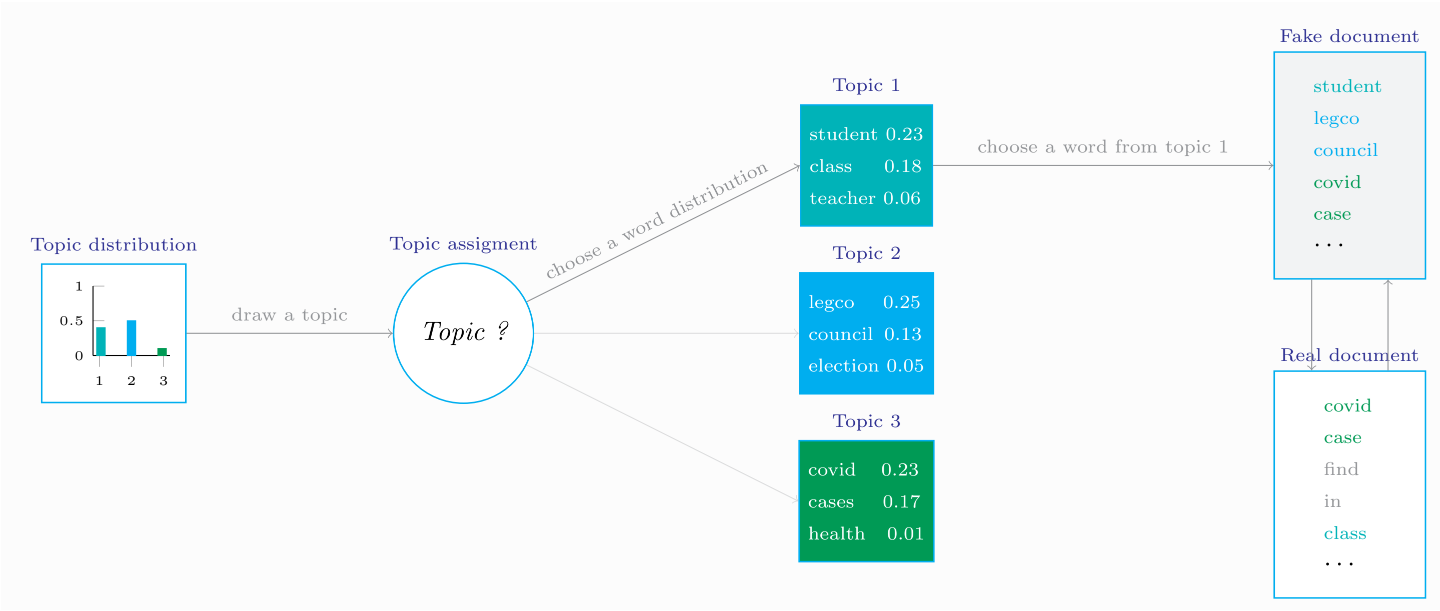

What is Latent Dirichlet allocation?

LDA is an unsupervised method for classification in machine learning. It works on documents, sound, and graphics.
In this article, we will focus on document classification only.

Document classification with LDA

Let say we have 3000+ RTHK news documents. It is hard to classify all news manually and also very time-consuming.

In 2003 LDA was applied in machine learning by David Blei, Andrew Ng, and Michael I. Jordan which allow us to do it automatically now.

Abstract of LDA process

Let's take a look at the graphic from above. The goal of LDA is to generate a fake document and compare it with a real document to find the best topic distribution and the best word topic distribution.

The computer program will continuously update the topic distribution *(the leftmost)* and the word topic distribution *(Three different colors square at the middle)* to make the fake document more close to the real document

The generative process of the fake document:

1. K number of topics will be defined by a user casually. It's three in this case.

2. Topic distribution will be randomly assigned then draw a topic. Let's say topic 1

3. Randomly draw a word from topic 1's word distribution. Word "student" is chosen
(note that the distribution is also random at start)

4. Word "student" will be the first word in the fake document

5. Repeat this process until the number of words same as the real document

What is the best K topics?

The performance of the classification with best K topics can score by coherence.

We can try different no of topics with Dirichlet distribution hyper-parameter alpha and eta *(This will not be shown here)* to maximize the score

Why don't we define the topics at start?

Actually, in machine learning, the process is in a black box it means we cannot control or observe the result directly, and no formula usage.

As a result, we can only label the topics after processing by LDA

Extra benefit from word topic distribution

Word topic distribution not only help in classification but also tell us how important a specific word in its topic.

Example : Word topic distribution of 3000+ local RTHK news in 2020:

```
Topic 1: (0, '0.022*"students" + 0.015*"school" +...+ 0.002*"exam"')
Topic 2: (1, '0.008*"legco" + 0.007*"government" + ... + 0.003*"law"')
. . .
Topic 7: (6, '0.014*"government" + 0.008*"covid" + ... + 0.003*"social"')
```

The computer programs

The good news is:
We don't need to build up all stuff from scratch several Python library Sklearn, Gensim, Spacy are available.

Python program with gensim and result(rar files):
github.com/ccuuttww/TopicModeling/tree/main/LDA_RTHK