Project Phase 1 Report - Data Pre-processing

Team Members: Arno Dunstatter, Ethan Reyna, Daniel Fernandez
Our GitHub repository can be found here.

Data Description

The goal is to perform the Natural Language Processing task of sentiment analysis on financial headlines, classifying them as either negative, neutral, or positive. To this end three datasets were obtained for the purpose of training a sentiment analysis model, all of which contain news headlines, which will be used as input variables, and sentiment classifications, which will be used as output variables.

The primary dataset contains 4837 unique observations of news headlines generally relevant to finance, each classified as either having a “positive”, “neutral”, or “negative” sentiment. Originally this dataset had the following columns and datatypes:

```
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Sentiment      4846 non-null   object
 1   News Headline  4846 non-null   object
```

The secondary dataset is similar, but only has 211 unique observations, and instead classifies the sentiment with numbers: 0 for negative, 1 for neutral, and 2 for positive. Originally this dataset had the following columns and datatypes:

```
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   text       211 non-null    object
 1   sentiment  211 non-null    int64
```

The tertiary dataset is specific to a particular company, namely Reliance Industries Limited. This dataset contains 598 unique observations and, as in the primary dataset, uses verbal categories of “positive”, “neutral”, and “negative”. This dataset differs from the other two in that it has a few additional columns, namely a description of the article, the articles’ URLs, as well as a real numbered sentiment score. Originally this dataset had the following columns and datatypes:

```
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   published_at     598 non-null    object
 1   title            598 non-null    object
 2   description      598 non-null    object
 3   url              598 non-null    object
 4   sentiment        598 non-null    object
 5   sentiment_score  598 non-null    float64
```

Preprocessing and Feature Engineering

In order to reduce the dimensionality of the three datasets the following unnecessary columns were dropped from the tertiary dataset: 'published_at', 'description', 'url', 'sentiment_score'. These

columns were not present in the other two data sets, so this reduction of dimensionality brought this dataset into agreement with the dimensionality of the other two.

Next the values for the respective sentiment columns of each dataset were adjusted to agree with each other. Originally the three datasets used different conventions to signify each news headline's sentiment towards their respective subjects. Specifically, the primary and tertiary datasets had three different strings as their sentiment values: "positive", "negative", and "neutral". The secondary dataset has the three sentiments listed out as 0 (negative sentiment), 1 (neutral sentiment), and 2 (positive sentiment). While performing data preprocessing the sentiment encoding was standardized as follows: news articles with a positive sentiment were assigned a value of '1', articles with a neutral sentiment were assigned '0', and articles with a negative sentiment were assigned '-1'. Furthermore the names of the columns in each dataset were standardized. Both of these standardizations made it possible to then combine all three datasets into a single pandas dataframe simply entitled "data", then containing all 5655 of the observations from the original three datasets. This dataframe had the following columns and datatypes:

#	Column	Non-Null Count	Dtype
0	headline	5655 non-null	object
1	sentiment	5655 non-null	int64

Furthermore, as this project revolves around the analysis of news articles' headlines, it was necessary to make sure there were no invalid headlines that could cause the model to become skewed or otherwise corrupted. For example, in the primary dataset, there were originally a few headlines with the character value: 'Æ', which made the relevant headlines not make sense, particularly in regards to a financial standpoint. The python regular expressions library was employed to remove all observations which contained irregular characters which might obscure the headlines' actual sentiment. This brought the length of our training data down to 5003 observations.

In addition, we removed all individual numbers and for the visualization we also removed any nouns that included a number such as "N95", such that only letters and the spaces remain.

Next we adjusted all contractions in the headlines to read out as full words. For example, rather than having 'isn't', we needed this to read out as 'is not', or this would cause troubles down the line when we begin our training process, as the machine learning program could misread some of these words and cause changes in the overall sentiment values. To do so, we devised an algorithm to search through the dataset to find any present contractions based off of a contraction dataset we found online. Once this ran, it was able to make the proper changes to our final dataset, and with the final appropriate data for our project, we were able to proceed with exploratory data analysis.

After contraction expansion we utilized the spacy and en_core_web libraries to remove all stopwords and lemmatized the remaining meaningful words. These same libraries were then used to perform named entity recognition wherein new columns were added to our data's dataframe. One column contained an ordered dictionary where the keys were tuples with the first element being the name, and

the second element being the entity type, and each key's item was the number of occurrences of that entity in the given headline. The other columns added represented the total counts of each entity type. For instance if the headline was "Congress must pass a defense spending bill soon or the Airforce will run out of money for their next weapon system!" the tags_ORG column would have an entry of 2, since there are two instances of an organization being mentioned. These features were added to our data under the presumption that there will be a relationship between certain organization types being present in a headline and the overall headline sentiment. For instance, often mentions of governments in headlines about cryptocurrency are harbingers of bad news.

