Harry Potter Analysis

Ashish Batra, Arjun Berry, Chen-Fan (Annie) Lo, Kunal Wagh, Xi Chen

Date: Dec 14th 2016

Course: MSBA 6330

Section: Morning

Contents

Introduction

Harry Potter is one of the best-selling books, having millions of loyal fans around the world. With numerous characters, locations, spells, and curses, it is interesting to explore the magical world of Harry, using the text provided in the series, to see the change in statistics and relationships between various characters. This gives the direction to the story and the events that happen in this series. The 7 books contain over 1 million words. To process this, big data tools were ideal for the analysis. This report will first discuss the methodology and procedures that we used for text analysis in Python and Spark, then be concluded by interesting insights.

Objective

To discover interesting insights from all Harry Potter books, the following analysis was performed in Apache Spark and Python:

1. Find the top 20 frequent words across the series.
2. Find the change in frequency of specific words, including character names, locations, spell names, and curse names, across the series
3. Find top N grams in across the series
4. Perform sentiment analysis to compare positive and negative sentiments among 7 books
5. Create informative visuals

Data Description

Data includes 7 text files of Harry Potter books (Chapters 1 to 7). **(Note 1)**

Methodology

Tools that have been used (Note 2)

- Python

We used Natural Language Toolkit (NLTK) to remove all the stopwords in our books. NLTK is used to deal with human language data. It has text processing libraries including tokenization, stemming, tagging. We used the functions including WordNetLemmatizer().lemmatize() and stopwords.words('english') to pre-processing our data.

- Spark

We used the power of Resilient Distributed Data or RDD part of Spark. As we wanted to perform computation on our data in a parallelized way to make it efficient, we read our file as RDD using spark context object. We performed various operations on this RDD like filter and map amongst others to transform RDDs and also, used actions to get values. We found (key, value) pairs for various analysis questions like count of specific words or count of words across chapters. As RDD uses cache memory to perform operations efficiently and quickly, we used cache function.

Procedures

1) Data pre-processing

The following procedures were performed to transform the data into a **bag of words** in RDD:

- Split sentences by “.” (full stop)
- Removed end of line (\n) characters from the text
- Split the text by whitespace to break into words
- Normalized the words to lowercase.
- Removed all punctuation using regular expression

- Lemmatize most of words: we used the function named WordNetLemmatizer().lemmatize() in NLTK to lemmatize most of words. So noun plurals will be changed to singular forms, and suffixes were removed.
- Removed all stopwords: stopwords like the, and, of and on are very common in so they are typically removed (Provost & Fawcett, 2013). We used the function named stopwords.words('english') to remove stopwords in our books.

2) Count all words and unique words in seven books

After loading our text files into RDD and removing punctuation, we used count() and distinct.count() to calculate the total words and unique words, respectively. The purpose of the step is to show, if Harry Potter books have more words and involve more characters, as it becomes significantly more popular.

3) Most frequent words in seven books

Using the cleaned data, we mapped the words to key-value pairs, aggregated by key, and then took the most frequent 20 words for all the seven books. For example, from book 1 to book 7, the most frequent words was “harry”. We further shortened the list by including only the top 10 words, and divided their frequency by the total word count of the books, in order to effectively compare the frequency of these words across 7 books.

4) Count specific words in seven books

After we obtained the word frequency from 7 books, we turned key-value pairs into Dataframe via pandas library (pd.DataFrame). Alternatively, we also used sqlContext to get the same Dataframe. We have counted the frequency of character names, locations, spell names, and curse names as follows. Bi-gram were used to count the frequency of two-word vocabulary.

a. Character names:

- Harry, Ron, Hermione, Hagrid, Malfoy, Snape, Dumbledore, Voldemort

b. Location

- Privet Drive, Diagon Alley, Hogwarts, Azkaban, Grimmauld Place

c. Spell names

- Expecto Patronum, Accio, Expelliarmus, Stupefy, Lumos

d. Curse names

- Imperio, Crucio, Avada Kedavra

5) N-Gram

To find the N-grams, we first needed to get the text splitted based on full stop. This gave us a list of lists of sentence. We then normalized the texts to make it consistent by bringing them to lower text. Next, following the standard procedure of stemming the words was something inevitable we had to do. However, as we were dealing with text here from a series of books, we skipped the stemming or lemmatization of words. This was followed by removal of stop words. This provided us with a bag of words. This bag of words was the right choice for performing n-grams. Next we used the nltk package which has functions for both bigrams and trigrams that we can calculate n-grams. This was followed by the use of the FreqDist function which gave us a key value pair, containing the key as the trigram or bigram, and value as the count of that bigram or trigram.

