# Introduction

Collaboration is a way of pooling the skills of individuals together to accomplish a task or project (add some reference). It has become increasingly more important in the modern world, as we have become more connected around the globe.

The purpose of this project is to gain insight into how we as individuals play a role in a collaborating on a data science project. As communication can be seen as a critical success measure, we wanted to analyse how we participated with each other.

To measure our participation, we collected data from chat conversations in Slack and JIRA issue tickets to get an understanding of how we collaborated with each other. We will be applying text mining techniques to transform and evaluate our data as well as using tools such as python and power BI to visualize and explain the insights that we have uncovered.

Text mining is being used increasingly as organizations recognize the untapped information contained in text. Social media, such as Twitter and Facebook, have been used effectively by organizations to uncover positive and negative trends that, when identified through text mining, can be used to leverage the positive trends and provide corrective action to counteract any negative comments. (maybe add reference)

#### Slack[¶](http://localhost:8880/notebooks/Desktop/DSI_Assignment/D-Matrix_Notebook.ipynb#Slack)

[Slack](http://localhost:8880/notebooks/Desktop/DSI_Assignment/www.slack.com) is an instant message and team collaboration tool. Some of the features of slack is being able to create channels with different topics. For example, in the case of our assignment we had channels based on the CRISP-DM framework as well as channels which were less about data science and more about free flowing discussions.

Some of the additional features of slack is the ability to incorporate customized addons. One of the addons we used was a messaging bot called Howdy. The purpose of howdy is to be an additional member of the team and allow people to reflect on what they had done as well as highlight any problems or issues occuring in the project.

**Jira**

[Jira](https://www.atlassian.com/software/jira) is an issue tracking and project management tool that allows you to track any kind of unit of work (issue, bug, story, project task, etc.) through a predefined workflow.

The item of work and the workflow can be highly customized for your team's specific requirements, whether simple or more complex.

Collaboration's also a big point of emphasis in JIRA - @mentioning, formatted commenting, and sharing issues via email all help make your work more visible to your teams so folks stay on the same page throughout their project, release, or set of tasks.

Some common use cases include software development, feature implementation, bug tracking, agile project management (with JIRA Agile), and service desk ticket tracking (with JIRA Service Desk).

## Part 2 Data Collection and Preprocessing

The data preparation stage is a critical stage of the data science project. Ensuring the quality of the data is essential before any analysis can be done. Accessing the data for this project was simple as Slack provided a way to extract the data into JSON file. As the majority of our data comes from messages, we spent the most amount of time formatting and transforming the text. To address these issues, we applied a number of Natural Language Processing techniques.

In this section, we will go through steps involved to transform the data as well as a background and evaluation of the techniques used in Natural Language Processing and our final decision of techniques that we used in this project. The main tools we will be using is Python Programming language with emphasis on the packages NLTK (Natural Language Tool Kit).

### Stopwords / Remove Words

<https://www.researchgate.net/publication/221254145_Evaluation_of_stop_word_lists_in_text_retrieval_using_Latent_Semantic_Indexing>

In every language, some words are particularly common. While their use in the language is crucial, they don’t usually convey a particular meaning, especially if taken out of context. This is the case of articles (e.g., a, an, the), prepositions (e.g., at, by, in, to, from, with) and conjunctions (e.g., and, but, as, because) which are commonly called stop-words (Zaman, Matsakis and Brown 2011). Stop word removal is one important step that should be considered during the pre-processing stage.

NLTK package provided a simple list of English Stop words which we used. However, we noticed that certain messages needed to be removed as they were automatically added during the setup of Slack. As such, we created our own custom list of words and messages to removed.

### Stemming and Lemmatizing

One of the issues that arises in natural language processing is that words have morphological variants which will not be recognised by term matching algorithms (Hull 1996). This means you could have words like organize, organizes, and organizing all have the same meaning but won't be grouped together. Some of the ways used to tackle this problem include Stemming and Lemmatizing

Stemming is a procedure which attempts to reduce a word to its base form by cutting the ends of the words off with the most common stemming algorithm being Porter’s stemmer.

Lemmatization on the other hand, uses vocabulary and linguistic analysis of words, with the aim to remove impure endings only and to return the base or dictionary form of a word, which is known as the lemma. (Manning et al. 2008).

Balakrishnan and Lloyed-Yemoh (2014) did study on comparing the performance of stemming and lemmatization for information retrieval. In their results they found that lemmatization outperformed stemming but the differences in accuracy were insignificant.