	original_headline	sentiment	cleaned_headline	has_numbers	has_monetary_units	tags	tags_MONEY	tags_NORP	tags_GPE	tags_QUANTITY	...	tags_EVENT	tags_C
0	According to Gran , the company has no plans t...	0	accord gran company plan production russia com...	0	0	[{('russia', 'GPE'): 1}]	0	0	1	0	...	0	
1	Technopolis plans to develop in stages an area...	0	technopoli plan develop stage area square mete...	1	0	[]	0	0	0	0	...	0	
3	With the new production plant the company woul...	1	new production plant company increase capacity...	0	0	[]	0	0	0	0	...	0	
4	According to the company 's updated strategy f...	1	accord company update strategy year basware ta...	1	0	[]	0	0	0	0	...	0	
5	FINANCING OF ASPOCOMP 'S GROWTH Aspocomp is ag...	1	financing aspocomp growth aspocomp aggressivel...	0	0	[{('hdi print circuit board', 'ORG'): 1}]	0	0	0	0	...	0	

5 rows × 23 columns

Exploratory Data Analysis

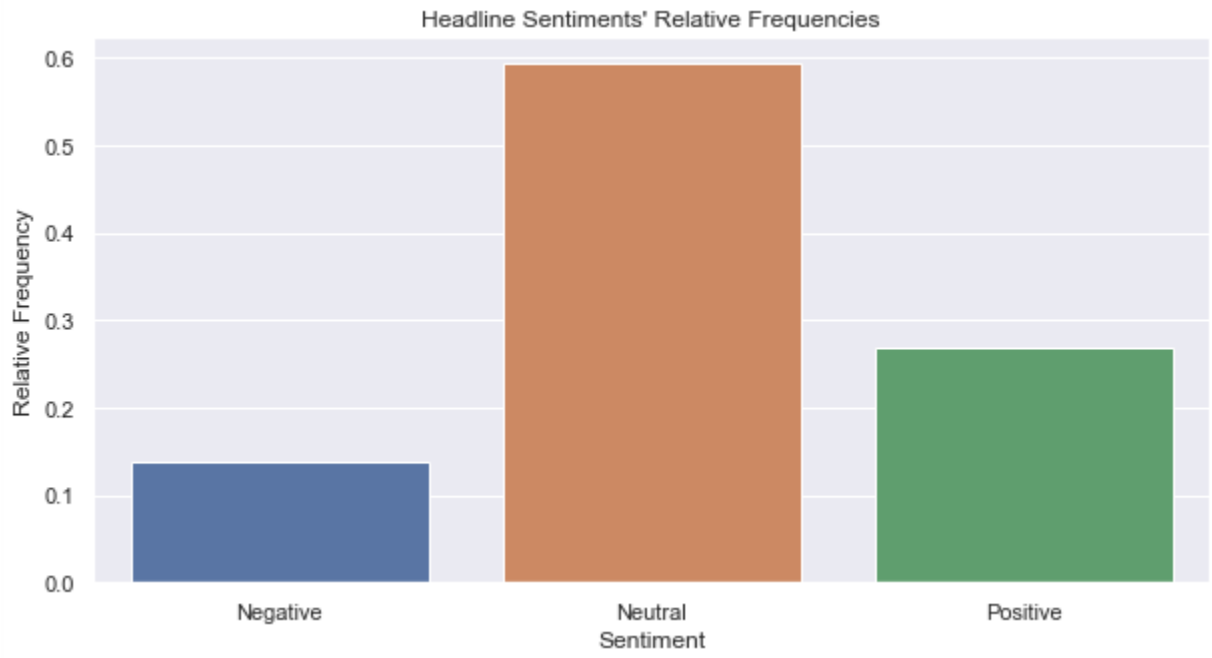
In regards to our data analysis for this project, we knew that our visualization would look a little bit different than the average EDA. This is due to the fact that our project revolves around Natural Language Processing, meaning a lot of our analysis would be around the individual words that these articles contain. In order to display relevant information to understand our data, we decided on four methods of visualization/explanation. These four methods include: displaying headline sentiment distribution via barcharts, counting the most frequent words in each sentiment class, developing a word cloud that would show the most common words in the overall dataset, and collecting N-Grams (frequency of assortments of words based on an N value). We've displayed and discussed our analysis using these methods below.