6) Sentiment Analysis

While we obtained insights by using N-grams, top frequency words, we wanted to check the sentiment across the harry potter series, as we wanted to identify and determine particular attitude and how it changed within chapters and throughout books.
We calculated the sentiment as positive, negative or neutral value. We used VADER(Valence Aware Dictionary and Sentiment Reasoner) package in python for sentiment analysis which is a simple rule based model and utilizes a general sentiment

lexicon following some basic rules of grammar, punctuation, and syntax and does not require an extensive data set to train, yet performs well across different domains.

a) Analyzing the Chapter Names

- We first extracted chapter names from each book and checked sentiment for that
- Since, the chapter names contained very few words, sentiment of most of the chapter names were neutral
- After this, the analysis was performed on the first few lines of the chapter

b) Analyzing the start of Each Chapter

- By looking at the first few lines, we wanted to determine the level of excitement each chapter creates in the mind of the reader with the first few lines. Whether it is dark and negative, or looks at the happier things
- We looked at the positive and negative sentiment index of the first few lines of each chapter in the whole series
- After importing Vader, we can import the list of sentences of starting of chapter. The sentiment is calculated using the vaderSentiment() method
- The vaderSentiment() method returns the amount of positive, negative and neutral sentiment values
- This analysis helped us to learn about the pattern of Harry potter series in terms of positivity and the negativity

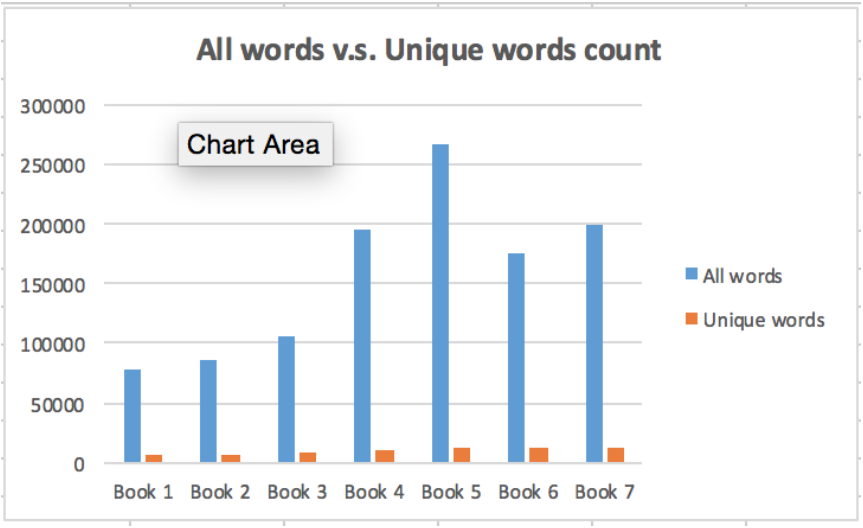
Please refer to Sentiment.ipynb for the code related to Sentiment by chapter names and starting of the chapter across whole series

Insights

1) Change in total and unique words

The following table and graph shows, the total and unique word count, which gets raised significantly after Book 4, as Harry's story moves into a mature stage. For example, Book 4 marks the presence of Voldemort. Book 5 has the highest number of words, since Dumbledore's Army or Harry and his associates started to join the forces against Voldemort and love story between Ron and Hermione starts to develop.