We tried out both techniques from NLTK package for this project and concluded that lemmatization was more suitable. This was due to stemming creating words that were unrecognisable and as such could not be analysed.

Refrerences

<http://www.lnse.org/papers/134-I3007.pdf>

Balakrishnan and Lloyd-Yemoh 2014,Lecture Notes on Software Engineering, Vol. 2, No. 3, August 2014

D. A. Hull, “Stemming algorithms: A case study for detailed evaluation,” Journal of the American Society for Information Science, vol. 47, pp. 70-84, 1996.

Jivani, A.G. & others 2011, 'A comparative study of stemming algorithms', Int. J. Comp. Tech. Appl, vol. 2, no. 6, pp. 1930–8. Karaa, W.B.A. 2013, 'A new stemmer to improve information retrieval',

International Journal of Network Security & Its Applications, vol. 5, no. 4, p. 143.

Manning, C.D., Raghavan, P., Schütze, H. & others 2008, Introduction to information retrieval, vol. 1, Cambridge university press Cambridge, viewed 8 May 2016,<http://www.langtoninfo.co.uk/web_content/9780521865715_frontmatter.pdf>.

Singh, S. & Pateriya, R.K. n.d., A Survey on various Stemming Algorithms, viewed 8 May 2016, <http://www.ijcert.org/V2I57.pdf>.

**Privacy / Anonymization of Names**

Privacy Identifies, contextualises, and reflects on the ethical, privacy, and legal issues relevant to the collection and analysis of personal data of self and others INTRUSIVE mystery box challenge privacy Privacy in conversation sharing as a contract between groups Privacy policies in software<https://slack.com/privacy-policy> <https://www.atlassian.com/legal/privacy-policy> <https://confluence.atlassian.com/doc/data-collection-policy-659783908.html>

(Need to add reference for why Privacy in data is important)

As our data contained the names of each member, it was important to anonymize everyone’s name. We decided to replace our names with the aliases Business Analyst, Data Scientist 1, Project Manager.

### Tokenizing

<http://airccse.org/journal/ijdms/papers/6614ijdms02.pdf>

Tokenization is the procedure to segregate all the words, numbers and characters in a given document and these identified words, numbers, and other characters are called tokens. In addition to the token generation process, it also evaluates the frequency value each token present (Singh and Saini 2014).

There are number of different types of tokenizers

* PunktWordTokenizer Can’t->[Can,’t]
* TreebankWordTokenizer (uses words in penn treebank corpus) can’t -> [ca, n’t]
* RegexpTokenizer o WhitespaceTokenizer Can’t->[Can’t] o WordPunctTokenizer Can’t-> [Can,’,t] Open Source Tools • Nlpdotnet Tokenizer • Mila Tokenizer • NLTK Word Tokenizer (mentioned above) • TextBlob Word Tonekize • MBSP Word Tokenize • Pattern Word Tokenize

Pantone, P. 2015, Adding Sentiment Analysis support to the NLTK Python Platform, viewed 8 May 2016, <http://project-archive.inf.ed.ac.uk/msc/20150231/msc_proj.pdf>.

Vijayarani, S. & Janani, R. 2016, 'Text Mining: Open Source Tokenization Tools - An Analysis', Advanced Computational Intelligence (ACII), vol. 3, no. 1, viewed 9 May 2016, <http://aircconline.com/acii/V3N1/3116acii04.pdf>.

Links<http://text-processing.com/demo/tokenize/> <https://www.ibm.com/developerworks/community/blogs/nlp/entry/tokenization?lang=en>

## Analysis

The remaining part of this paper will outline analysis completed and explores some of the questions we were trying to answer as well as interesting insights gathered.

**Term Frequency**

The initial analysis we conducted was exploring term frequencies to extract meaningful terms.

