Topic Modeling: Gensim

What is Gensim

Gensim is a Python-based topic-modeling LDA (Latent Dirichlet Allocation) algorithm (https://www.machinelearningplus.com/nlp/topic-modeling-gensim-python/). It is designed to automatically identify high quality semantic topics from a large number of documents.

Topic modeling and LDA

A topic-modeling tool takes a text corpus and looks for patterns in the use of words. For large amounts of text, topic modeling provides a quick way to get "the lay of the land", to get a sense of what the corpus is all about. This "**distant reading**" of a corpus is not a substitute for "**close reading**", but it is a good start, like statistics' EDA (Exploratory Data Analysis).

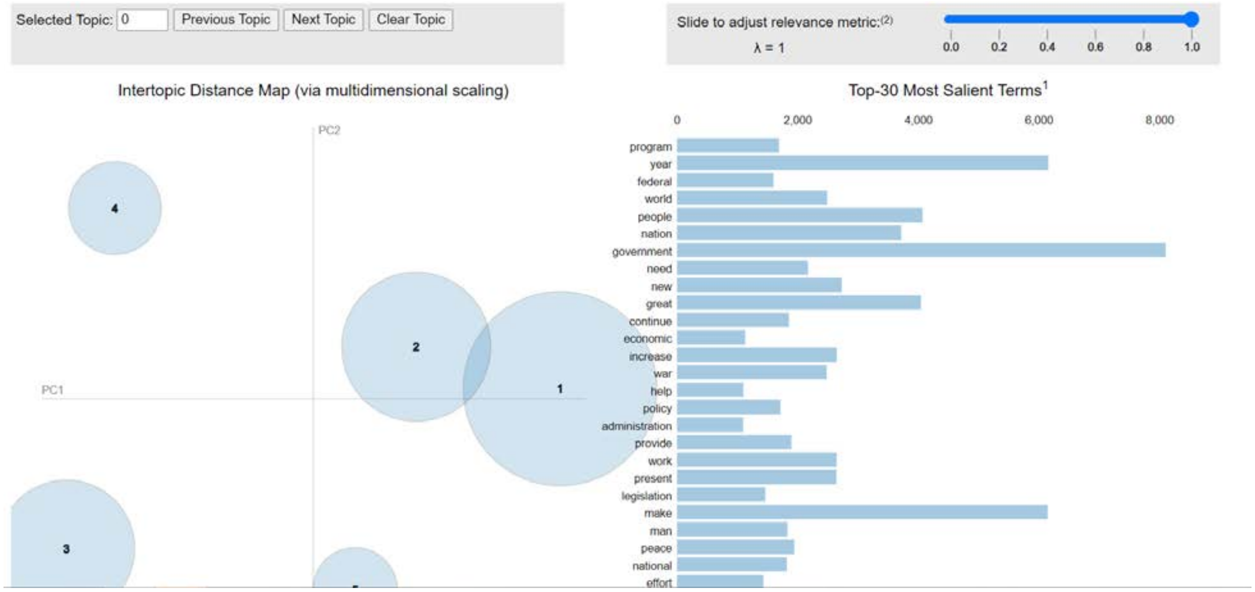
Topic models are computer programs that extract topics from texts. A **topic**, for these computer programs, is a list of words that occur in statistically meaningful ways. A **text** can be anything: a novel, a university mission statement, a newspaper editorial, an email, a blog post, a book chapter, a journal article, a diary entry. This text is unstructured, i.e., it does not contain any computer-readable annotations (tags) that tell the computer the semantic meaning of the words in the text.

Topic Modeling and LDA are often cited together. But LDA is a special case of topic modeling created by Blei et al. (2002). Among the many topic modeling approaches, LDA is by far the most popular. The myriad variations of topic modeling have resulted in an alphabet soup of techniques and programs to implement them that might be confusing or overwhelming to the uninitiated; ignore them for now. They all work in much the same way.