	Book 1	Book 2	Book 3	Book 4	Book 5	Book 6	Book 7
All words	78,444	86,420	106,073	195,634	266,418	175,799	199,348
Unique words	6,037	7,171	7,751	10,805	13,100	11,833	11,883



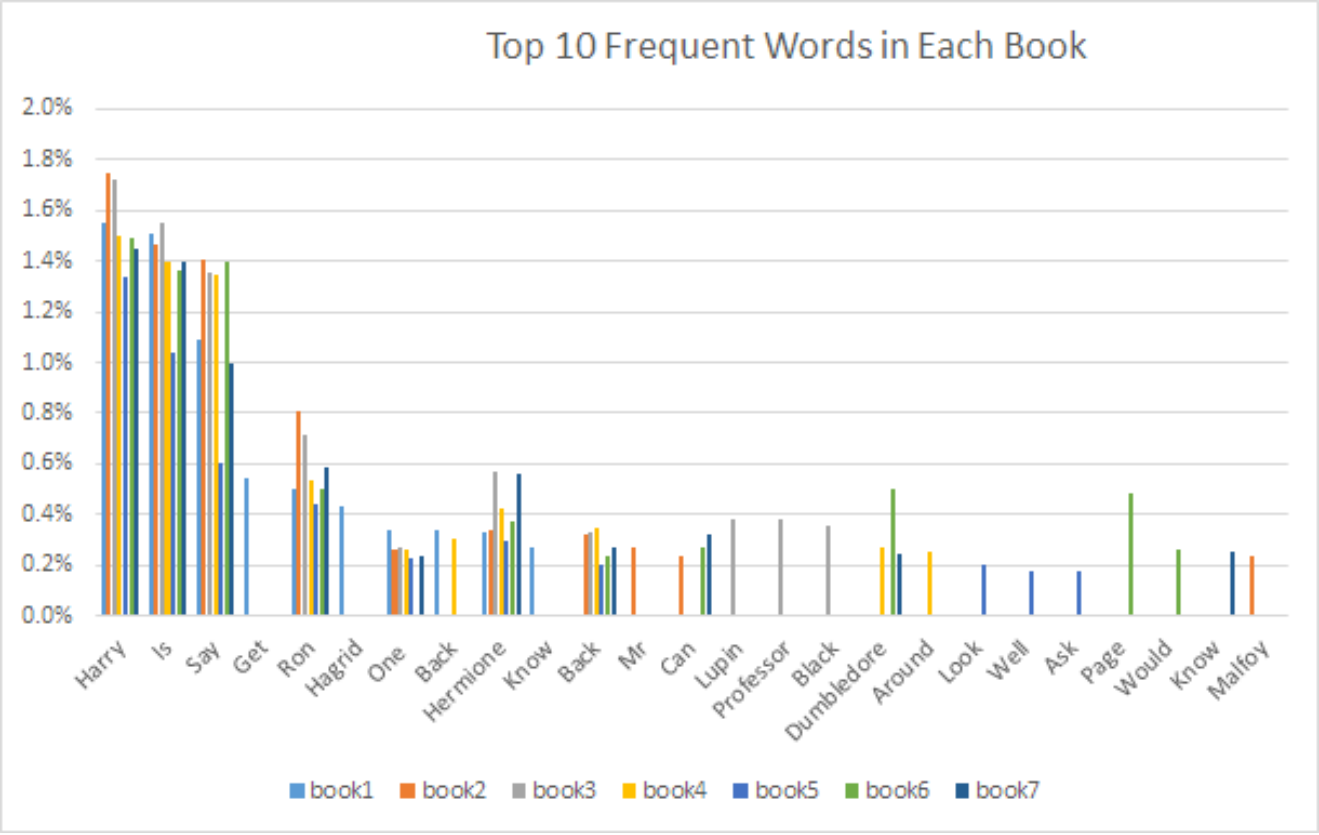
2) Change in top 20 word frequency

In our top 20 word lists **(Note 3)**, all the words are very simple, such as was, said, one, back, know, get, etc. In 5 out of 7 books, the most frequent nouns are the three young leading characters' name: Harry, Ron and Hermione. However in Book 1, Hagrid is

more important than Hermione. While in Book 6, Dumbledore is a leading character because he died in this book.

For some reason, some of the words didn't appear correctly in our result lists. For example, words 'wa' and 'uwa' came from words 'was' or 'u'was'' after removing 's'. The method also failed to combined words like 'got' and 'get'.

Then we got the correct version of the top 10 words in each book, and divided the result by the total words of that book to compare frequency of these popular words among 7 books. 'Harry' is definitely the superstar in our book. Simple verbs, including 'is' and 'say', are also very frequent among all the books. 'Ron' is a little more popular than 'Hermione' especially in the first and second book.



