Latent Dirichlet Allocation on 1M ABC News Headlines

Data Description. The dataset consists of one million published news headlines from the Australian Broadcasting Corporation (ABC) accumulated over 19 years. Each data row contains the news headline document and a timestamp of the publication date. The data repository is publicly available here: https://www.kaggle.com/datasets/therohk/million-headlines.

Model Selection. We apply a mixed membership model using Latent Dirichlet Allocation (LDA) on the ABC news dataset. We examine the convergence of the online LDA algorithm and its performance on a held out dataset of remaining test documents. We aim to interpret and analyze the allocated topics to uncover how the topics have been determined and assigned.

Data Pre-Processing. To perform LDA, we must first decide on how to represent our headlines as input vectors that represent each sample. We decided to transform the dataset through (1) bag-of-words and (2) TF-IDF vectorization. We use bag-of-words for tokenization and formulate a vocabulary of the training data. TF-IDF vectorization encodes additional information on the relevance of each word to the headline across the training corpus. Applying the vectorizer helps account for artifacts such as irrelevant words without the need to thoroughly pre-process the data. We decided to avoid applying dimensionality reduction as we would like to examine the individual words that remain highly relevant to the each of the topics.

LDA Generative Process. The vectorized processed dataset consists of data groups (documents) which represent a collection of data points (words). Similar to mixture models, we have components and proportions but we use mixed membership in topic models to provide additional flexibility given that documents can belong to multiple topics. LDA generative process can be represented as shown in **Algorithm 1**.

Algorithm 1 Latent Dirichlet Allocation

```
$K \leftarrow 10$
Initialize hyperparameters $\eta, \alpha$
for each topic k in K do
    draw $\beta_k$ from exchangeable $\text{Dir}_V(\eta)$
end for
for each document $i$ in $n$ documents do
    randomly sample $\theta_i \sim \text{Dir}(\alpha)$
    for each word $j$ in document $i$ do
        draw topic assignment $z_{ij}|\theta_i \sim \text{Cat}(\theta_i)$
        draw word $x_{ij}|z_{ij}, \beta \sim \text{Cat}(\beta_{z_{ij}})$
    end for
end for
```

We preset the number of topics K to 10. β_k represents the mixture components (topics) which are shared across the entire corpus. The total number of samples n is one million. θ represents the mixture proportions. z_{ij} is a latent categorical variable representing the cluster assignments for each data point $x_i j$, where i corresponds to the data group or document. We use the probabilistic library `gensim` to build an LDA model and perform stochastic variational inference. We used the computationally efficient multicore version of the `LDAModel` class within gensim to process all the training samples.

Convergence using Gensim LDA. After defining the parameters for `gensim` LDA model, we perform SVI on the ABC text documents. However, given that the multicore version of LDA does not support callbacks for retrieving convergence metrics and training perplexity, the following convergence plots were formulated using 50K samples over 240 epochs:

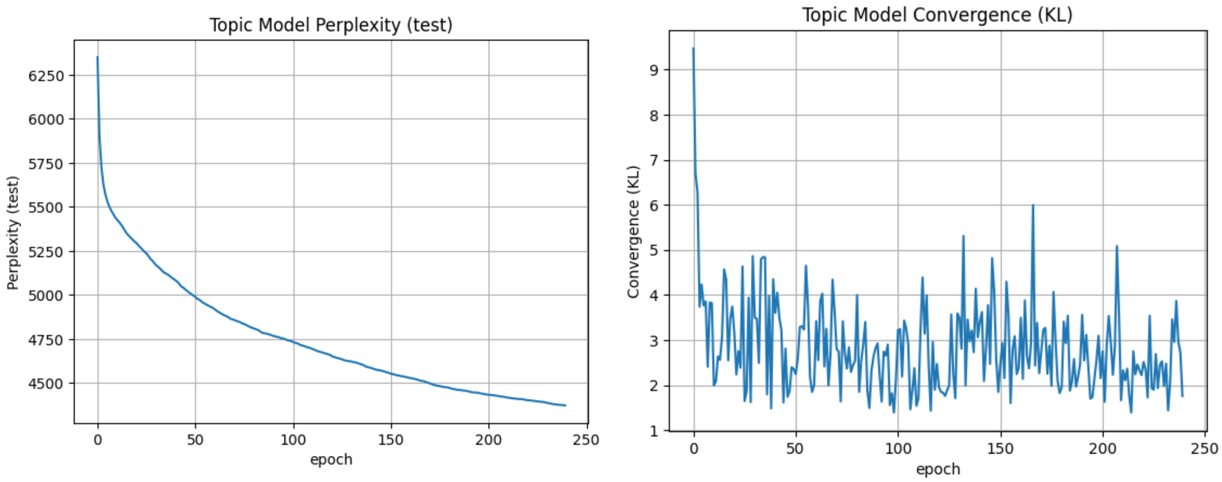


Figure 1. LDA convergence measured by training perplexity (left) and KL-Divergence (right).

In Fig. 1, we are able to observe the training loss convergence based on the training perplexity and convergence metric based on KL divergence. The convergence metric measures the differences between individual topic distributions, and we can see that it convergences faster from around 2-4 but in a much noisier fashion compared to training perplexity.

We observe the performance of our LDA fit based on the perplexity of the held-out dataset over each training epoch in Fig. 2. The perplexity begins to gradually converge after a steep decline in the initial epochs. The left-most figure represents metrics retrieved from training single-core LDA on a training set of 50K and a held-out set of 50K, while the right-most figure is based on metrics from the multi-core LDA on the full 1M training corpus with a held-out dataset of 184 documents. We can observe while both decrease in perplexity, it is much less noisy with the single-core test perplexity, most likely due to the greater number of documents present in the held-out dataset.

