# 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 the language used whilst collaborating and the (need to fill). 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 (be it an 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.

On top of all that, JIRA's really good at tracking (via detailed, custom reports and dashboards) where all of your project items/issues lie at a teamwide, company or individual level - for example, what are all of the issues assigned to me, created in the last 7 days?

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 in any 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 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 how we transformed 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.

### Stemming and Lemmatizing

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

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.

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

**Copy from Duhita Document Need to Paraphrase**

Stemming is a process that chops off the ends of the words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes.

Lemmatization usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma. (Manning et al. 2008).

There are generally 2 types of errors that can error while performing stemming operations. Under Stemming could occur when the 2 words with the same interpretation are not stemmed to the same root. Over Stemming could occur when the 2 words with different interpretation are stemmed to the same root. (Jivani & others 2011).

Types of Stemming Algorithm are

• Truncating: It is a simple stripping technique which removes the affixes after the nth symbol. Different algorithms in this class are Lovins, Porters, Paice, Dawson

• Statistical: Methods which removes affixes after statistical procedures examples being N-Gram, HMM, YASSM • Mixed: approach to stemming and it involves both the inflectional as well as the derivational morphology consideration. Some of the mixed algorithms are Inflectional and Derivational, Corpus Based, Context Sensitive

As shown in the study “A Survey on various Stemming Algorithms” (Singh & Pateriya n.d.) that none of the above mentioned stemmers give 100% productivity but are satisfactory useful to the text mining, Natural Language Processing(NLP) or Information Retrieval(IR) applications.

The first one as in Porter stemmer handles inflectional morphology (plural, verb conjugation, etc.). The second step treats derivational morphology; it maps complex suffixes (suffixes compound of more than one suffix) to a single suffix from which they were derived (e.g. transform the suffix –istic to –ist). The third step deletes simple suffixes (uncompounded suffixes). The fourth step defines a set of recoding rules to normalize stems. The last step treats irregular forms that do not follow any pattern.

Refrerences

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>.

### Stopwords / Remove Words

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

Need to Paraphrase below

"""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, conjunctions, some adverbs, etc. which are commonly called stop-words. In the example above, we can see three common stop-words – to, and and on. Stop-word removal is one important step that should be considered during the pre-processing stages. One can build a custom list of stop-words, or use available lists (e.g. NLTK provides a simple list for English stop-words)."""

Bag of Words

<https://thesai.org/Downloads/Volume7No1/Paper_34-Enhancement_Bag_of_Words_Model_for_Solving.pdf><http://www.emis.de/journals/NSJOM/Papers/38_3/NSJOM_38_3_227_234.pdf>

### Tokenizing

Tokenization Tokenization is the process of segmenting text into words or sentences. It is a pre-processing step necessary for text mining.

The token’s use then is in its ability to be 1) Linguistically Significant 2) Methodologically useful The current best practice approach is to identify patterns with significant collocation rather than using delimiters on either side of a word Why is it important? Errors in this phase will significantly propagate that error into further phases The best tokenizers are custom ones for their specific domain. This is because issues of tokenization are language-specific. It thus requires the language of the document to be known.

Steps in Tokenization • Determine if two or more words should stand together (piece-meal) • Segment text into words • Handling Abbreviations o Common acronyms with punctuation o Common words containing periods • Handling hyphenated words o End of line hyphens (“okay great – but can you do this”) o True hyphens (lexical hyphen i.e. “Multi-faceted”, sententially determined hyphenation “case-based”) • Numerical and special expressions – use custom o Email o URL’s o Enumerations o Telephone numbers o Dates o Time o measures • Named Entity Extraction (“Business Analyst”) should be a single token – use custom dictionaries • English Enclitics – (‘s in she’s -> ambiguous can mean she has or she is)

  Types of Tokenizers • Naïve Whitespace Parser (split) • Apache Open NLP • Stanford 2.0.3 • Custom – BEST! (Use Regex!) • NLTK

* 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

NLTK Word Tokenizer good – general advice is to use a custom tokenizer based on regex for your specific domain language. Need to do morphological parsing if want to get to ideal state.

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>.