3) Specific word counts

a. Characters

Harry, Ron, Hermione are the main characters of Harry Potter, so they show up the most compared to other characters. We also found that Dumbledore appearance increased since Book 4, as the story began to involve more and more confrontations with the dementor forces. Voldemort also is more popular throughout the series, as he is the main villain and has many important fights with Harry in the later stage.

- Word frequency in percentage

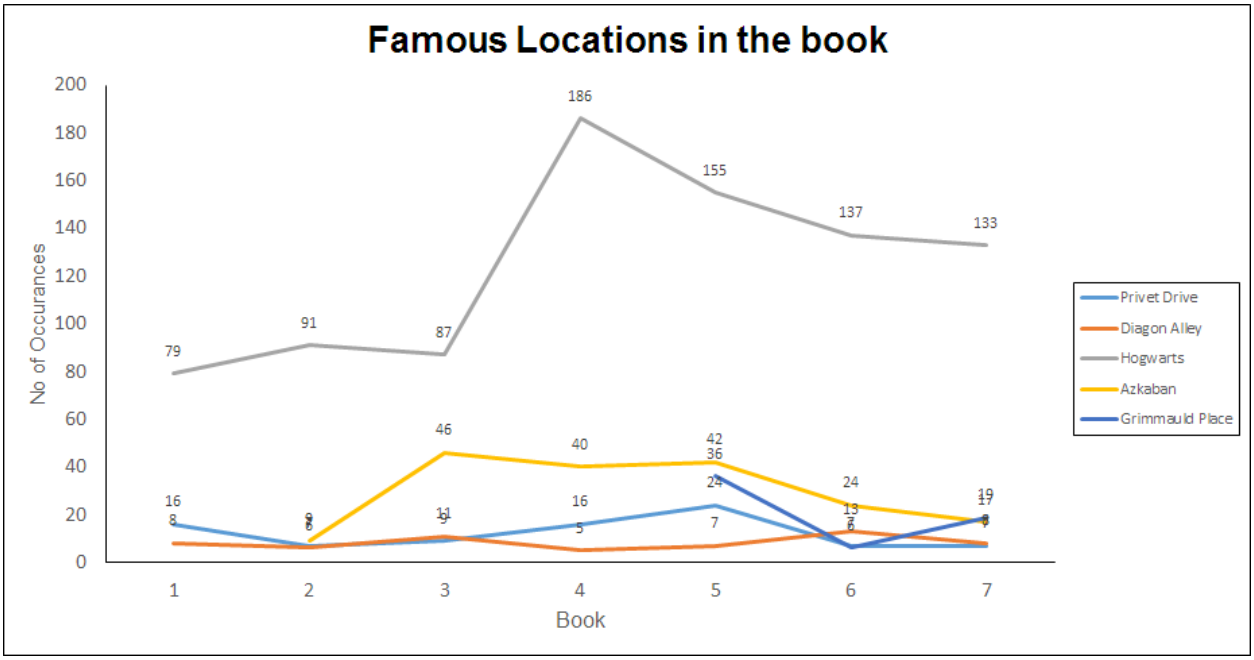
	1	2	3	4	5	6	7
Harry	1.55%	1.74%	1.72%	1.50%	1.41%	1.49%	1.45%
Ron	0.52%	0.76%	0.65%	0.50%	0.45%	0.45%	0.53%
Hermione	0.33%	0.34%	0.57%	0.42%	0.46%	0.37%	0.56%
Hagrid	0.43%	0.15%	0.19%	0.17%	0.15%	0.11%	0.08%
Malfoy	0.14%	0.23%	0.14%	0.06%	0.07%	0.19%	0.04%
Snape	0.18%	0.10%	0.19%	0.11%	0.12%	0.19%	0.13%
Dumbledore	0.18%	0.16%	0.14%	0.27%	0.21%	0.50%	0.24%
Voldemort	0.04%	0.02%	0.03%	0.09%	0.06%	0.11%	0.18%

b. Location

Different places have different stories, and thus we can imagine us being at these places when we read the book. We are familiar with the path to Diagon Alley and we also know how terrifying the Prison of Azkaban is.