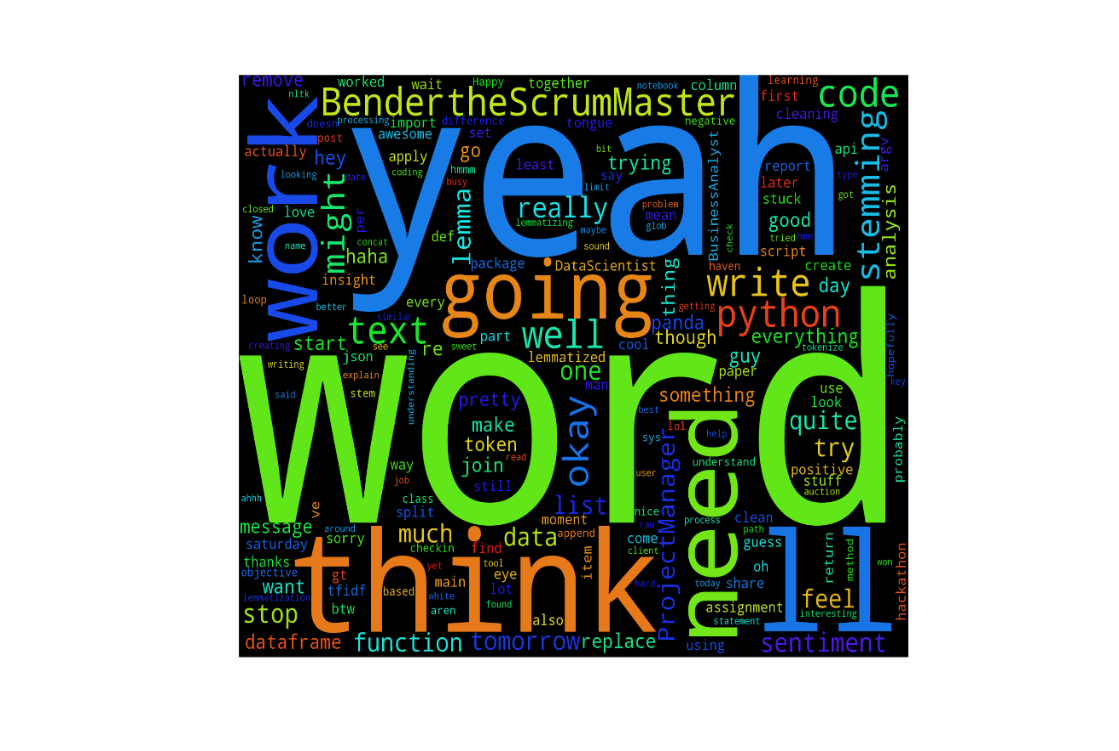
Using tokenization, we could aggregate each word and generate a list of the most frequent words used. Table 1 Indicates the top twenty words I used.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Word** | **Count** |  | **Word** | **Count** |
| word | 39 |  | Bender the Scrum Master | 17 |
| yeah | 39 |  | okay | 15 |
| think | 29 |  | text | 15 |
| like | 27 |  | well | 15 |
| data | 22 |  | code | 15 |
| hey | 21 |  | quite | 14 |
| work | 19 |  | feel | 14 |
| need | 19 |  | one | 14 |
| going | 19 |  | Business Analyst | 14 |
| might | 18 |  | Project Manager | 14 |

The table drew an interesting link to the members I engaged most with during the course of the project. Based on these results, I had the most communication with the Business Analyst and Project Manager. This could be a way of evaluating the communication within the group. It would be a way to highlight

Another insight, was looking at the language used in different time segments of the project. To achieve this the data was broken two parts, the first part was everything before the 28/04/2016 and the second part was everything after that date. This could then be visualized by generating a word cloud for both periods. It was interesting to see that there was a change in types of words being used in the first half (e.g think, start, might, discussion) compared with the second half (e.g work, need, yeah, write). You can tell there is a change from

Before After



## Sentiment Analysis

<https://indico.io/blog/sentimenthq-new-accuracy-standard/>

Indico, Aylien and Alchemy API's

<https://indico.io/> <http://aylien.com/> <http://www.alchemyapi.com/>

Twitter (2014) Twitter apis. [https://​dev.​twitter.​com/​start](https://dev.twitter.com/%E2%80%8Bstart).