How to interpret the Gensim output

Gensim will display dynamic and interactive topic results as an html file using pyLDAvis.

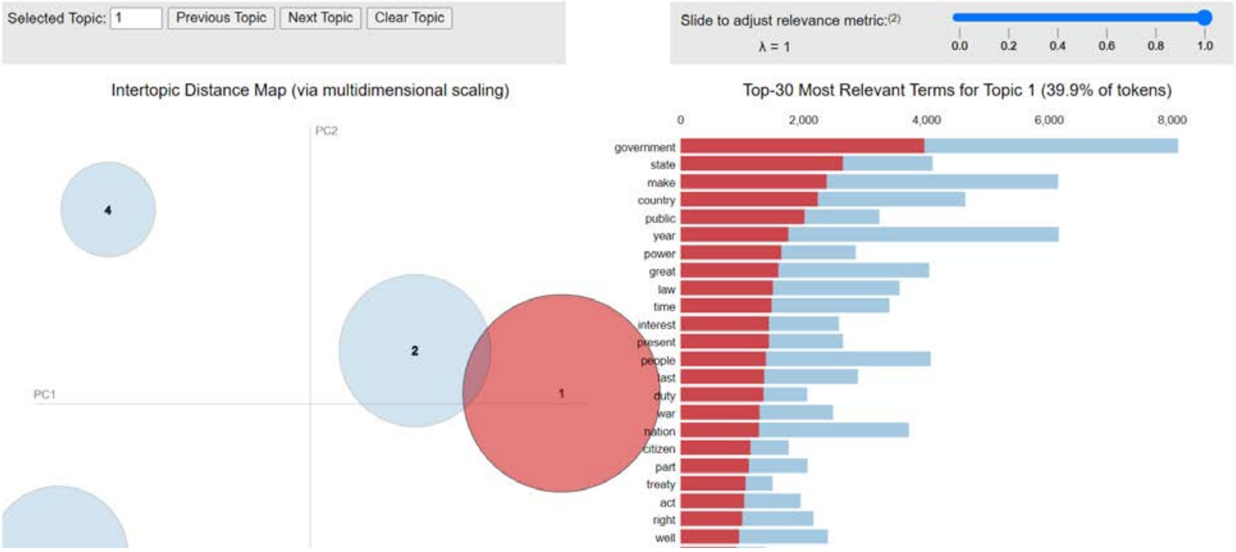
An example of Gensim output looks like the following, using the corpus of US Presidents State of the Union speeches and 5 topics, selected for minimum overlap:



Intertropic distance map

The **left panel** is a two-dimensional projection by implementing multidimensional scaling. Each circle on the left represents a topic and is sorted in the decreasing order of prevalence. Circles that are nearer are more similar to each other. Meanwhile, the larger the circle is, the more prevalent the topic is among the input files. In general, **we would desire an output with non-overlapping and big bubbles scattered throughout the chart, instead of small bubbles clustering together.**

On the **right panel**, the horizontal bar charts represent individual terms that can best represent the currently selected topic on the left. Selecting a topic on the left can reveal the conditional distribution over topics for the selected term. The **blue bars** represent the "overall term frequency" in the entire corpus, while the **red bars** represent the "estimated term frequency within the selected topic."

