Topic modeling

Hi everyone, I’m Peitian Zhang from Renmin University of China. Today I’m going to present several works about topic modeling and share some insights. Feel free to interrupt me if you have any questions.

I will first give an overview of topic modeling, then I will introduce common datasets, metrics in this field. After that I will present 3 types of methods from non-neural to PLM based, each category contains three papers.

So what is topic modeling? It is a kind of unsupervised tool to automatically discover topics from the text corpus. It has to be unsupervised because the topics are actually unobserved, and we just involve them in explaining the data generation process.

Some applications of topic modeling includes document analysis, text summaries, and ad-hoc information retrieval.

Some papers, especially traditional works, may contain complex equations, so I tried my best to unify all the notations:

Generally, all the topic models seek for three distributions: ; Most topic models make the bag-of-words assumption.

There are quite a lot of metrics to mention in topic modeling, each paper may adopt different metrics. In the early times, perplexity is used. However, a high perplexity score is proven to be irrelevant to good topic modeling. Therefore, the topic coherence is considered to be an important feature. There are many quantitative score to measure topic coherence including UMass, UCI and NPMI. Some papers introduce a new task to evaluate topic models: they may cluster documents by their topic distribution. In summary, there isn’t a consensus in evaluating topic models.

Next, let’s move to the actual methods.

First I will introduce traditional methods which always resort to analytic form of the aforementioned three distributions.

PLSA is a probabilistic version of LSA because it maximizes the likelihood of data generation process. It assumes that each document d in the corpus is generated by:

Because all the words are generated independently, we can write pwd as. We can rewrite the formula because the words can be though of sampled from a multivariate distribution of N categories.

Let’s first look at the architecture of the model. It contains three parts. The encoder models p(z|d) by a neural network; the generator models p(w|z), however, here the topic is a fake one. Both module concatenates the topic vector and the document vector and feed it into the discriminator. The discriminator outputs a score for each pair, a larger score means the model believes the pair is real. In the training, the three modules are jointly trained so that the generator can learn a better topic-word distribution; and the encoder can learn a better document-topic distribution.