STARSCHEMA

WHITE PAPER

Understanding Topic Modeling and Planning Its Implementation

Topic modeling is an algorithm for structuring, understanding and labeling textual data — a field of natural-language understanding (NLU), which is a branch of unsupervised machine learning (ML) models.

Topic modeling recognizes and learns latent — hidden — topics from a corpus — a structured collection of documents. The methodology is primarily used in text mining to reveal the semantic structures within texts.

Topic modeling is based on two assumptions. First, every document — any cohesive text like an article, social media comment or entire Wikipedia entry — is an amalgam of multiple topics. And second, that every topic is a collection of different words. We can thus conceptualize topics as groupings of certain words. In topic modeling, it intuitively follows that certain expressions appear more frequently in a given topic than in others. To illustrate, a document that contains the words "rocket," "planet" and "star" is more closely related to the topic of space, while a document with the words "forest," "mountain" and "meadow" has more to do with the topic of nature. In most cases, a document contains different ratios of words from multiple topics. Thus, if 99% of words in a document are related to space, and 1% is related to nature, it is safe to assume that the document is about the topic of space.

Similarly to the example above, by understanding the semantics of documents and revealing the latent topics in them, we can use topic modeling to create labels to group documents. A model can recognize similar comments and questions within a corpus (for example, the comments on the social media pages of a streaming service provider can provide insights about the reception of a new show or if there is an issue with the service), leverage the connections it recognizes for a recommendation system or compile textual information into one or more labels. In each case, it's uncovering the connections or topics in the text.

Topic Modeling Algorithms

There are several established ways to use topic modeling, and multiple algorithms have been developed over the years. Often, these algorithms build on each other or rectify the weaknesses and disadvantages of pre-existing algorithms. The four most popular algorithms are SVD, LSA, pLSA and LDA, as well as lda2vec, which is based on a deep neural network. LDA is one of the most advanced algorithms in topic modeling and is discussed in detail below.

The LDA (Latent Dirichlet Allocation) algorithm was developed by Prof. David M. Blei in 2003. To understand its core principle, we can look at the significance of the words that make up its name:

Latent: An unknown label or topic is hidden in a text, we have no preliminary knowledge of it, but we assume it exists.

Dirichlet: The Dirichlet distribution — named after Peter Gustav Lejeune Dirichlet — is a type of categorical probability distribution. While Normal distribution takes samples from the realm of real numbers, the Dirichlet distribution takes samples from a probability simplex.

A probability simplex contains K separate categories, with each category possessing a probability and the sum of these probabilities equaling 1. A simple example is a coin or a die. The coin has two categories (heads or tails), and their probability is 0,5 in each case. The die has six categories, and its probability simplex is (1/6, 1/6, 1/6, 1/6, 1/6, 1/6). Unlike the coin, where heads or tails have the same probability of occurring, different words in a language appear with different frequency. Words like "the," "a," etc. appear much more frequently than "mountain" or "star." In a corpus, the probability simplex of words depends on the frequency of words in the corpus, which can differ for each word.

Allocation: From the Dirichlet distribution, we can assign words in documents to topics and topics to documents.

In LDA models, a document is functionally viewed as a set of words. This means that the punctuation, the order of words and their grammatical role are irrelevant to the model. Note that common words that appear in at least 80% of documents don't contain additional information regarding topics — so we need to filter them out. In addition, some words contain barely any information (e.g. "the," "and," "or"). These are so-called "stop words," and they also need to be filtered out during the document's preprocessing. The difference between words that appear in at least 80% of documents and stop words is that stop words have low information content (they're usually conjunctions), while common words can be expressions that are rarer in spoken language but appear frequently in a corpus, such as "rocket," "spaceship," etc.

By preprocessing documents, we get a word list for each one. We can then use these word lists to define what words belong to a given topic and with what probability. The algorithm features the following steps:

1. All documents are iterated. Every word in a document is randomly assigned to one of the *k* topics, where *k* is the number of topics — this needs to be decided in advance in an LDA model as one of the model's input parameters.

2. For every word *w* in a document *d*, we calculate the following:

   - p(topic *t* | document *d*), which is the document distribution of topics, i.e. the ratio of words assigned to topic *t* of document *d*.

