Natural Language Processing

Topic Modelling

2023-24

terri.hoare@dbs.ie

Unsupervised Text Modelling

Introduction

Once documents are vectorised (using the same techniques that we explored for supervised learning (classification)), documents can be grouped (clustered using traditional clustering algorithms) with the resulting groups broadly describing the overall themes, topics, and patterns within the corpus.

The patterns can be discrete (when groups don't overlap) or fuzzy (data points belonging to all cluster groups with varying degrees of membership from 0 to 1).

Unsupervised Text Modelling

Case Study - Context

The case study we will use for unsupervised learning is based on a case study available online from Matthew North, Data Mining for the Masses (2012), pp. 201-226, licensed under a CC-BY-3.0 (opens in a new tab) license.

In the years preceding ratification of the Constitution of the United States in 1789, letters were published in two newspapers promoting ratification under a pseudonym, ‘Publius’. On the death of Alexander Hamilton in 1804, notes were discovered and some of the authors of the letters were identified from the style of the notes. These letters became known as the federalist papers. It was identified that John Jay had written papers 3, 4, and 5. James Madison had written paper 14, while Alexander Hamilton had written paper 17. It was suspected that paper 18 was a collaboration between Madison and Hamilton. An historian and archivist is tasked with using text mining to support the theory of a collaboration on paper 18.

Unsupervised Text Modelling

Case Study - Context

The case study we will use for unsupervised learning is based on a case study available online from Matthew North, Data Mining for the Masses (2012), pp. 201-226, licensed under a CC-BY-3.0 (opens in a new tab) license.

The data to be examined includes the full text of Federalist Papers numbers 3, 4, and 5 (Jay), 14 (Madison), 17 (Hamilton), and 18 (a suspected collaboration between Madison and Hamilton) (refer Moodle).

Unsupervised Text Modelling

Topic Modelling

Before exploring clustering techniques, we'll explore topic modelling, an unsupervised machine learning technique for abstracting topics from collections of documents. While clustering seeks to establish groups of documents within a corpus, topic modelling aims to abstract core themes; clustering is deductive, while topic modelling is inductive.

Methods for topic modelling, together with open-source implementations, have evolved significantly over the last decade. We'll compare three of these techniques: **Latent Dirichlet Allocation** (LDA), **Latent Semantic Analysis** (LSA), and **Non-Negative Matrix Factorization** (NNMF).

Unsupervised Text Modelling

Topic Modelling - LDA

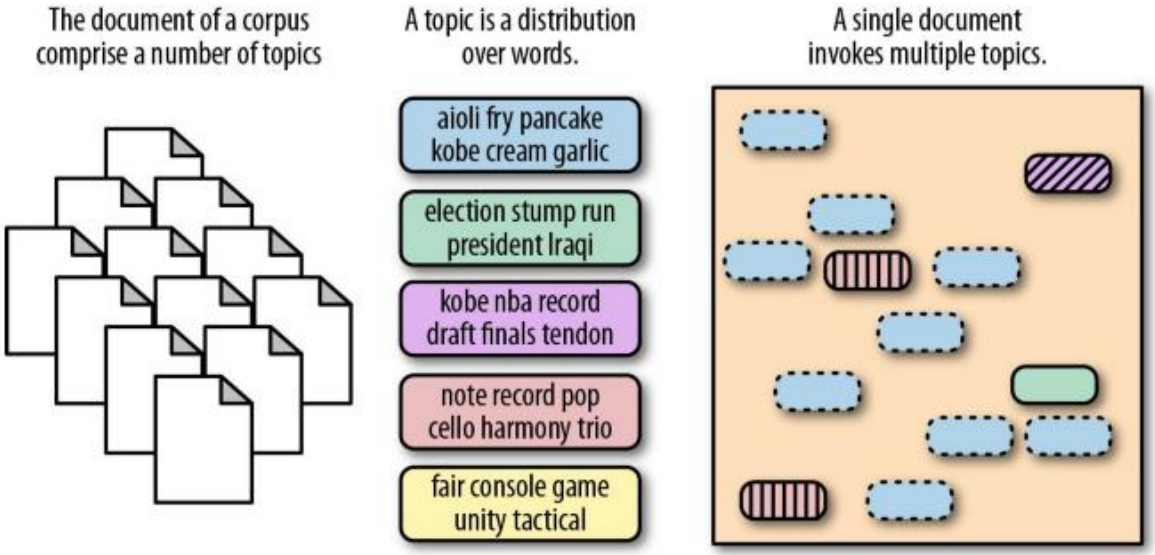
Introduced by David Blei, Andrew Ng, and Michael Jordan in 2003, **Latent Dirichlet Allocation (LDA)** is a topic discovery technique.