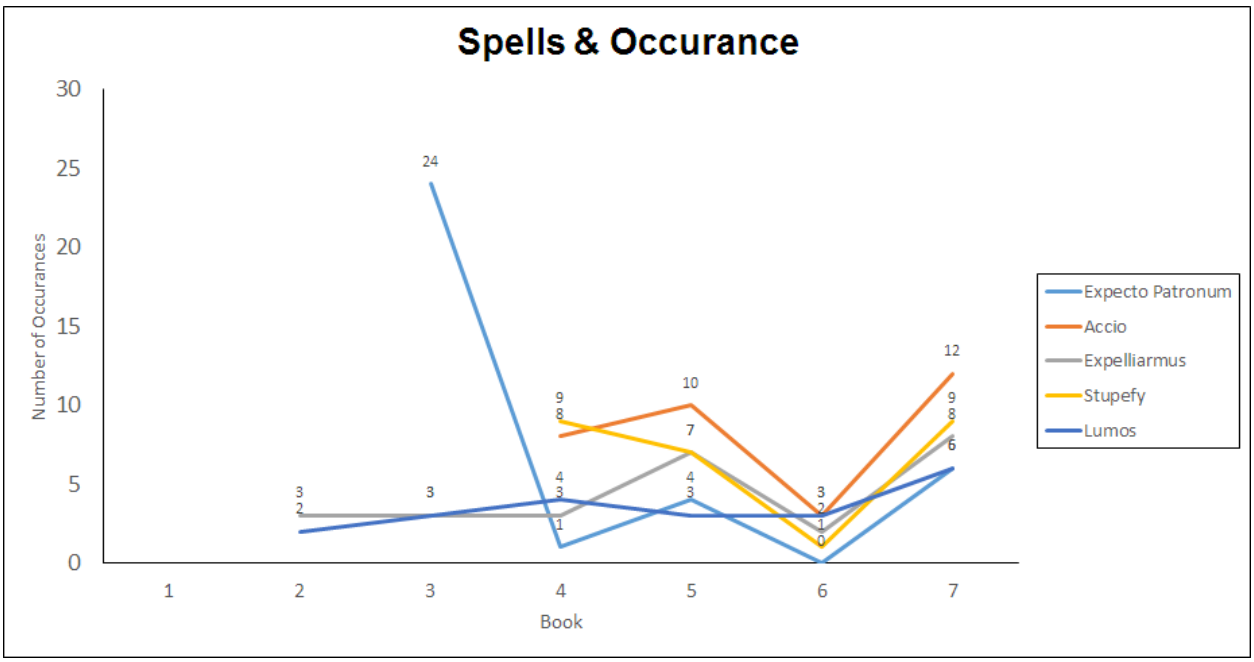
We looked at the occurrence of different locations throughout the books. Hogwarts, Private Drive and Diagon Alley are all places mentioned in all 7 books. Hogwarts is the most popular place as plenty of stories happened there. After Azkaban first appeared in Book 2, it becomes second popular place from Book 3 to Book 6, as Sirius Black

escapes the Prison. In Book 5, various new locations such as Grimmauld Place are added, contributing to more sentences in the book.