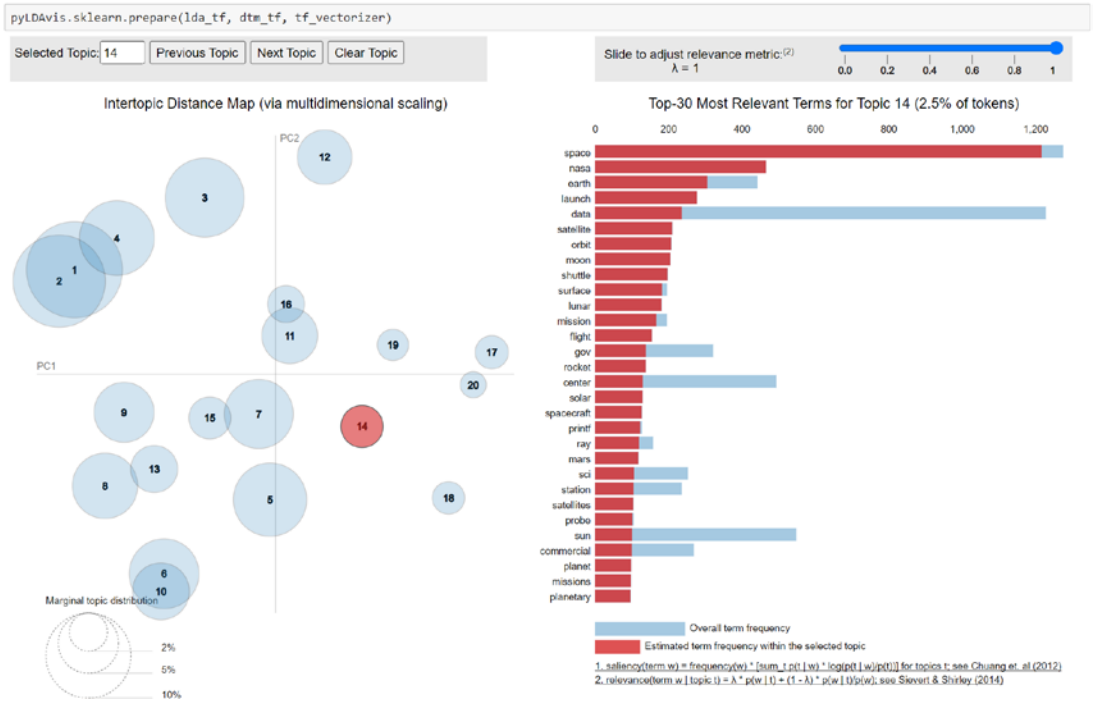
   - p(word *w* | topic *t*), which is the word distribution of topics, i.e. the ratio of word *w* in the corpus (all documents) to topic *t*. This tells us how many documents are assigned to topic *t* because of word *w*.

3. The probability of word *w* for topic *t* is updated according to the following: p(topic *t* | document *d*) * p(word *w* | topic *t*).

Finally, the LDA algorithm builds two models: a word-topic model, which shows what words belong to a given topic, and a topic-document model, which shows what topics a given document belongs to.

The success of the LDA model can be visualized with the pyLDAvis Python package. The finished visualization comprises two parts. On the left, circles represent each topic. The size of the circles is determined by the ratio of documents belonging to the topic. The central x and y coordinates of a circle are determined by the two main components resulting from dimensionality reduction (PCA/t-SNE), and these two coordinates enable the visualization of the distance between the different topics' content. When circles representing two topics overlap, there is a high probability that the topics contain identical words. On the right, the blue bars show the number of times a word occurs in the corpus, while the red bars show how many times a word occurs in the document belonging to a given topic.

FIGURE 1



These two visualizations make it easy to understand the relationship between various topics, as well as what words make up each topic and how frequently these words occur in the entire corpus.

Such a model enables the user to analyze textual documents faster and more easily. It can be used, for example, to group the comments that a streaming service provider receives and see what topics they form. One topic can be service interruptions, another the launch of a new show, and expressions of general satisfaction or dissatisfaction can also form a pair of topics. These insights can help the company see how reliable a service is and can be used to increase customer satisfaction by identifying, investigating and rectifying issues that are frequently mentioned in comments.

How to Get Started Implementing Topic Modeling

There are a few established best practices for realizing a project involving topic modeling. The following steps are generally applicable to most situations and can serve as a starting point to plan a more custom-tailored approach.