LDA belongs to the **generative probabilistic model** family, in which topics are represented as the probability that each of a given set of terms will occur. Documents can in turn be represented in terms of a mixture of these topics.

A unique feature of LDA models is that topics are not required to be distinct, and **words may occur in multiple topics**; this allows fuzziness that is useful for handling the flexibility of language.

Unsupervised Text Modelling

Topic Modelling - LDA



Latent Dirichlet Allocation (Bengfort et. al, 2019, p. 111)

Unsupervised Text Modelling

Topic Modelling - LDA

Blei et al. (2003) use the Dirichlet prior, a continuous mixture distribution (a way of measuring a distribution over distributions), to discover topics that occur across a corpus and in different mixtures within each document in the corpus.

Given an observed word or token, the probability of topics, the distribution of words for each topic, and the mixture of topics within a document can be modelled.

Scikit-Learn and Gensim include implementations of LDA.

```
from sklearn.pipeline import Pipeline
from sklearn.decomposition import
LatentDirichletAllocation
from sklearn.feature_extraction.text import
CountVectorizer

class SklearnTopicModels(object):

    def __init__(self, n_topics=50):
        """
        n_topics is the desired number of topics
        """
CountVectorizer

class SklearnTopicModels(object):

    def __init__(self, n_topics=50):
        """
        n_topics is the desired number of topics
        """
        self.n_topics = n_topics
        self.model = Pipeline([
            ('norm', TextNormalizer()),
            ('vect', CountVectorizer(tokenizer=identity,
                                     preprocessor=None,
lowercase=False)),
            ('model',
LatentDirichletAllocation(n_topics=self.n_topics)),
        ])
```

Topic Modelling

LDA – Scikit-Learn

A pipeline with TextNormalizer, CountVectorizer, and Scikit-Learn's implementation of LatentDirichletAllocation.

Specify the number of topics (50).

.

Unsupervised Text Modelling

Topic Modelling - LDA

```
    def get_topics(self, n=25):
        """
        n is the number of top terms to show for each
topic
        """
        vectorizer = self.model.named_steps['vect']
        model = self.model.steps[-1][1]
        names = vectorizer.get_feature_names()
        topics = dict()

        for idx, topic in enumerate(model.components_):
            features = topic.argsort()[:-(n - 1): -1]
            tokens = [names[i] for i in features]
            topics[idx] = tokens

        return topics
```

Each topic is inspected in terms of the words it has the highest probability of generating. The 25 highest weighted terms are ranked first. The topics are stored as a dictionary where the key is the index of one of the 50 topics and the values are the top words associated with that topic.

Bengfort, Benjamin; Bilbro, Rebecca; Ojeda, Tony. Applied Text Analysis with Python 2019

Unsupervised Text Modelling

Topic Modelling - LDA

```
if __name__ == '__main__':
    corpus = PickledCorpusReader('corpus/')

    lda       = SklearnTopicModels()
    documents = corpus.docs()

    lda.fit_transform(documents)
    topics = lda.get_topics()
    for topic, terms in topics.items():
        print("Topic #{}:".format(topic+1))
        print(terms)
```