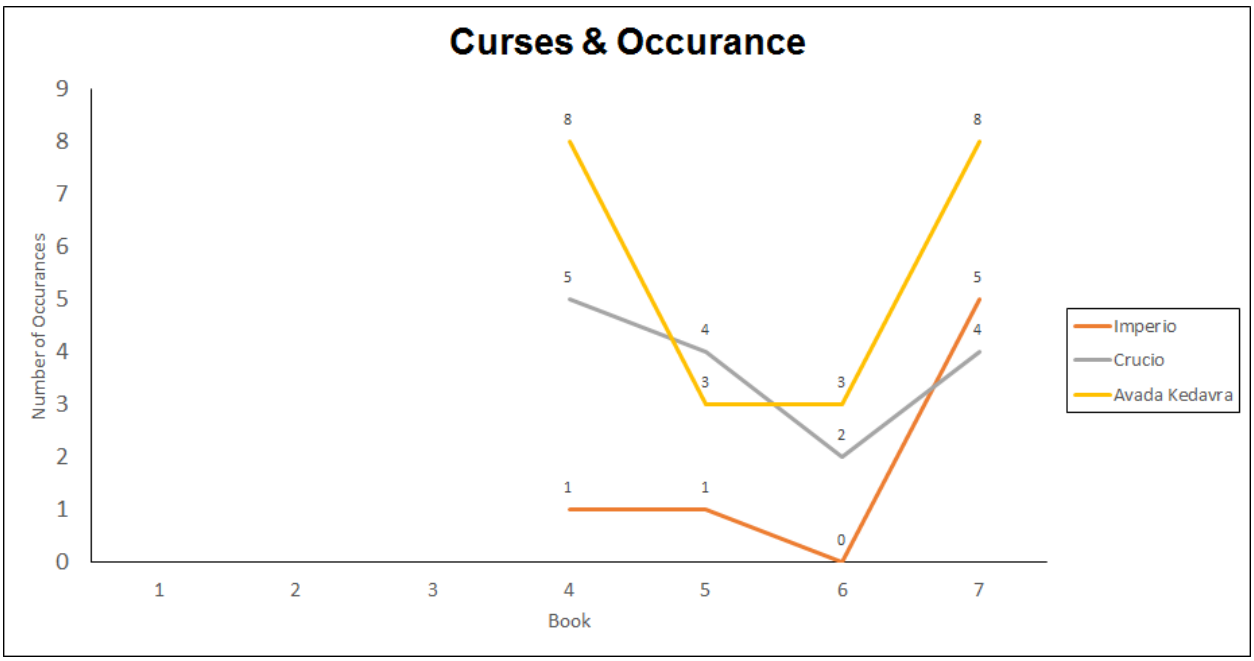
c. Spells

Harry Potter books are all about magic. Spells make the series interesting. From this graph, we can see how frequent the spells show up in the book. From Book 2, characters start to cast these famous spells. In Book 3, Expecto Patronum appears 24 times, a spike from others. After a deeper research, we found that in the chapter named Dementor's Kiss, it appeared 16 times. From Book 4, all these spells appear more often, alluding towards the enhanced abilities of our leading characters to cast them.



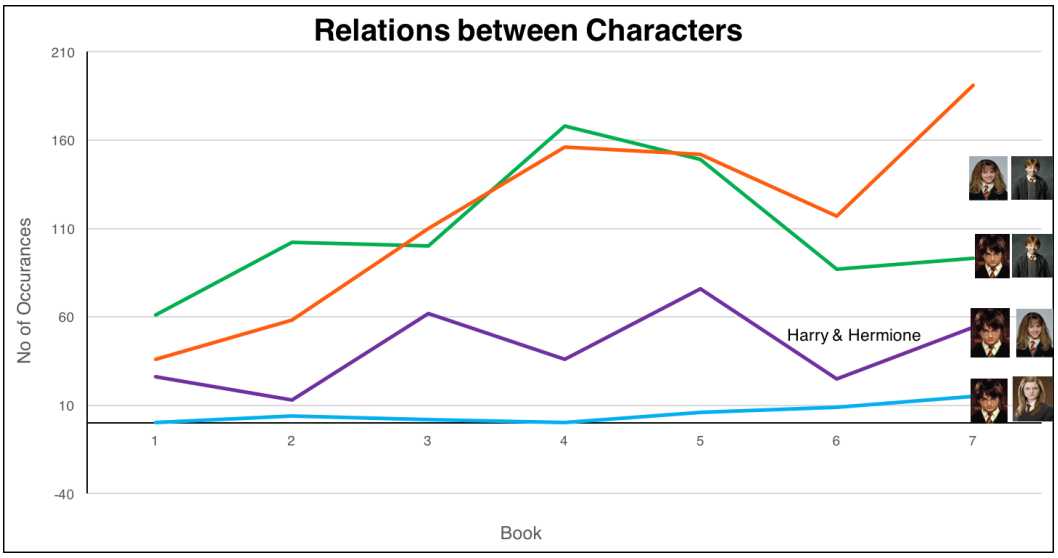
d. Curses

Along with the spells, curses go hand in hand. We can see that the wizards learned spells first and curses later. Curses are meant to be used for superior purpose. With the advent of story line, the usage of curses came from Book 4 where they actually were used in fights and later in wars.



4) N Gram

The results of N-grams were very intuitive. We found that when we removed the stop words, we found associations between words which made sense to us. For example, we saw Harry and Ron made up a bigram, and so did romantic relationship between Hermione and Ron which eventually led to their marriage. Also, with stopwords we wanted to be assured that this trend was true and without removing stopwords we created trigrams. These trigrams reinstated our analysis by giving us trigram - “ron_and_hermione”. Some other bigrams like fred and george were also popular amongst the books as both of them were always together. Finally, it would make much more sense to visualize these in context of various books to understand how the relationships have been built across the books.



