
This workshop goes through the primary process of generating an LDA model with a highly simplified set of documents. This workshop is not an exhaustive explanation of text analysis, machine learning or LDA. The goal of this workshop is to guide analysts through critical steps in preparing their data, writing code and in providing an example of text analysis. It can benefit from interdisciplinary collaborations from AI, corpus linguistics and discourse analysis. The workshop emphasises research ethical approaches (Townsend and Wallace 2016) for dealing with large-scale text data.

Learning and Skills
The analyst will engage with the material presented here to develop an understanding on:

- How to write simple programs that can help the analysts manipulate and analyse language data (Bird et al. 2014).
- How key concepts from natural language processing and linguistics are used to describe and analyse language (Bird et al. 2014; Jurafsky and Martin, 2008).
- How to deal with the complexity of linguistic features and computational linguistic models (Bird et al. 2014; Mitkov, 2003).

The workshop looks broadly at how to combine discourse analyses (Gee, 1999), corpus linguistics (Gabrielatos et al. 2010; Baker, 2010) with artificial intelligence computational analyses (Blei et al. 2003, Blei and Smythe, 2017). AI includes latent Dirichlet allocation (LDA) topic modelling.

There are essential elements of text analysis and AI text analyses to think about:

- What can corpus linguist achieve by combining simple/AI programming techniques with large quantities of text?
- How can corpus linguist automatically extract key words and phrases that sum up the style and content of a text?
- What tools and techniques do the Python programming language provide for such research?
- What are some of the exciting challenges of natural language processing?

Why learn and use the Python Programming Language?
Python is relatively simple to use. It can help to see the text analysis in action. It is a powerful programming language with excellent functionality for processing linguistic data (Bird et al. 2014). Python is open source which means it can be downloaded for free from http://python.org/. Installers are available for all platforms from the Windows to the Linux operating systems. The workshop is using Windows.

For beginners to programming, the following program demonstrates some of the main features of Python. For example, whitespace is used to nest lines of code, thus the line starting with if falls inside the scope of the previous line starting with for; this ensures that the tion test is performed for each word. The program is a five-line Python program that processes file.txt and prints all the words ending in tion.
for line in open("file.txt"):
    for word in line.split():
        if word.endswith('tion'):
            print(word)

For advanced programmers, Python is an object-oriented programming OOP language where each variable is an entity that has specific defined attributes and methods. The value of the variable line is greater than a sequence of characters. It is a string object that has a ‘method’ (or operation) called split() that programmers can use to break a line of text into its words. To apply a method to an object, programmers can write the object name, followed by a period, followed by the method name, i.e. line.split(). Methods have arguments expressed inside parentheses. For instance, in the example, word.endswith('tion') had the argument 'tion' to indicate that the text analysts wanted words ending with tion and not anything else. An assumption is that Python programs are highly readable by humans, and it may be intuitive to guess what the program does even if someone had never written a program before.

What is LDA?
Jurafsky and Martin (2008) argue that ‘semantic fields and topic models are a very useful tool for discovering topical structure in documents.’ They use the example, of words that might be related by being in the semantic field of hospitals (surgeon, scalpel, nurse, anaesthetic, hospital). Latent Dirichlet allocation (LDA) (Blei et al. 2003) is an AI topic model that can generate topics based on word frequency from a set of documents. LDA is particularly useful for finding reasonably accurate mixtures of topics within a given document or sets of documents automatically - the artificial intelligence community refers to it as unsupervised learning.

Cleaning the documents
Data cleaning (Nitin, and Damerau, 2010) is crucial for generating a useful topic model. Stemming is merging words that are equivalent in meaning. Tokenizing is converting a document to its atomic elements. Stopping is where ‘meaningless’ words or Stop words are removed. However, this workshop suggests that stop words are significant in discourse analyses. LDA models should be done by removing stop words and by keeping stop words. Researchers may also use their stop word lists that keep for example pronouns. AI can consider certain parts of English speech, like conjunctions, (“for”, “or”) or the word “the” to be meaningless to a topic model.

The workshop will need an IT suite with Python 3.7; Anaconda, and Spyder. These are freely available software. Python needs to have the Text and Machine learning modules, for example, NLTK, NLTK-Data, NumPy, Scikit, Matplotlib and LDA (I can liaise directly with Cardiff University IT Department to get the most suitable ones installed).

Python text and AI analysis coverage

1. Basic High-Frequency feature extraction using text data
   a. Simple statistics
   b. Frequency distributions
2. Basic Data Cleaning and Text Pre-processing of text data
   a. Stopwords leave/removal
   b. Tokenisation, Stemming, Lemmatisation
3. Basic Introduction to Algorithms and AI LDA Text Processing
   a. N-grams
   b. Term Frequency-Inverse Document Frequency (TF-IDF)
   c. Bag of Words
   d. Latent Dirichlet Allocation LDA
This workshop should be an introduction and lead on to the exciting fields of sentiment analysis and subjectivity (or opinion mining) (Nitin and Damerau, 2010).

References and additional reading