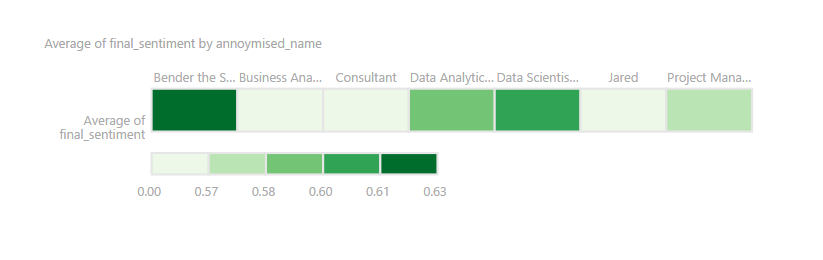
* <http://journalofbigdata.springeropen.com/articles/10.1186/s40537-015-0015-2>
* <https://www.cs.uic.edu/~liub/FBS/SentimentAnalysis-and-OpinionMining.pdf> (Liu, 2012)

Sentiment analysis is the study of people's attitude, emotion and opinion towards a particular product, service, event or even an individual (Liu 2012). Analysing sentiment has gained a lot of attention in recent years with many companies utilizing the data from social media to understand more about their customer and using this to improve their product or service.

The wide range of applications in almost every domain has led companies providing services to meet this demand. Some of the providers of these services include Indico, Aylien and Alchemy API. These services all provide an API (application programming interface) to connect to your data source.

We decided to use Indico for our analysis because it achieved the highest accuracy standard for sentiment analysis based on the 2015 article. In addition, there were 1000 daily limitations for Aylien.

look into the emotions of our text which was very relevant to our analysis. The below graph is a summary of some of the insights we uncovered.



**Conclusion**

​

In this paper, we wanted to

## Reflection

Some points

* We originally were applying the CRISP DM framework for data mining but found that it was difficult to apply for this task
* Cleaning was a hard task as there were lots of things we need to remove/adjust
* The recursive approach of analyzing work together
* 1942 Big Brother feel?

References

Agarwal, A., Xie, B., Vovsha, I., Rambow, O. & Passonneau, R. 2011, 'Sentiment Analysis of Twitter Data', Proceedings of the Workshop on Languages in Social Media, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 30–38, viewed 14 May 2016, <http://dl.acm.org/citation.cfm?id=2021109.2021114>.

Dai, A.M. & Le, Q.V. 2015, 'Semi-supervised Sequence Learning', arXiv:1511.01432 [cs], viewed 13 May 2016, <http://arxiv.org/abs/1511.01432>.

Kiritchenko, S., Xiaodan, Z. & Saif M., M. 2014, 'Sentiment Analysis of Short Informal Texts', Journal of Artificial Intelligence Research, vol. 50, viewed 11 May 2016, <https://www.jair.org/media/4272/live-4272-8102-jair.pdf>.

Maas L., A., Daly E., R., Pham T., P., Huang, D., Ng Y., A. & Potts, C. 2011, 'Learning Word Vectors for Sentiment Analysis', Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pp. 142–150.

Trim, C. 2013, The Art of Tokenization (Language Processing), viewed 13 May 2016,<https://www.ibm.com/developerworks/community/blogs/nlp/entry/tokenization?lang=en>.

Pantone, P. 2015, Adding Sentiment Analysis support to the NLTK Python Platform, viewed 8 May 2016, <http://project-archive.inf.ed.ac.uk/msc/20150231/msc_proj.pdf>.

Zhu, C., Tang, J., Li, H., Tou Ng, H. & Zhao, T.-J. 2007, 'A Unified Tagging Approach to Text Normalization', Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pp. 688–95.

In [ ]:

​

## Appendicies[¶](http://localhost:8880/notebooks/Desktop/DSI_Assignment/D-Matrix_Notebook.ipynb#Appendicies)

Need to add Cleaning Summary Table

Indico

Indico is helping individuals, small to medium sized teams and businesses translate their community’s pictures, documents and conversations into insightful feedback in minutes. Built with real life data and tailored to what you need, our pre-trained models balance accuracy and speed, allowing you to use powerful machine learning in realistic settings.

1) What is the indicio business model and user license?

The model is to build API endpoints that allow developers to rapidly prototype and deploy solutions within a predictive application. We love chatting with our users and are always looking to improve what we offer to better fit their needs. Staying up to date with the latest research papers is a huge part of our development process, thus ensuring all models are tuned to industry standards. We also have a private cloud offering for enterprise.

Aylien

Aylien’s text analytics API consists of eight distinct natural language processing, information retrieval, and machine learning APIs for article extraction, article summarization, classification, entity extraction, concept extraction, language detection, sentiment analysis and hashtag suggestions.

Alchemy API

AlchemyAPI uses natural language processing technology and machine learning algorithms to extract semantic meta-data from content, such as information on people, places, companies, topics, facts, relationships, authors, and languages.

API endpoints are provided for performing content analysis on Internet-accessible web pages, posted HTML or text content.