1. Define business goals

Before you think about data sets or technology, you need to have a clear idea of how you want to leverage the insights gained through topic modeling. Start by isolating the immediate business problem you want to work on and anticipate the long-term implications the solution will have for further business development. For example, if the solution involves analyzing comments, you will likely need to take three steps: designate who will integrate the insights gained via labeling comments into specific business processes, define the expected outcome of the integration and designate a subject matter expert to set acceptance criteria.

As you begin to define opportunities and set goals for implementing topic modeling to solve business challenges, you need to review existing functions, areas and processes to identify those where topic modeling can promote development. One scenario where topic modeling is particularly effective is

when an analyst needs to divide a large amount of textual data into easier-to-digest groups. Dividing the data into smaller groups reduces the effort it would normally take to categorize data manually. A typical use case is when different departments handle incoming communications and need to ensure that each message finds the most appropriate recipient within the organization. Here, topic modeling can promote cost optimization by repurposing human labor to focus on higher-value work.

Another area where such an algorithm can be more effective than human labor is identifying frequently recurring patterns and events — such as a frequent issue that people regularly complain about or the reasons for leaving that people cite to HR in exit interviews. By identifying problems more effectively, quicker and stronger measures to prevent them and drive innovation can be taken.

As a last example, topic modeling can provide a much faster solution for understanding answers to a questionnaire with open-ended questions than having humans read, assess and categorize each answer.

2. Define project scope

Typically, topic modeling projects are divided into two groups based on whether they involve a one-time inquiry or are intended to provide continuous insights.

- One-off projects:

For one-time inquiries, prepare to have a subject matter expert review and evaluate the results. A subject matter expert's analysis is needed because topic modeling can mistakenly see connections where there aren't any. This necessitates a so-called "sanity check" — for example, it would be a mistake to conclude that a message containing the phrase "explosive deals" has anything to do with actual explosives.

It's important to review topics and set topic numbers in accordance with business needs, such as identifying sources of errors.

Once a model is finished, you can also use it to make repeated predictions along its original parameters. The trained model will not be changed and will always produce the same topics for a given document. Based on the most frequently occurring words from each topic, the model will help determine the subject of the document.

- Continuous projects:

If a machine learning model will see repeated use in an open-ended manner or over a set period, you can use it to assign new data to topics. Normally, such use cases don't require retraining the model, but when there's a considerable increase in the amount of data or the content of the new data differs significantly from the original training data, model retraining will be necessary.

When retraining the model, it is important to involve a subject matter expert and decide if fine-tuning the model is needed. Occasional retraining can improve the model's accuracy, but constant training and review is highly labor-intensive.

3. Ensure the availability of data

Once you've decided what purpose the result of your foray into topic modeling will serve, you need textual data that is suitable for the task at hand. If such data does not exist or is insufficient, in most cases, there are two options: expand the range of existing data-collection solutions or introduce new ones. When expanding data-collection capabilities is not an option, consider buying data. Several companies (Neticle, Sentione, etc.) can provide textual data about your company or even competitors, which is especially useful for comparative analysis.

STARSCHEMA

Conclusion

Topic modeling can be a powerful and flexible tool for understanding and labeling natural-language-based data. It can represent a significant step towards more comprehensive data utilization for a wide range of organizations by extracting hidden value contained in text within sets of documents — but it requires careful preparation and expertise in analyst target domain and data science. Once a project is set up, however, it is relatively easy to repeat and optimize to provide updated and regular insights that can support a variety of business processes.

ABOUT THE AUTHOR

Balázs Zempléni

Data Scientist

Balázs holds a degree in Engineering, and specializes in digital image and signal processing. He has worked for multiple banks in different roles in the fields of data engineering and business intelligence. In recent years, he has focused on developing a natural language processing solution to improve internal business processes based on textual data. In addition to his work, Balázs likes to hold presentations at meetups and conferences.

COMPANY

At Starschema we believe that data has the power to change the world and data-driven organizations are leading the way. We help organizations use data to make better business decisions, build smarter products, and deliver more value for their customers, employees and investors. We dig into our customers toughest business problems, design solutions and build the technology needed to compete and profit in a data-driven world.

GET IN TOUCH

info@starschema.com
starschema.com