Fitting and transforming the pipeline on the Hobbies corpus documents and unpacking the dictionary printing out the corresponding topics and their most informative terms.

```
Topic #1:
['science', 'scientist', 'data', 'daviau', 'human',
'earth', 'bayesian',
'method', 'scientific', 'jableh', 'probability',
'inference', 'crater',
'transhumanism', 'sequence', 'python', 'engineer',
'conscience',
'attitude', 'layer', 'pee', 'probabilistic', 'radio']
Topic #2:
['franchise', 'rhoden', 'rosemary', 'allergy', 'dewine',
'microwave',
'charleston', 'q', 'pike', 'relmicro', '($', 'wicket',
'infant',
't20', 'piketon', 'points', 'mug', 'snakeskin',
'skinnytaste',
'frankie', 'uninitiated', 'spirit', 'kosher']
Topic #3:
['cosby', 'vehicle', 'moon', 'tesla', 'module',
'mission', 'hastert',
'air', 'mars', 'spacex', 'kazakhstan', 'accuser',
'earth', 'makemake',
'dragon', 'model', 'input', 'musk', 'recall', 'buffon',
'stage',
'journey', 'capsule']
```

Unsupervised Text Modelling

Topic Modelling – Gensim

The Gensim implementation for Latent Dirichlet Allocation has some convenient attributes over Scikit-Learn. Gensim (starting with version 2.2.0) provides a wrapper for its LDAModel, called ldamodel.LdaTransformer.

To use Gensim's LdaTransformer, we need to create a custom Scikit-Learn wrapper for Gensim's TfidfVectorizer so that it can function inside a Scikit-Learn Pipeline. GensimTfidfVectorizer will vectorize our documents ahead of LDA, as well as saving, holding, and loading a custom-fitted lexicon and vectorizer for later use.

Unsupervised Text Modelling

Topic Modelling – Visualising

It s helpful to be able to visually explore the results of a model, since traditional model evaluation techniques are useful only for supervised learning problems.

The **pyLDAvis** library is designed to provide a visual interface for interpreting the topics derived from a topic model.

PyLDAvis works by extracting information from fitted LDA topic models as input to an interactive web-based visualization, which can easily be run from inside a Jupyter notebook or saved as HTML.

Bengfort, Benjamin; Bilbro, Rebecca; Ojeda, Tony. Applied Text Analysis with Python 2019

Unsupervised Text Modelling

Topic Modelling – Visualising

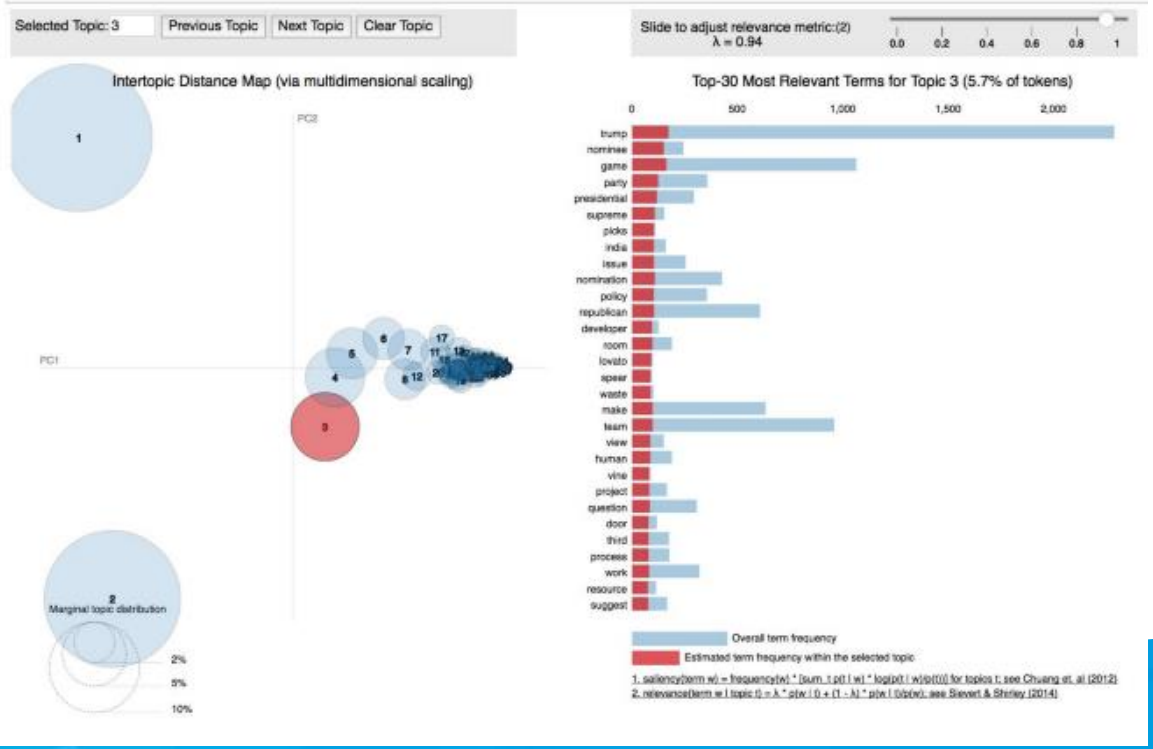
```
import pyLDAvis
import pyLDAvis.gensim

lda = gensim_lda.model.named_steps['model'].gensim_model

corpus = [

gensim_lda.model.named_steps['vect'].lexicon.doc2bow(doc)
    for doc in
gensim_lda.model.named_steps['norm'].transform(docs)
]

lexicon = gensim_lda.model.named_steps['vect'].lexicon

data = pyLDAvis.gensim.prepare(model,corpus,lexicon)
pyLDAvis.display(data)
```