Headlines' Sentiment Distribution via Barcharts

The frequency of all classes (-1 for negative, 0 for neutral, and 1 for positive) were counted and the relative frequencies were computed:

	Sentiment	Frequency	Relative Frequency
0	Negative	692	0.138317
1	Neutral	2966	0.592844
2	Positive	1345	0.268839

The distribution across the three sentiment-classes is shown by a barchart of the relative frequency of each sentiment within the overall data:

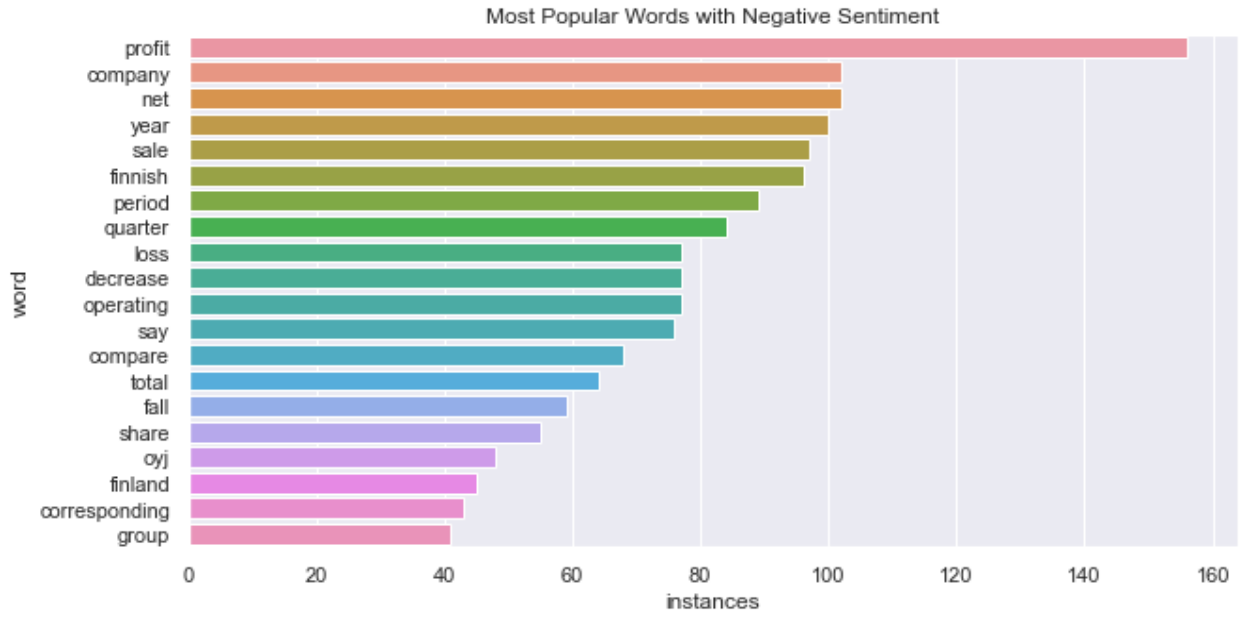


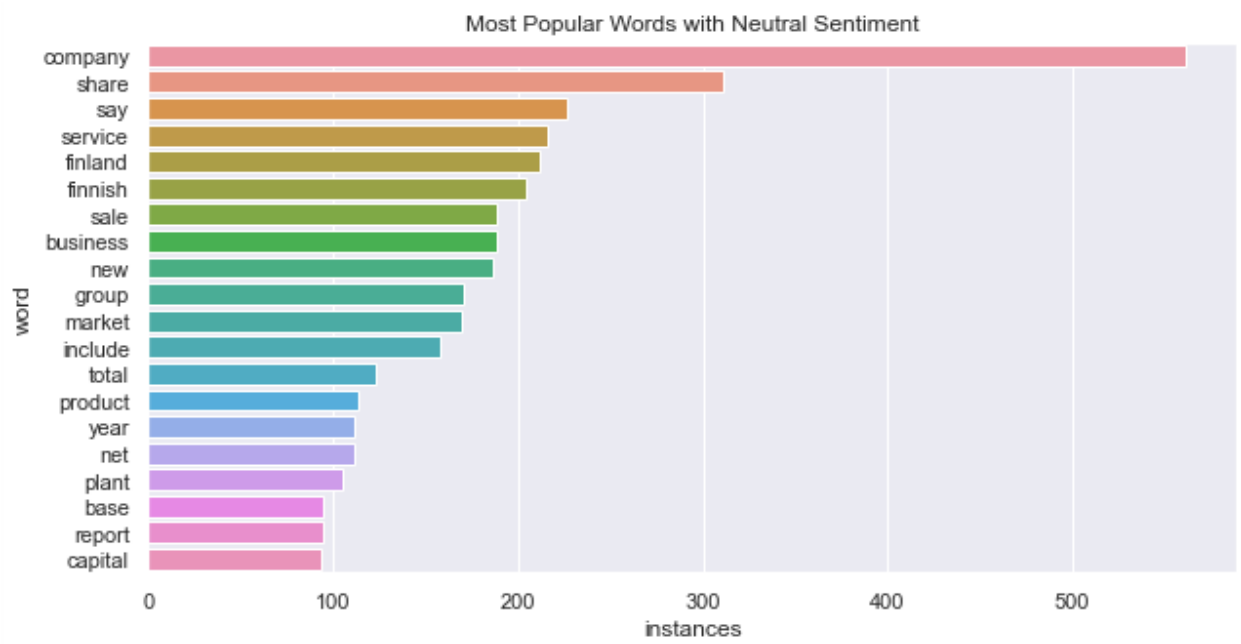
Counting the Most Frequent Words in each Sentiment Class