5) Sentiment Analysis Insights

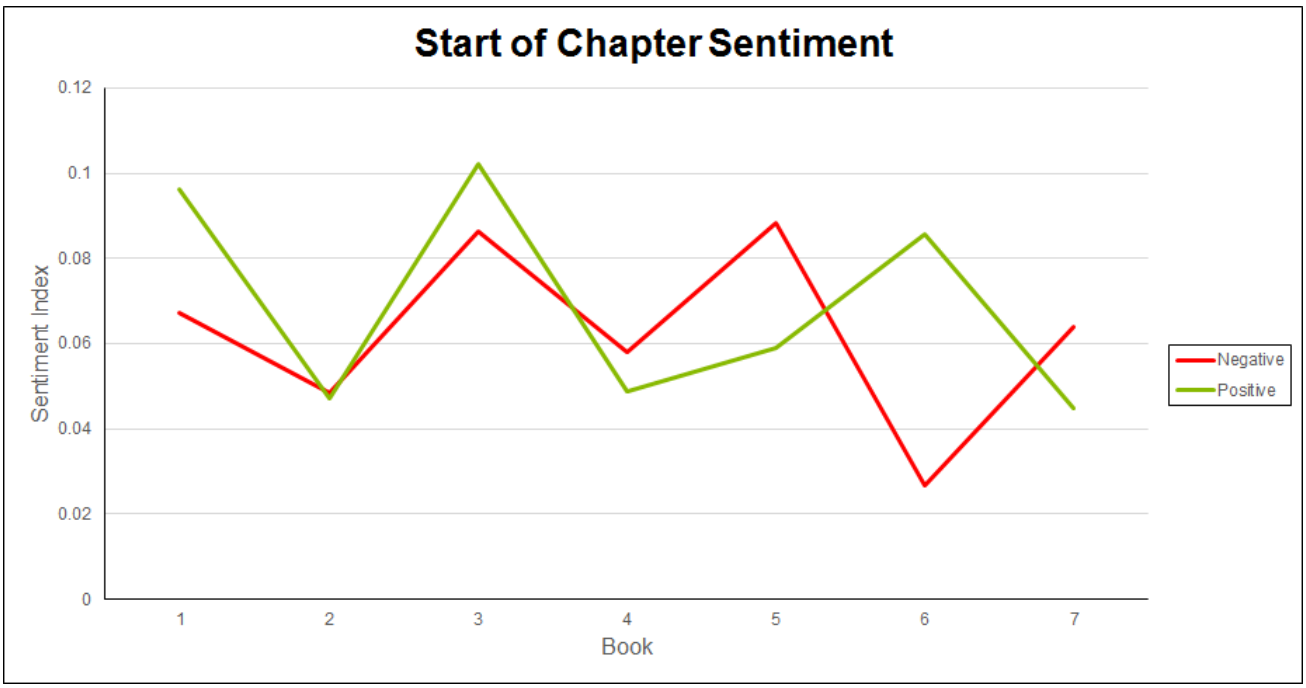
Sentiment Analysis helped us to analyze the writings of J.K.Rowling in the Harry Potter Series. It helped us to determine how and when she dwelled into the darker things and the pattern of the good and the bad things happening in the lives of people at Hogwarts. When we first looked at the chapter names only, most of the results were neutral because of less number of words and only a few chapters had positive or sentiment index.

When we looked at Chapter 20 from Book 3, we analyzed that chapter names don't present a clear picture. The sentiment based on the chapter name is :
Chapter 20: Chapter Name: The Dementor's Kiss
{'neg': 0.0, 'neu': 0.417, 'pos': 0.583, 'compound': 0.4215}

The chapter is related to dementor's sucking good memories from Harry and Sirius. It is more negative whereas based on our analysis, the index was positive because of the word kiss. As the results had limitations because of the word count, we analyzed the sentiment of the first few lines of the text in each chapter of the book. This gave us the sentiment as:
Chapter 20: By Starting of Chapter :
{'neg': 0.243, 'neu': 0.644, 'pos': 0.114, 'compound': -0.3818}

The result was more negative and most of the chapters had accurate predictions about its starting when we looked at the first few lines of the chapters. Also, we tried to analyze the pattern across the series and found some interesting insights which are shown below.

