! The ads are there to subsidize some of the $ 200 MONEY price tag. I might pay the $ 15 MONEY to get rid of them so I can customize it, but I might not. I feel like this has been blown out of proportion by other customers.SOUNDThe sound from the speakers is great. Much better than you would get from more expensive devices, very crisp and clean. I have the official Amazon ORG case on and it has not affected the sound at all. Nothing much else to say, I doubt anyone will complain about this.CRASHINGI've had two CARDINAL apps crash on opening. I don't know if it is the app or the OS ORG . It's probably somewhere in the middle. Again, not a big deal for me. If it crashes, then I just tap it again and it works. I've also watched a few movies using the built in player as well as Netflix PERSON and Amazon ORG Prime. No crashes for me at all. I'm sure OS stability will be improved as time goes on.OVERALL SATISFACTIONCompared to my iPad 3, obviously the Fire HD is not as "good" so to speak. I mainly got it because I wanted something smaller. I also mainly used the iPad ORG to surf the web, watch videos, and play some simple games. The Fire HD accomplishes this and does so much more. If you are expecting an iPad ORG killer, or a desktop replacement, or a productivity machine, then you should look elsewhere.I bought this to be a media device, and I believe that is what Amazon ORG meant this to be. In this regard, I think this is a great device. In fact, I decided to keep this and sell my iPad ORG 3, which will give me another $200+ MONEY to spend on other things. Just remember, this device is not for everyone. If you want a media device, you will be happy with this. Do not expect an iPad ORG for $ 200.UPDATE MONEY 9/18/12Just wanted to add a few more things I have noticed over the past 24 DATE hours.- Power/Volume Buttons ORG : There are a bit hard to press, which is somewhat alleviated by having the official case. Maybe it's because I'm a longtime iPad ORG user, but this will definitely take some getting used to.- PERSON Screen Glare PERSON : It took me a little while to notice, but I was playing a Seek & Find ORG game while on Caltrain GPE , with the bright Palo Alto GPE sun shining right on me, and didn't have any trouble seeing the screen. I remembered that Amazon ORG mentioned how the screen was changed to reduce glare, and they did an amazing job.UPDATE 9/20/12Two days later DATE and I am still very happy

Dependency Tree

The capability of *spaCy*'s NER is based on deciphering the structure of the sentence by breaking down how tokens interact with and influence each other. Below is the dependency trees of the first two sentences of the `most_helpful_text`.



Topic Modeling

Because Latent Dirichlet Allocation (*LDA*) can cluster documents together according to topic, the reviews can be classified and grouped according to the type of electronics product they correspond to. The product reviews will have weights assigned to each of the topic and the topics themselves will have weights on every token. As it is a clustering-based model, LDA is unsupervised and only the `num_topics` is configurable.

The following are the top five words that are salient to the first group of product reviews.

```
import multiprocessing

from gensim.models.ldamulticore import LdaMulticore

cores = multiprocessing.cpu_count()

num_topics = 10
bow_lda = LdaMulticore(bow, num_topics=num_topics, id2word=vocabulary, \
                       passes=5, workers=cores, random_state=42)

for token, frequency in bow_lda.show_topic(0, topn=5):
    print(token, frequency)
```

```
get 0.011734774
work 0.011198829
use 0.0095958505
router 0.009158912
device 0.008855804
```

The words that are the most characteristic of the topics are indeed thematic. And each word group do conjure a distinct topic.

```
Topic 1:
 get, 0.011734774336218834
 work, 0.011198828928172588
 use, 0.009595850482583046
 router, 0.009158912114799023
 device, 0.008855803869664669

Topic 2:
 sound, 0.034996841102838516
 speaker, 0.0209796279668808
 good, 0.015011309646070004
 headphone, 0.013791842386126518
 quality, 0.011160140857100487

Topic 3:
 lens, 0.03160819783806801
 bag, 0.018301470205187798
 camera, 0.014126963913440704
 use, 0.01146912481635809
 strap, 0.008649271912872791

Topic 4:
 cable, 0.031109070405364037
 work, 0.02897009812295437
 usb, 0.025115681812167168
 drive, 0.023603621870279312
 great, 0.017832685261964798

Topic 5:
 case, 0.03297600895166397
 cover, 0.012749520130455494
 screen, 0.011409485712647438
 like, 0.011152341961860657
 ipad, 0.010687867179512978

Topic 6:
 battery, 0.02778122015297413
 charge, 0.025451648980379105
 use, 0.017372693866491318
 phone, 0.016959380358457565
 one, 0.01309022307395935

Topic 7:
 camera, 0.0475146621465683
 use, 0.01570322923362255
 picture, 0.012167769484221935
 take, 0.011743015609681606
 video, 0.011700318194925785

Topic 8:
 card, 0.03473054617643356
 drive, 0.01358714234083891
 fan, 0.010703792795538902
 run, 0.0096812155097723
 memory, 0.009252899326384068

Topic 9:
 use, 0.023249300196766853
 keyboard, 0.02211669646203518
 tablet, 0.015327784232795238
 mouse, 0.011338042095303535
 like, 0.01100770290941

Topic 10:
 tv, 0.029768439009785652
 remote, 0.011976256966590881
 use, 0.010298002511262894
 get, 0.009141155518591404
 watch, 0.008147124201059341
```

Using *pyLDAvis*, we can interactively explore the words associated with the topics derived by LDA. The Intertopic Distance Map shows how some product reviews in one topic converge with others due to similarity. If needed, we

can adjust the `num_topics` accordingly to cluster together topic intersections so a more evident decision boundary between classes can be established.

In the visualization, topic 3 is shown to be describing reviews that pertain to audio products and thus the key words: *sound*, *speaker*, *headphone*, etc.

```
import pyLDAvis.gensim

lda_idm = pyLDAvis.gensim.prepare(bow_lda, bow, vocabulary)

pyLDAvis.display(lda_idm)
```