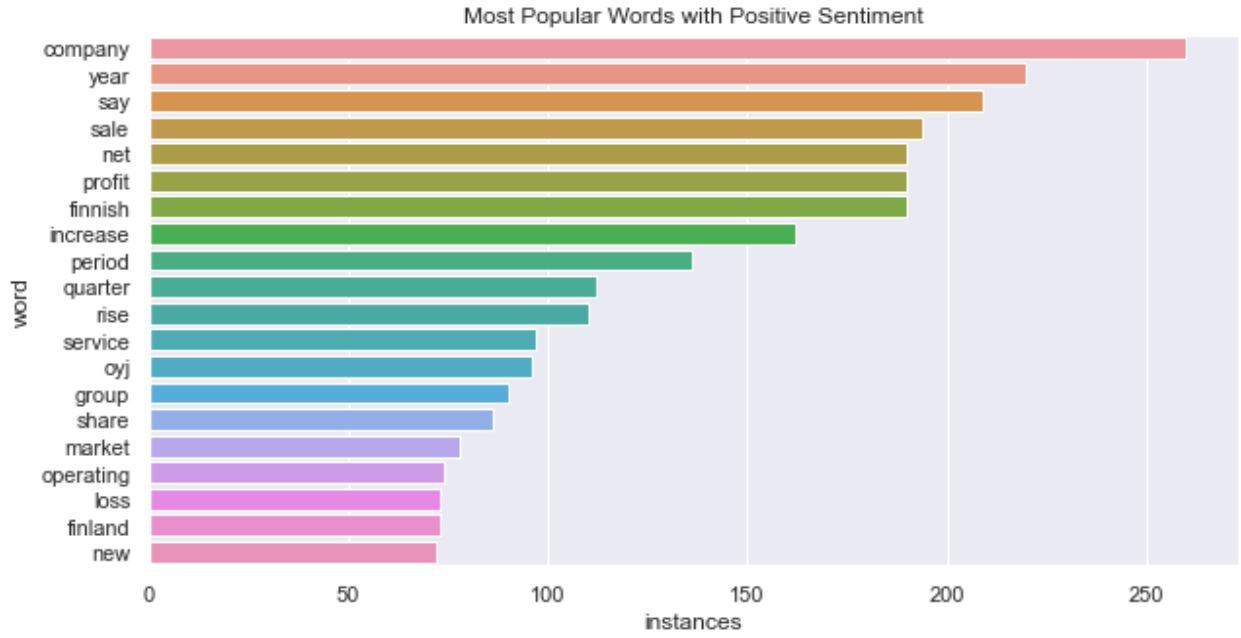
In order to get further insight on our data post-processing, and to have a better understanding of how our machine learning will interpret our data, we decided to do a specific analysis of the individual words listed in each article. Initially, in order to get a full set of words that would be relevant to this process, we removed any outliers that would cloud our final outcome. This ended up being symbols or characters that would be useful to a training set, but would not be applicable to this specific analysis, such as a dollar sign or a colon. Once we reduced the data set down to these relevant words, we separated the articles into three separate datasets based on their sentiment: positive, negative, and neutral. With these separated datasets, we now had all of our articles with a wide variety of words segmented into their sentiment, which would allow us to find which words seemed to have the largest impact on an article's sentiment. We implemented a Counter function from the collections library, which counted every word in the data set and listed out the most common. We pushed this to another dataframe, which we could then

apply our visualization to. With this final data, we gathered the twenty most popular words, then displayed them on a horizontal bar graph. We've listed the graphs for this analysis below.

By undergoing this process, we were able to find a good bit of interesting information regarding the impact of some words in the dataset. There were a few instances of words that come up a lot in financial topics such as quantification of money (million, mostly), and other nouns such as business, company, market. However, upon further inspection, there were some clear give-aways as to why an article was marked with a specific sentiment. For example, the graph displaying the 'Most Popular Words with Positive Sentiment' has many words that would come with the territory of being a generally beneficial thing to hear, such as 'rise', 'increase', and 'new'. These tended to be much more relevant in this dataset than the other two, and can almost always be a clear sign that a headline would be a positive sentiment. On the other hand, in the graph displaying the 'Most Popular Words with Negative Sentiment', we received words with a more detrimental connotation such as 'loss' an/d 'decrease'. Through this process, we also found out that the word 'profit', once thought to be a word that would glean positive results, happened to actually be as present in negative headlines as they would be with the positive ones. This is most likely due to the different ways profit is used in a sentence and with more connotation, but despite this, the word itself wasn't found at all when scanning through the most common words with neutral sentiment. This seemed to be an important word to flag when trying to determine if an article swayed one way or the other.