As we look at the graph for the whole series, we can see that till Book 4, the positive and negative sentiments are positively correlated and go hand in hand whereas from Book 5, the positive and negative sentiments have different trends. In addition, the series becomes more negative in the last books.



Limitation

- One of the limitations of Sentiment Index was that the package could not understand the meanings of words like mudblood and patronus and considered it as neutral. Mudblood which is negative and patronus which is positive in the context of the series were not represented as part of sentiments in the chapters. This is due to the fact that these words are fictional words only contained in Harry Potter literature and do not have particularly any meaning in English literature, which we assume the Vader library was built on.
- Other thing about text analysis in this scope is that, for the relationship graph, we have consider only names such as “Harry” or “Hermione”. We could get accurate picture when we consider the pronouns such as “He” or “She” to address the individuals

Future Scope

There is lot of opportunity to do more analysis using big data tools in text mining.

1) For all the characters, we can create a relationship map - of how many times they interacted with each other and with whom all they interacted.
2) We can perform topic modelling to see the prevalent topics or the theme of each books which are not obvious and also across the series. We can also find the genres of new books if not mentioned explicitly.
3) We can compare multiple novels or writings by multiple authors using the text mining.

Conclusion

This project gave us great opportunity to combine our passion of fantasy novels and text mining in big data. We worked on an enormous data of about million rows and spark was perfect choice for working with this data. We applied the concepts of text mining we learnt from the the text Data Science for Business in the course 6420 along with the big data tools learned in the course 6330 and generated interesting insights.

Before mining the data, we had a domain knowledge of what we want to achieve from this exercise. It involved the data preparations steps, data mining steps and insight generation steps. Most of the data preparation and data mining steps were performed in spark though we used python in some places.

The choice of using spark was wise, as we can scale the scope of analysis to more books using the scripts which we already have. The usage of Jupyter Notebook was also an advantage to us, as it gave better UI as well as step by step collection of results for testing purpose. Visualization of these insight generations, were done by plotting graphs in basic excel and tableau tools.

This project motivated us to look beyond the traditional usage of big data tools for analytical purpose. It helped to envision that using the right technology and context, we can envision our passion in analytics come true.

Data Source

https://www.stats.ox.ac.uk/~snijders/siena/HarryPotterData.html

https://cdn.preterhuman.net/texts/literature/fiction/fiction_books/Harry%20Potter/

Bibliography

Data Science for Business. Foster Provost and Tom Fawcett. 2013.

Supplemental Material

- Harry Potter Text Analysis Published by Zareen Farooqui
  - This published article has a lot of visualizations with applications of N-grams, Sentiment Analysis, Punctuation Analysis etc.
- A textual analysis of Harry Potter by an amateur data analyst by Ian Minoso
  - This published article is written by an enthusiatic amature analyst who has analyzed the text in the book using simple visualizations and techniques such as N-grams, word-count, sentiment analysis etc.

Remark

Note 1:

Harry Potter txt files of 7 books are attached in the Zipfile.

Note 2:

All codes are attached in the Zipfile, with names that refer to each section of the report.

Note 3:

- **Top 20 words in Book 1 - 7**

Book 1	Book 2	Book 3	Book 4	Book 5	Book 6	Book 7
harry	harry	harry	harry	harry	harry	harry
wa	Uwa	uwa	uwa	uwa	said	uwa
said	Said	said	said	said	uwa	said
ron	Ron	ron	ron	ron	dumbledore	ron
hagrid	hermione	hermione	hermione	hermione	ron	hermione
one	Back	lupin	mr	one	page	uwand
back	Mr	professor	back	back	hermione	could
hermione	One	black	dumbledore	looked	could	back
know	could	back	one	well	would	know
get	malfoy	one	around	asked	back	dumbledore
got	lockhart	like	looked	like	one	one
could	professor	around	like	yelled	well	like
like	Got	looked	could	malfoy	know	would
didn’t	Like	see	though	professor	like	looked
see	know	could	got	time	time	ueye
professor	Time	got	ueye	get	think	around
looked	around	snape	would	could	slughorn	think
snape	ueye	hagrid	uknow	wand	snape	voldemort
dont	Go	ueye	moody	would	looked	still
go	weasley	know	weasley	umr	dont	hand