pyLDAvis.gensim.prepare takes as an argument the LDA model, the vectorized corpus, and the derived lexicon and produces, upon calling display, visualizations like the one shown in Figure (next slide).

Bengfort, Benjamin; Bilbro, Rebecca; Ojeda, Tony. Applied Text Analysis with Python 2019

Unsupervised Text Modelling

Topic Modelling – Visualising



Interactive topic model visualization with pyLDAvis

(Bengfort et. al, 2019, p. 118)

Unsupervised Text Modelling

Topic Modelling – LSA

Deerwester et. al (1990) Latent Semantic Analysis (LSA) is a vector-based approach. While Latent Dirichlet Allocation works by abstracting topics from documents, which can then be used to score documents by their proportion of topical terms, Latent Semantic Analysis finds groups of documents with the same words.

The LSA approach identifies themes within a corpus by creating a sparse term-document matrix, where each row is a token and each column is a document. Each value in the matrix corresponds to the frequency with which the given term appears in that document and can be normalized using TF–IDF.

Unsupervised Text Modelling

Topic Modelling – LSA

Deerwester et. al (1990) Singular Value Decomposition (SVD) can then be applied to the matrix to factorize into matrices that represent the term-topics, the topic importances, and the topic-documents.

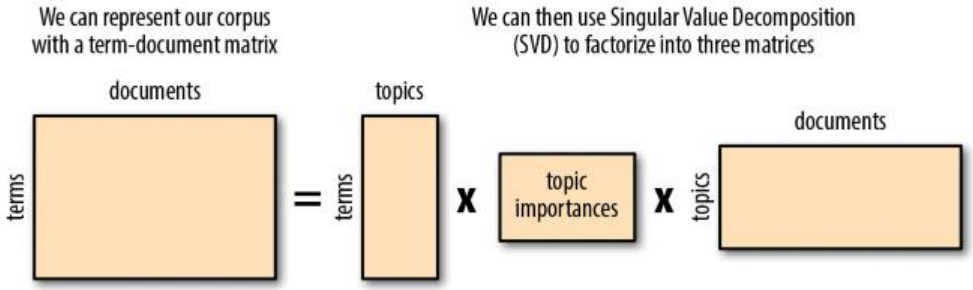
Note:- Both **PCA and SVD** are used in exploratory data analysis and machine learning. PCA is the most widely used. PCA is a special case of SVD. PCA needs the data normalized, ideally same unit. SVD doesn't need to compute the covariance matrix so it is numerically more stable than PCA. There exist rare cases where computing the covariance matrix leads to numerical problems.