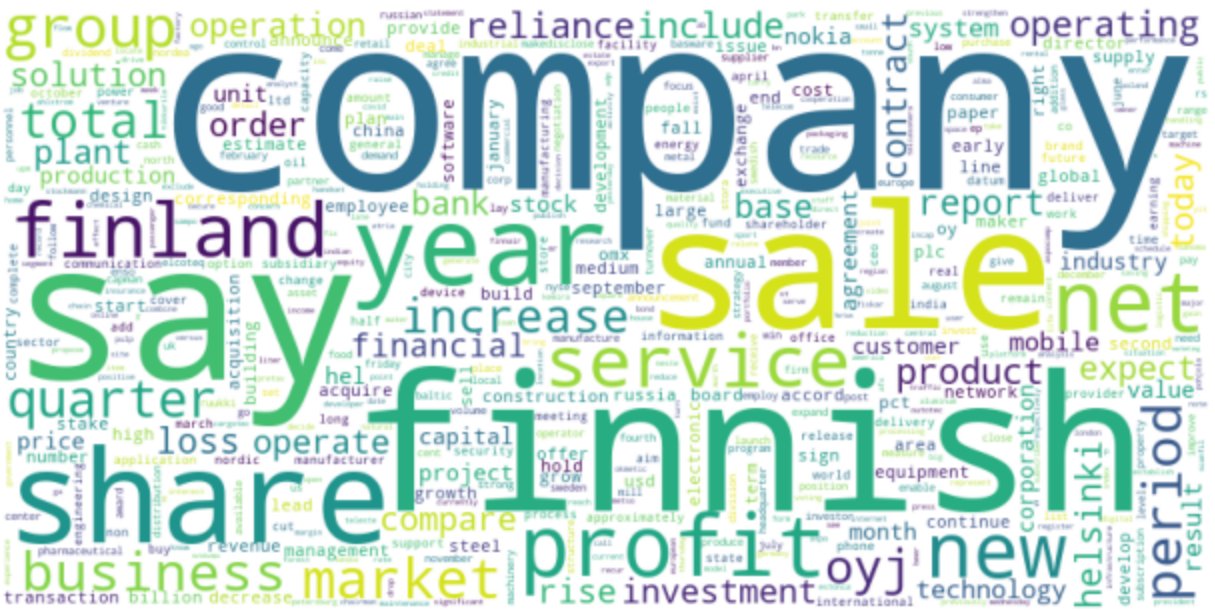






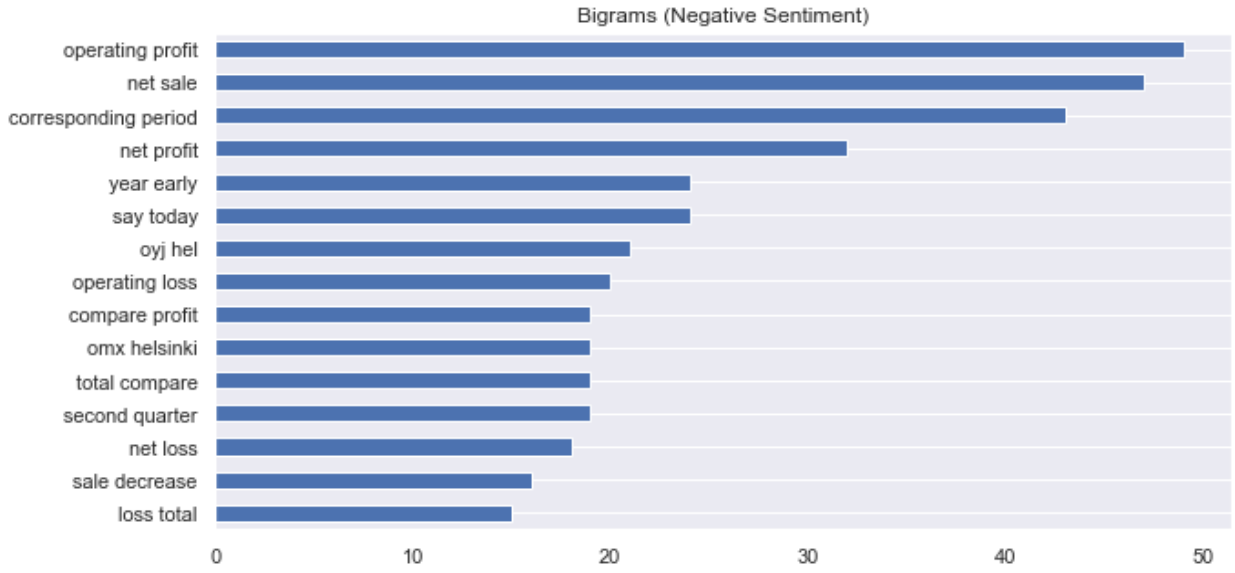
Word Cloud

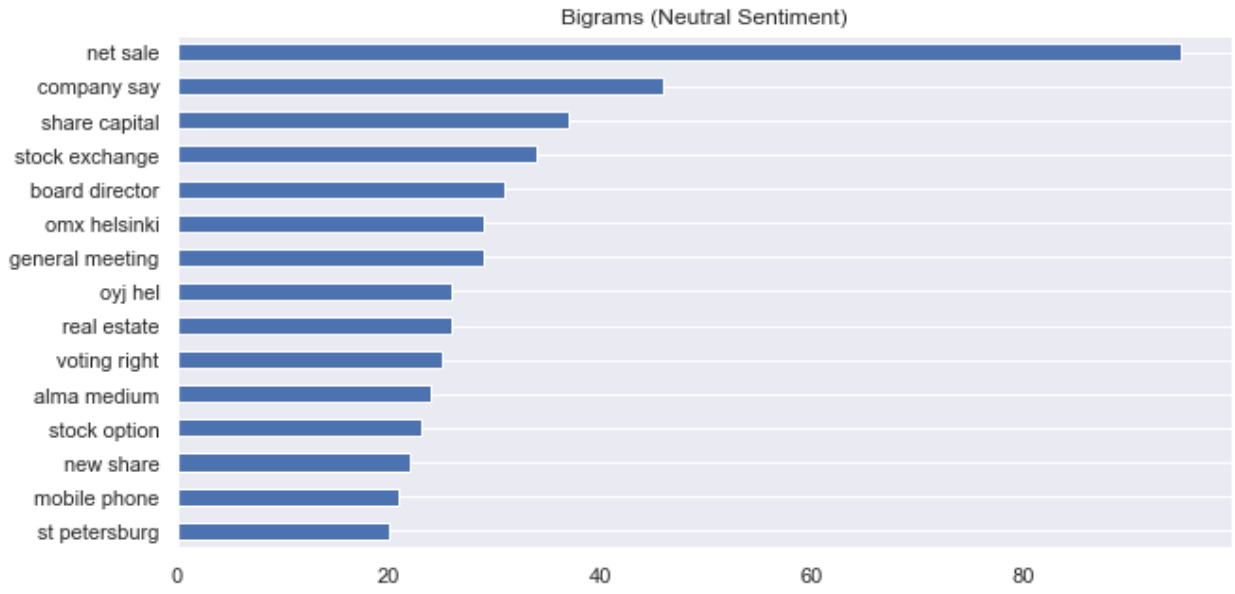
A word cloud was generated for easy visualization of the 500 most prevalent words throughout our 5003 headlines. This corroborates what is seen in the above barcharts, albeit this word cloud represents the aggregate of all three sentiment classes.

