**Full Details on Topic Modeling using LDA**

First we tried step by step LDA topic modeling in Google Co-lab on abstracts only. LDA is an unsupervised machine learning algorithm that **discovers hidden topics** in a large collection of documents. It assumes:

* Each document is a mixture of topics.
* Each topic is a distribution over words.

Your goal is to apply this model to the **"Abstract" column** of patent documents to find **dominant themes** in innovations related to cyberbullying or cyber safety.

**Step 1: Upload Excel File**

A close-up of a computer screen

AI-generated content may be incorrect.

Colab runs in the cloud. files.upload() opens a browser interface to let you upload your Excel file into the Colab runtime.

**Step 2: Install Required Libraries**



pandas: Load and manipulate the Excel dataset.

openpyxl: Needed to read .xlsx files.

gensim: Core library for LDA modeling.

nltk: For tokenization, stopword removal, lemmatization.

pyLDAvis: To visualize and interpret LDA topics.

**Step 3: Load the data**

A computer screen shot of a computer code

AI-generated content may be incorrect.

This loads your Excel file into a DataFrame and sets up the dataset for processing.

**Step 4: Preprocess the Text Data**

A screen shot of a computer program

AI-generated content may be incorrect.

* **Lowercasing**: Helps normalize words (e.g., "Cyber" vs "cyber").
* **Removing non-words**: Gets rid of punctuation and symbols.
* **Tokenization**: Splits the text into words.
* **Stopword removal**: Removes common words like “the”, “and”, etc., which are not informative.
* **Lemmatization**: Reduces words to their root form (e.g., “children” → “child”, “running” → “run”), helping unify terms.

These steps significantly reduce noise and dimensionality, improving topic detection quality.

**Step 5: Create a Dictionary and Corpus for Gensim**

A screenshot of a computer code

AI-generated content may be incorrect.

* **Dictionary**: Maps each word to a unique ID.
* filter\_extremes:
  + no\_below=2: Ignore words that appear in fewer than 2 documents (too rare = noise).
  + no\_above=0.5: Ignore words that appear in more than 50% of documents (too common = not topic-specific).
* **Corpus**: Converts each document to a bag-of-words format (word ID + frequency).

This creates the necessary format to feed into the LDA model.

**Step 6: Train LDA Model**

A screenshot of a computer program

AI-generated content may be incorrect.

| **Parameter** | **Choices explained:** |
| --- | --- |
| passes | 20 (good balance for small-medium datasets) |
| num\_topics | We Tried 4–10, use coherence to select best (See below) |
| chunksize | 10 |
| alpha | 'auto' for better learning of topic distributions |

**Step 7: Visualize the topics**

A white background with black text

AI-generated content may be incorrect.

* pyLDAvis generates an interactive plot.
* It helps you:
  + Explore **top keywords** per topic.
  + Assess **topic overlap**.
  + Investigate document-to-topic distributions.

This is essential for making sense of your results beyond raw numbers.

**passes**: 10–30 is typical for small/medium datasets.

**chunksize**: 10–100 is good depending on dataset size.

we Tries different num\_topics (from 4 to 10) and evaluate coherence (explained below).

**Step 8: Evaluate coherence score**

from gensim.models import CoherenceModel

coherence\_model\_lda = CoherenceModel(model=lda\_model, texts=df['tokens'], dictionary=dictionary, coherence='c\_v')

coherence\_lda = coherence\_model\_lda.get\_coherence()

print('\nCoherence Score: ', coherence\_lda)

* **Coherence score** quantifies how interpretable the topics are.
* c\_v coherence (based on cosine similarity) is more aligned with human judgment.

Typical range:

* 0.3–0.5 = ok
* 0.5–0.65 = good
* 0.65 = very good

We use this to **compare models with different topic numbers**.

|  |  |
| --- | --- |
| k | Coherence Score |
| 4 | 0.3780258097807615 |
| 5 | 0.33286056287200 |
| 6 | 0.3115560759997975 |
| 7 | 0.34771445653160565 |
| 8 | 0.35821526506480283 |
| 9 | 0.3910421684911002 |
| 10 | 0.37917127115287896 |

Summary of the results:

A score of ~0.39 (k=9) is **modest but acceptable** in short-text corpora like patent abstracts. Patent language is often technical and vague, which can limit coherence scores. A value above 0.35 is considered informative, especially if topics make semantic sense.

Topic by topic Analysis

| **Topic** | **Top Keywords** | **Interpretation** | **Relevance** |
| --- | --- | --- | --- |
| **0** | application, access, web, information, request, dashboard | **Web-based software applications** managing user access or content — could include dashboards for moderation or parental controls | ✅ Possibly relevant to managing user interfaces or access for vulnerable groups |
| **1** | abuse, service, image, medium, processing, social | **Abuse detection in media/social services** — analyzing images or multimedia on platforms like social media | ✅ Highly relevant for cyberbullying, especially visual abuse |
| **2** | detection, video, network, control, url | **Network-based abuse/video content detection**, likely involving deep packet inspection or streaming video moderation | ✅ Relevant for online video/chat safety |
| **3** | content, input, module, message, virtual | **Content moderation systems**, modules processing virtual inputs (like messages or chats) | ✅ Important for chat and messaging safety, especially for kids |
| **4** | data, digital, event, threat, machine | **Cybersecurity / digital threat detection**, using AI to detect malicious activity in online events | ✅ Foundational to cyber safety infrastructure |
| **5** | device, user, response, call, communication | **Mobile or communication device protection**, maybe anti-phishing or spam control | 🟡 Possibly relevant, depending on application context |
| **6** | model, vehicle, apparatus, motor, signal | **Autonomous or physical systems**, less aligned with cyber safety (likely about cars or IoT) | ❌ Not directly relevant |
| **7** | wearable, air, child, internet, designed | **IoT/wearable devices for children**, possibly air quality monitors or parental monitoring | ✅ Great connection to safety for children |
| **8** | server, security, unit, managing, database | **Back-end security infrastructure**, managing user data securely | 🟡 Infrastructure-level safety, could support cyber safety systems |

**🧭 Insights Summary**

**✅ High-Relevance Topics**

* **Topic 1**: Abuse in media and social systems
* **Topic 2**: Video/network-based content detection
* **Topic 3**: Chat/message filtering modules
* **Topic 4**: Digital threat & ML-based cybersecurity
* **Topic 7**: Wearables & IoT for kids

**🟡 Medium-Relevance Topics**

* **Topic 0**: Application/web dashboard design
* **Topic 5**: Mobile device abuse detection
* **Topic 8**: Secure server and backend management

**❌ Low-Relevance/Outlier**

* **Topic 6**: Autonomous vehicles/sensors (likely out of scope)

A screenshot of a computer

AI-generated content may be incorrect.