Machine Learning pipeline:

Step1. Encoding 10,000 news articles using word2vec. [note1]

Step2. Using the LDA model [note2], divided the existing 10,000 news articles into 100 topics.

Step3. Record each article's embedding and classification date (time-stamp) of each article.

Step4. Use article's embedding to vectorize 100 topics. [note3]

Step5. Record each topic's embedding.

Step6. Every time there is a new article to analyze, perform the following actions:

6.1 Calculate the embedding vector [note1] of the article and match it to the closest topic. [note4]

6.2 The article which has been analyzed is added to the topic, while one article which belongs to the topic and is the farthest from the topic is removed to ensure that only 10,000 articles are stored.

6.3 Recalculate embedding vectors for the topic with articles in and out.

Step7. Every 3 months, using the stored 10,000 articles, recalculate 100 topics [note5].

Note:

[note1]

Encoding of the article, steps to do:

1.Tokenize-> Extract each word or phrase in the article to form a list.

2.Remove stop words-> Removes words or phrases belonging to stop words in the list.

3.Prepare a model-> Pick up a trained word embedding model, such as Glove or Word2Vec.

4.Embedding-> Using the embedding model calculate the mean of the word vectors in the list, and record it as the encoding of the article.

[note2]

LDA model (Latent Dirichlet allocation)

The LDA model is a popular topic classification model.

First, It assumes the generation logic of each word in the article, and then uses this assumed generation logic and real data to reverse engineer the parameters in the generation logic.

The word generation logic of LDA is: first, we randomly select 1 topic from the distribution of topics; then, we randomly select words according to the distribution of words in the selected topic to generate sentences or paragraphs; finally, we have a article's length distribution to decide whether ends the generation of the article.

[note3]

Encoding of the topic, steps to do:

1.Find the vector for each article in the topic.

2.Weighted average of those article vectors to obtain the vector of the topic.

[note4]

Similarity matrix:

We can use Euclidean distance or cosine similarity to calculate the distance between the article and each topic, and select the topic with the smallest distance as the topic of the article.

[note5]

Over time, as the distribution of article topics changes, so does the distribution of words in the topics. In order to make sure that the model can keep up with the trend of the article's style, the model is reconstructed every 3 months.

#------------------------------

What if extracting topics from a collection of news articles is an expensive operation and should only be performed sparingly?

In my pipeline scheme, the encoding of new articles can be calculated separately, independent of other articles, and the calculation efficiency is high; in addition, we store the vectors of 100 topics in advance, and only need to calculate 100 times when matching the article to a topic , which does not require excessive computing resources.

If the consumption of resources or the time required is still unacceptable, we can use PCA or other dimensionality reduction methods to reduce the dimension of the vectors of articles and topics to reduce resource consumption.

#------------------------------

How would you decide when it is a good idea to update your topic list?

A relatively simple and feasible way is to update the topic list periodically, for example, every 3 months. If this method cannot meet the required accuracy, we can store the vectors of the initial 100 topics separately, calculate the distance between the existing 100 topics and the initial 100 topics every week, and we will update the topic list when the distance exceeds a threshold value which we set.