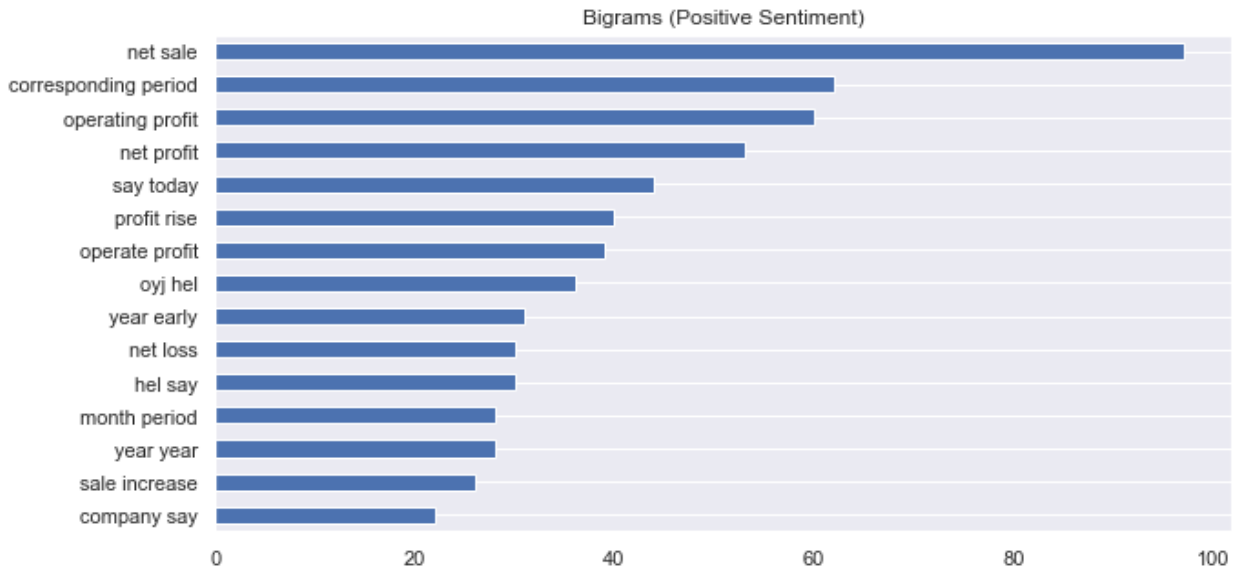


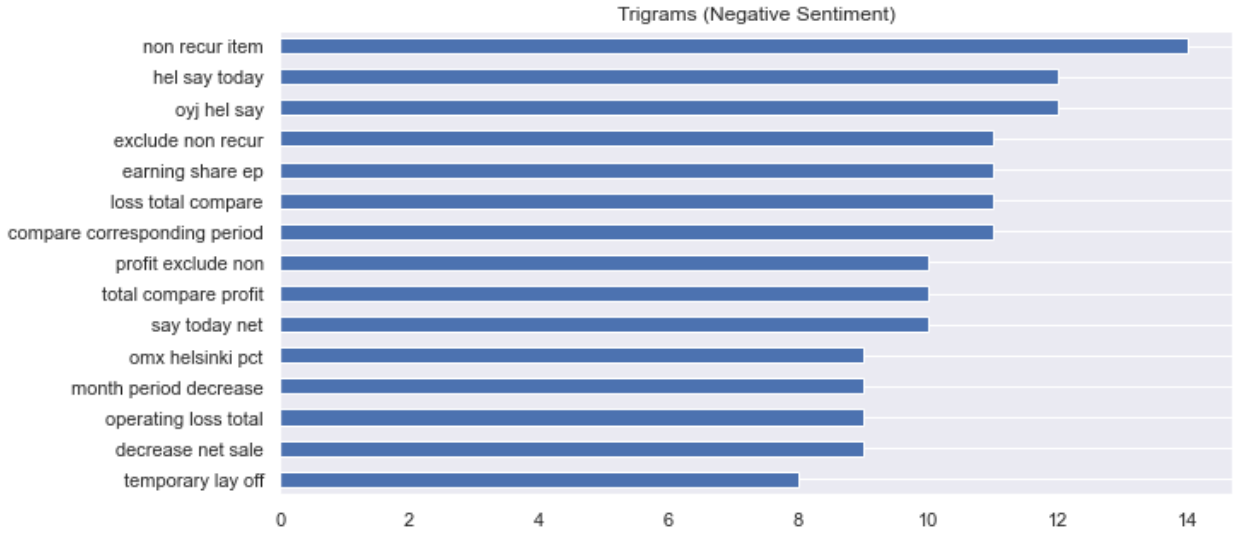
N-Grams

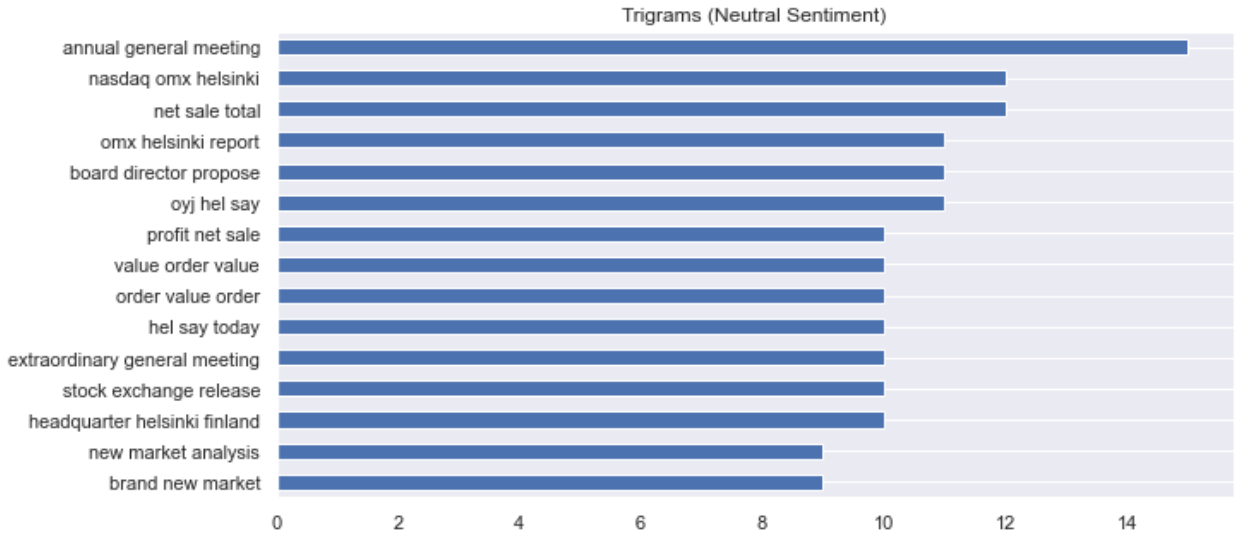
For this final portion of our EDA, we decided to implement N-Grams, and discover the most common short phrases in our data, to go beyond just looking at individual words. This information should prove useful, as it allows us to see some key phrases our model should consider when determining the headline's sentiment. For example, if we see a phrase is sufficiently correlated to a sentiment classification, then it would be useful for our model. We've provided bar graphs for bigrams and trigrams for each sentiment classification below:

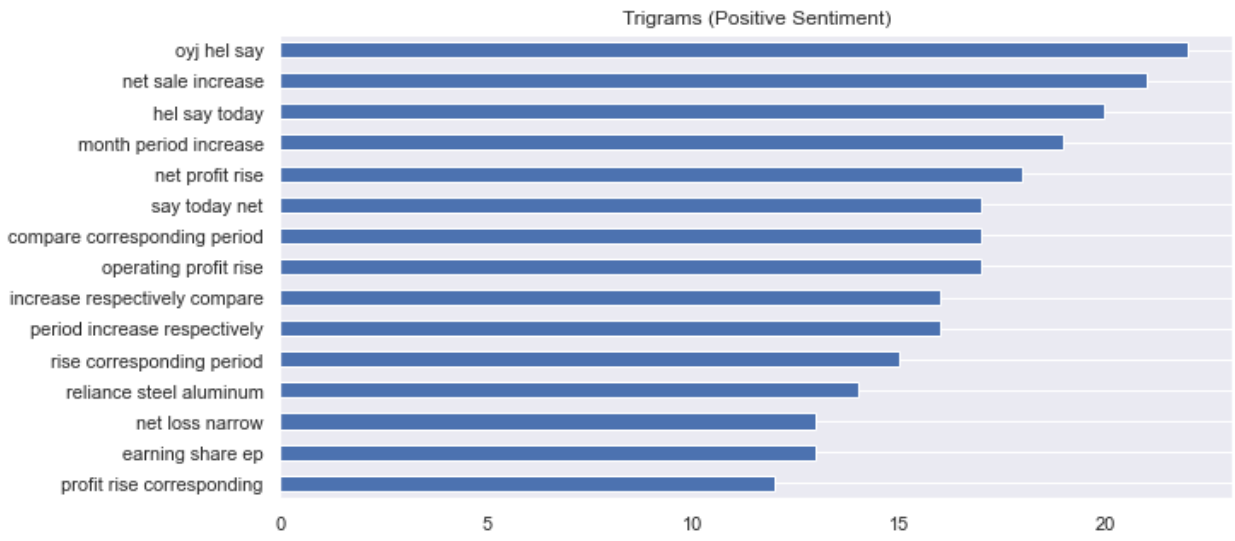



Bigrams (Neutral Sentiment)
net sale
company say
share capital
stock exchange
board director
omx helsinki
general meeting
oyj hel
real estate
voting right
alma medium
stock option
new share
mobile phone
st petersburg
0
20
40
60
80


Bigrams (Positive Sentiment)
net sale
corresponding period
operating profit
net profit
say today
profit rise
operate profit
oyj hel
year early
net loss
hel say
month period
year year
sale increase
company say
0
20
40
60
80
100


Trigrams (Negative Sentiment)
non recur item
hel say today
oyj hel say
exclude non recur
earning share ep
loss total compare
compare corresponding period
profit exclude non
total compare profit
say today net
omx helsinki pct
month period decrease
operating loss total
decrease net sale
temporary lay off
0
2
4
6
8
10
12
14


Trigrams (Neutral Sentiment)
annual general meeting
nasdaq omx helsinki
net sale total
omx helsinki report
board director propose
oyj hel say
profit net sale
value order value
order value order
hel say today
extraordinary general meeting
stock exchange release
headquarter helsinki finland
new market analysis
brand new market
0
2
4
6
8
10
12
14


Trigrams (Positive Sentiment)
oyj hel say
net sale increase
hel say today
month period increase
net profit rise
say today net
compare corresponding period
operating profit rise
increase respectively compare
period increase respectively
rise corresponding period
reliance steel aluminum
net loss narrow
earning share ep
profit rise corresponding
0
5
10
15
20