Bengfort, Benjamin; Bilbro, Rebecca; Ojeda, Tony. Applied Text Analysis with Python 2019

Unsupervised Text Modelling

Topic Modelling – LSA (code on Moodle)

Using the derived diagonal topic importance matrix, we can identify the topics that are the most significant in our corpus and remove rows that correspond to less important topic terms. Of the remaining rows (terms) and columns (documents), we can assign topics based on their highest corresponding topic importance weights.



Latent Semantic Analysis (Bengfort et. al, 2019, p. 119)

Bengfort, Benjamin; Bilbro, Rebecca; Ojeda, Tony. Applied Text Analysis with Python 2019

Unsupervised Text Modelling

Topic Modelling – NNMF

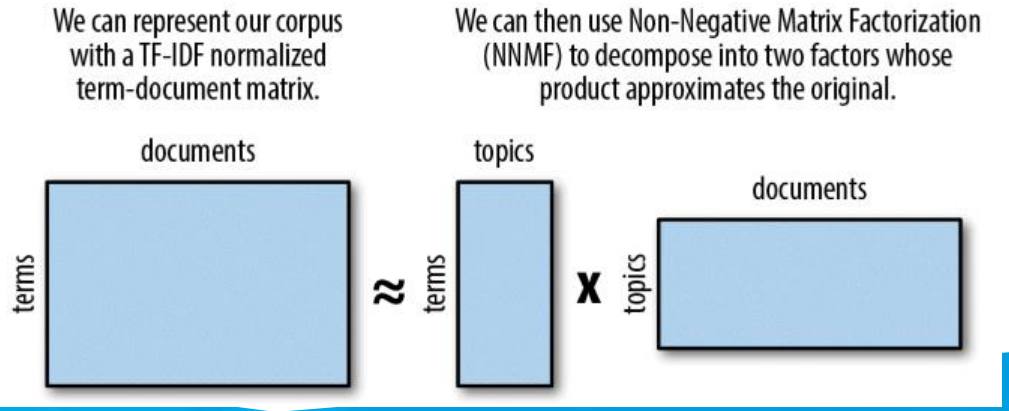
Paatero et.al (1994) Another unsupervised technique that can be used for topic modelling is non-negative matrix factorization (NNMF) popularized in a Nature article (Lee and Seung, 1999).

NNMF has many applications, including spectral data analysis, collaborative filtering for recommender systems, and topic extraction.

Unsupervised Text Modelling

Topic Modelling – NNMF (code on Moodle)

Using NNMF for topic modelling, the corpus is represented as a TF–IDF normalized term-document matrix. The matrix is then decomposed into two factors whose product approximates the original. The resulting matrices illustrate topics positively related to terms and documents of the corpus.



Non-negative matrix factorisation
(Bengfort et. al, 2019, p. 121)

Bengfort, Benjamin; Bilbro, Rebecca; Ojeda, Tony. Applied Text Analysis with Python 2019

Unsupervised Text Modelling

Topic Modelling – Federalist Papers

Applying the most frequently applied algorithm LDA (Blei *et al.*, 2003), with topics = 4 and words per topic = 10 to extract the themes of papers 5, 15, 17, and 18, yielded the results below.

topicid	word	weight	topicid	word	weight	topicid	word	weight	topicid	word	weight
0	cities	15	1	States	30	2	may	16	3	would	69
0	members	12	1	would	22	2	States	12	3	nations	15
0	Macedon	11	1	upon	11	2	Union	10	3	one	14
0	Achaeans	9	1	citizens	8	2	government	9	3	different	13
0	league	9	1	Connecticut	6	2	republic	7	3	foreign	12
0	Greece	9	1	part	6	2	distance	7	3	others	11
0	government	7	1	State	6	2	great	7	3	confederacies	11
0	council	6	1	rule	5	2	representatives	5	3	might	9
0	war	6	1	debt	4	2	democracy	5	3	America	9
0	confederacy	6	1	likely	4	2	new	5	3	another	8

Unsupervised Text Modelling

Topic Modelling – Federalist Papers

The themes are clearly distinguishable in the prediction table below. Paper 18 covers analogies with ancient Greece, Paper 17 covers possible encroachment of federal government on powers of state governments, Paper 14 covers forms of government, and Paper 5 covers troubles related to Britain's division into different nations. The confidence assigned to each topic for each Paper is also very high, easily differentiating the different themes.