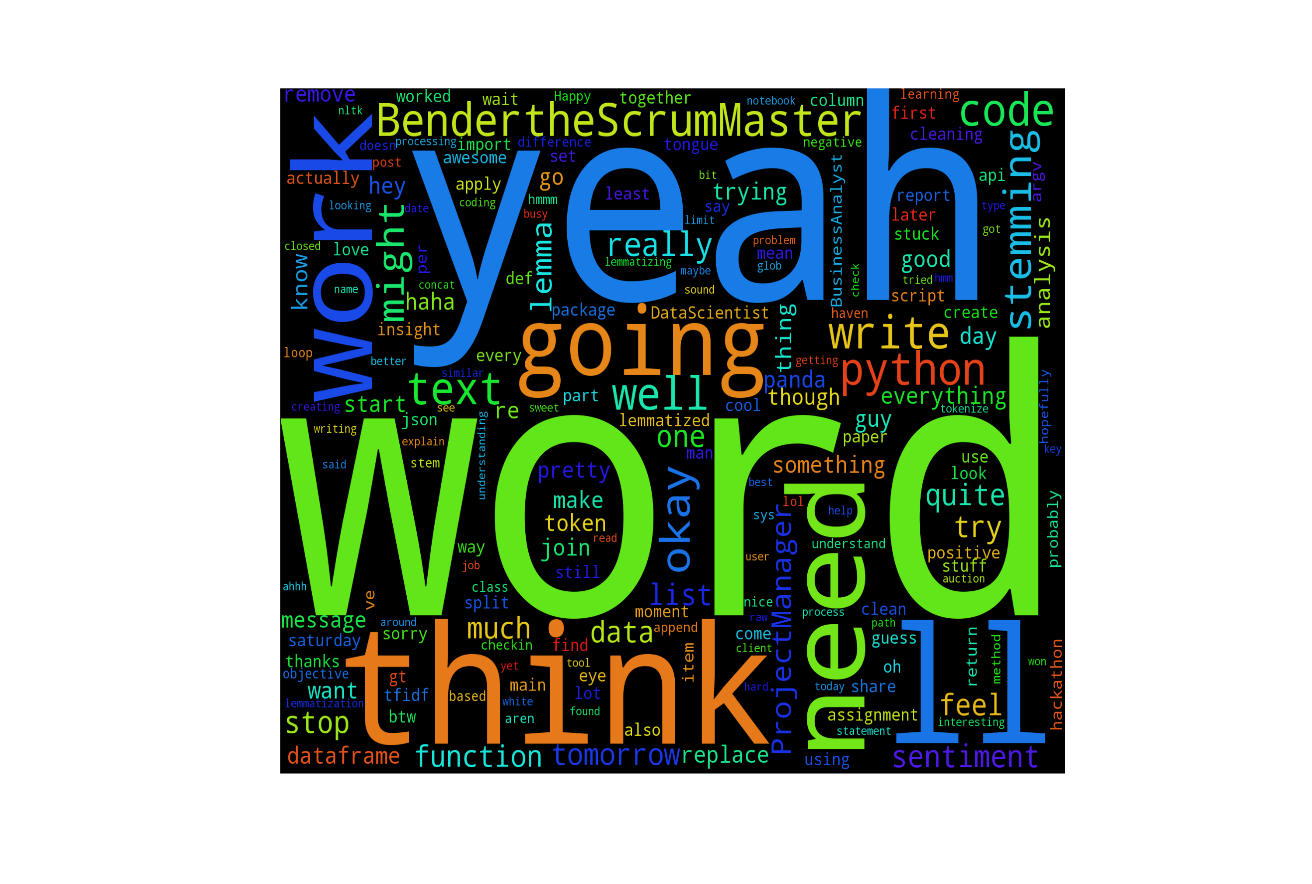
Links<http://text-processing.com/demo/tokenize/> <https://www.ibm.com/developerworks/community/blogs/nlp/entry/tokenization?lang=en>

## Analysis

For our analysis, one of the questions we were trying to answer was understanding the types of words we used to communicate with each other. As the words are now tokenized, it allowed us to aggregate each word and generate a list of the most frequent words used. Table 1 Indicates the top twenty word I used. The table drew an interesting link to what members







## Sentiment Analysis

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 which I will explain in more detail include Indico, Aylien and Alchemy API. I will also explain which one we choose for analysing our messaging data.

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.

We tested out all the above API's and decided to use Indico because it gave an indepth 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.

### Privacy

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>

​

## Conclusion

​

## 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

​

## References

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