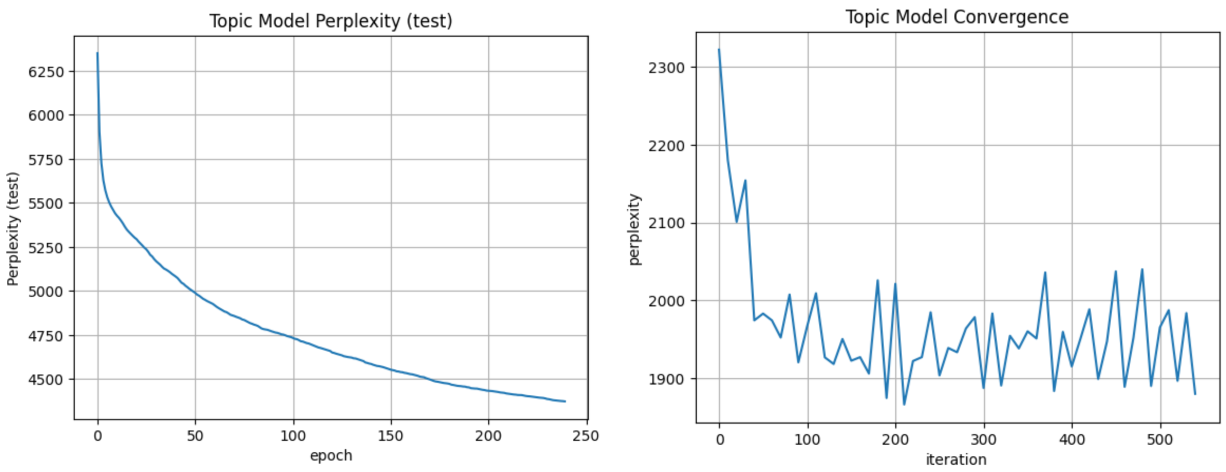
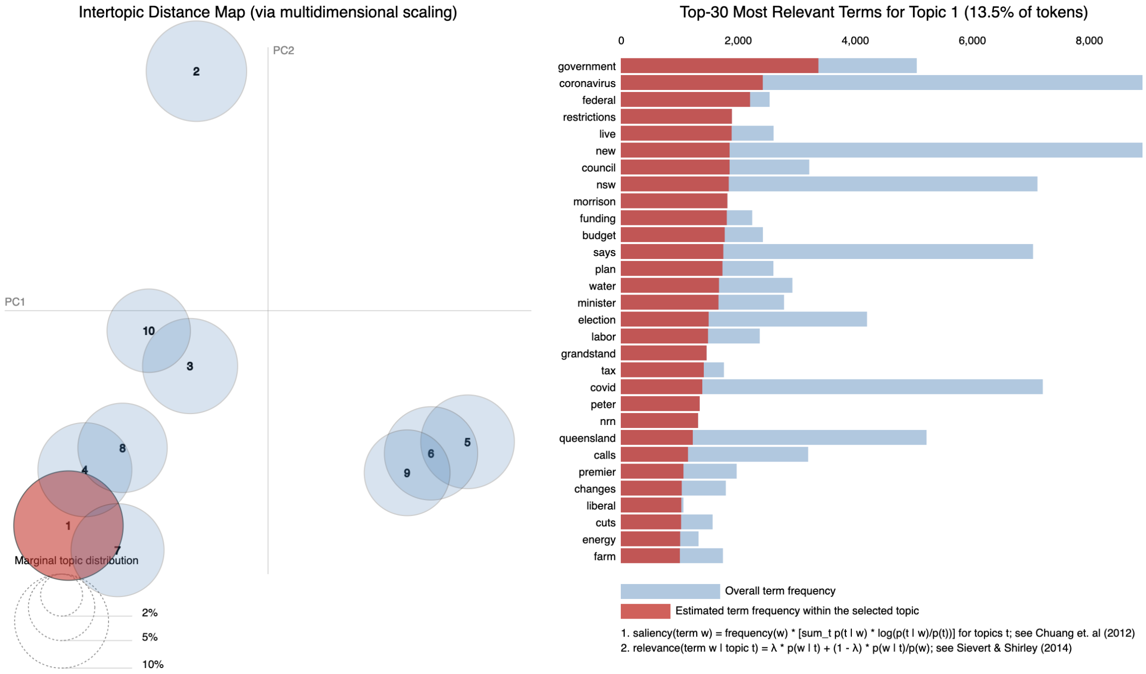


Figure 2. Measured perplexity for held-out corpus of 50K documents over 240 epochs (left) and held-out corpus of 184 documents over 40 epochs (right).

Interpretation and analysis of learned mixture components. Using the `pLDAvis` framework, we are able to visualize our components as represented in Fig. 3. The Intertopic map allows us to reduce the dimensionality and map the components on a 2D space. For each of the components, represented as bubbles on the map, we can observe the most relevant terms based on their term frequency within a selected topic. We see that there are some overlapping bubbles as shown in topics 1, 4, 7, and 8. We realize that given news headlines cover politics quite heavily, the overlapping bubbles featured subtopics within politics that still somewhat overlapped. "coronavirus" in particular was a featured term for each of the topics, given that a pandemic of such a massive scale would have a huge general impact across documents related to politics and economy. Other bubbles can be separated quite easily such as vehicular accidents.



1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Figure 3. Latent space of fitted LDA model (left) and most relevant terms for topic 1 (right)