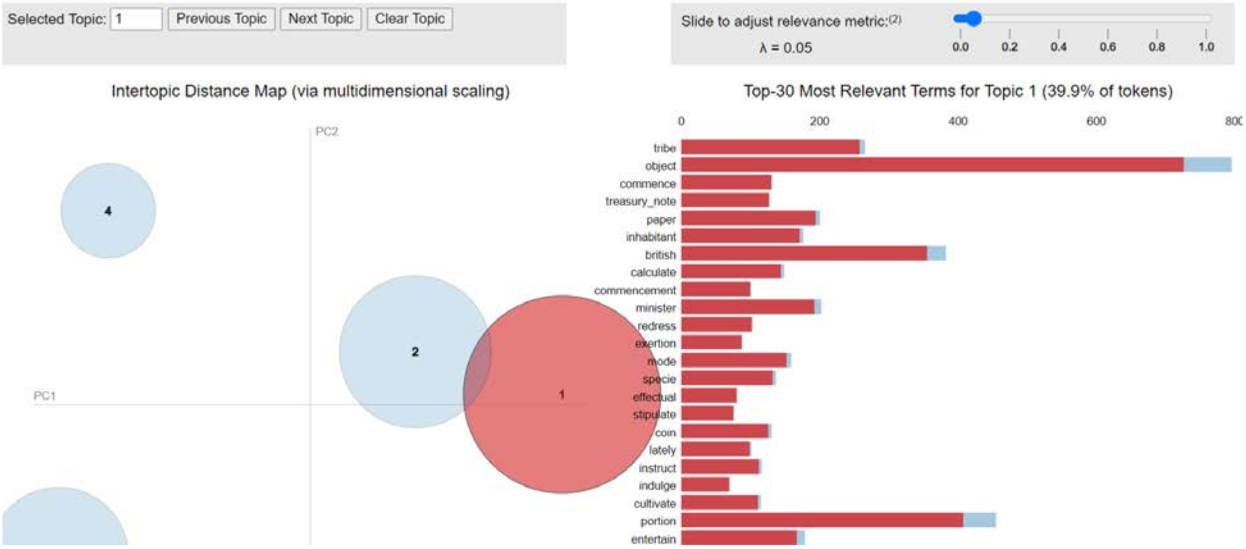

Selected Topic: 1

Slide to adjust relevance metric:(2) λ = 1

Relevance metric, λ value

By adjusting the value of λ, we can adjust the relevance metric, which is calculated by relevance (term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w). (Sievert and Shirley 2014)

Change the value of λ to adjust the term rankings – small values of **λ (near 0)** highlight potentially **rare, but exclusive terms for the selected topic**, and large values of **λ (near 1)** highlight **frequent, but not necessarily exclusive, terms for the selected topic**.


Selected Topic: 1

Slide to adjust relevance metric:(2) λ = 0.05

"Setting λ = 1 results in the familiar ranking of terms in decreasing order of their topic-specific probability, and setting λ = 0 ranks terms solely by their lift," where the lift is the ratio of reds to grays/blues; **high reds, high lift**.

Coherence value (and number of topics)

How should you select the appropriate number of topics (k) to run your model? One way to finding the optimal number of topics is to build several LDA models using different values of number of topics and pick the one that gives the **highest coherence value**.

Choosing a 'k' that marks the end of a rapid growth of topic coherence usually offers meaningful and interpretable topics. Picking an even higher value can sometimes provide more granular sub-topics. If we end up with the majority of our original texts all in a very limited number of topics, then we take that as a signal that we need to increase the number of topics; the settings were too coarse. (For more see Griffiths and Steyvers 2004).

If you see the same keywords being repeated in multiple topics, it's probably a sign that the 'k' is too large.

Gensim vs. Mallet

Mallet LDA algorithm typically gives a better quality of topics but has no visualization options. The NLP Suite builds this visualization using Excel charts. Mallet is also faster.

References

Blei, David M., Andrew Y. Ng, and Michael I. Jordan. 2002. "Latent Dirichlet Allocation." Advances in Neural Information Processing Systems, 14.

Griffiths, T. L. and M. Steyvers. 2004. "Finding scientific topics". Proceedings of the National Academy of Science, 101, 5228-5235.

Sievert, Carson, and Kenneth Shirley. 2014. "LDAvis: A Method for Visualizing and Interpreting Topics." Pp. 63–70 in Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces. Baltimore, Maryland, USA: Association for Computational Linguistics.

TIPS_NLP_Topic modeling Mallet.pdf