Federalist Paper	prediction (Topic)	confidence (Topic_0)	confidence (Topic_1)	confidence (Topic_2)	confidence (Topic_3)
No. 5	Topic_3	0.000	0.000	0.000	0.999
No. 14	Topic_2	0.000	0.000	0.997	0.002
No. 17	Topic_1	0.000	0.866	0.000	0.134
No. 18	Topic_0	0.998	0.000	0.000	0.002

Unsupervised Text Modelling

Topic Modelling

Which topic modelling algorithm is best? Anecdotally, LSA is sometimes considered better for learning descriptive topics, which is helpful with longer documents. Latent Dirichlet Allocation and non-negative matrix factorization, on the other hand, can be better for learning compact topics, which is useful for creating clear labels from topics. Ultimately, the best model will depend on the corpus you are working with and the goals of your application.

Unsupervised Text Modelling

Topic Modelling

Stevens et al. (2012) use **topic coherence scores, UCI** and **UMass** to reveal that LDA and LSA each have different strengths; LDA best learns descriptive topics while LSA is best at creating a compact semantic representation of documents and words in a corpus.

Topic coherence scores increase the more the topic is human interpretable. They score a single topic by measuring the degree of semantic similarity between high scoring words in the topic.

Unsupervised Text Modelling

Topic Modelling

Stevens et al. (2012) For the **UCI Metric**, word probabilities are computed by counting word co-occurrence frequencies in a sliding window over an external corpus such as Wikipedia. For the **UMass Metric,** the score is calculated based on document co-occurrence of words. It computes over the original corpus used to train the topic models rather than an external corpus.

Kido et al. (2016) describe a novel approach for topic modelling using the Louvain community detection algorithm which is run across a term co-occurrence graph network. We will be exploring the use of graph networks for text modelling later in the module.

Unsupervised Text Modelling

Topic Modelling – References

Bengfort, B., Bilbro, R. and Ojeda, T. (2019) *Applied Text Analysis with Python*. Newton, MA: O'Reilly Media.

Blei, D., Ng, A. and Jordan, M. (2003) *Latent Dirichlet Allocation*. *Journal of Machine Learning Research*, 3, pp.993-1022.

Deerwester, S., Dumais, S., Furnas, G., Landauer, T., Harshman, R. (1990) *Indexing by Latent Semantic Analysis*, Journal of the American Society for Information Sciences (Vol. 41, Issue 6, pp. 391-407)

Kido, G., Igawa, R., Barbon, S. (2016). *Topic Modelling Based on Louvain Method in Online Social Networks*. XII Brazilian Symposium on Information Systems (Vol 1).

Unsupervised Text Modelling

Topic Modelling – References

Kotu, V. and Deshpande, B. (2019) *Data Science Concepts and Practice*. 2nd ed. Burlington, MA: Morgan Kaufmann Publishers.

Lee, D., Seung, H. (1999) *Learning the parts of objects by non-negative matrix factorization. Nature* (Vol. 401, pp. 788–791)

North, M. (2018) *Data Mining for the Masses*. 3rd ed. Global Text Project.

Paatero, P., Tapper, U. (1994) 'Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values' Environmetrics (Vol. 5, Issue 2, pp.111-126)

Unsupervised Text Modelling

Topic Modelling – References

Stevens, K., Kegelmeyer, P., Andrzejewski, D., Buttler, D. (2012) *Exploring Topic Coherence over many models and many topics.* Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (pp. 952-961).

.