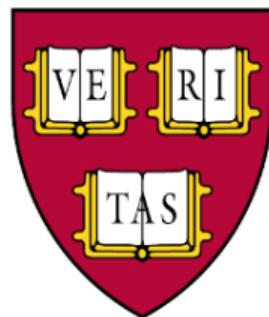


# Building a Search Engine for Educational Content

Academic Paper Analysis with LDA, Transformers, and Neural Networks

Seymur Hasanov



# Outline

1 Introduction

2 Technical Background

3 Implementation

4 Results & Evaluation

5 Conclusion

# Problem Statement

## The Challenge: Information Overload in Research

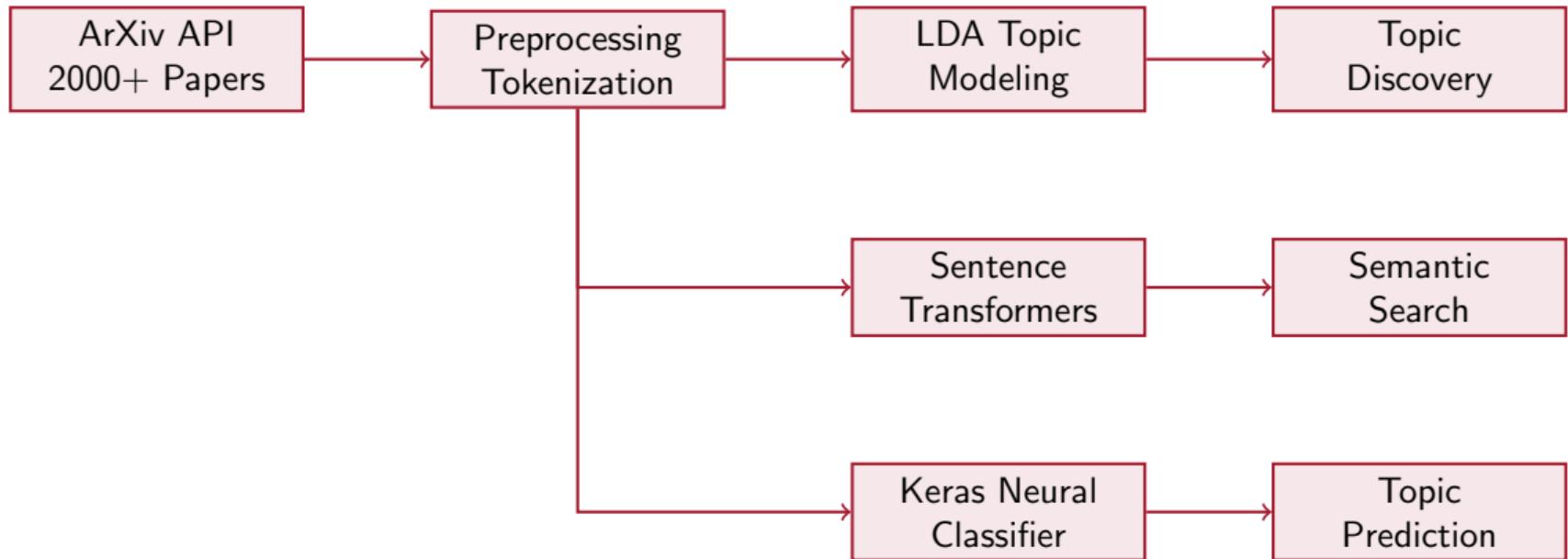
- ArXiv receives **16,000+ papers per month** in CS alone
- Researchers struggle to stay current with their field
- Manual literature review is time-consuming and incomplete
- Need for **automated discovery** of research trends

## Our Solution

An **Intelligent Research Assistant** that uses:

- **Topic Modeling (LDA)** to discover hidden themes
- **Sentence Transformers** for semantic search
- **Neural Networks** for topic classification

# Project Overview



## Technologies Used

**Gensim (LDA) • Sentence-Transformers (BERT) • Keras/TensorFlow (Neural Net) • Streamlit (Web App)**

# Latent Dirichlet Allocation (LDA)

## What is LDA?

A **probabilistic generative model**:

- Discovers hidden topics in text
- Documents = mixture of topics
- Topics = distribution over words

## Why LDA?

- **Unsupervised** - no labels
- **Interpretable** results
- **Scalable** to large corpora

## Mathematical Formula

$$P(w_i|d) = \sum_{t=1}^T P(w_i|t) \cdot P(t|d)$$

- $T$  = topics
- $P(w_i|t)$  = word-topic
- $P(t|d)$  = topic-doc

# Sentence Transformers

## From BERT to Sentence-BERT

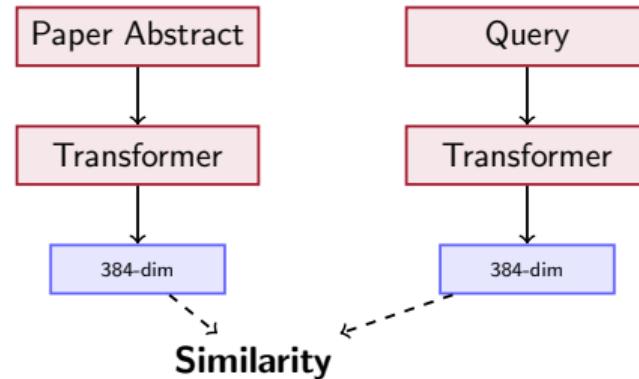
Standard BERT cross-encoding is **expensive**.

### Sentence Transformers:

- Fixed-size embeddings
- Fast cosine similarity
- Semantic search at scale

### Model: all-MiniLM-L6-v2

- 384-dim embeddings
- 1B+ training pairs
- 22M parameters



# Neural Topic Classifier

## Architecture

### A Keras Dense Network:

- Input: 384-dim embedding
- Hidden 1: 128 + ReLU + Dropout
- Hidden 2: 64 + ReLU + Dropout
- Output: Softmax over topics

## Tunable Hyperparameters

- Learning Rate
- Dropout Rate
- L2 Regularization

### neural\_classifier.py

```
model = tf.keras.Sequential([
    tf.keras.layers.Dense(
        hidden_units,
        activation='relu',
        kernel_regularizer=l2(l2_rate),
        input_shape=(384,))
    ),
    tf.keras.layers.Dropout(dropout_rate),
    tf.keras.layers.Dense(
        hidden_units // 2,
        activation='relu'
    ),
    tf.keras.layers.Dropout(dropout_rate),
    tf.keras.layers.Dense(
        num_topics,
        activation='softmax'
    )
])
```

# Evaluation Metrics

## Topic Modeling Quality

### Coherence Score ( $C_v$ )

- Measures semantic similarity of top words
- Range: 0.0 to 1.0 (higher = better)
- Good models:  $C_v \geq 0.4$

### Perplexity

- Measures how well model predicts unseen docs
- Lower = better

## Semantic Search Quality

### Cosine Similarity

$$\text{sim}(A, B) = \frac{A \cdot B}{\|A\| \|B\|}$$

- Range: -1 to 1
- Threshold:  $\geq 0.5$  for relevance

## Neural Classifier

### Accuracy on held-out test set:

- 80/20 train/test split
- Target:  $\geq 70\%$

# Data Pipeline

## 1. Data Acquisition

- **Source:** ArXiv API
- **Query:** Recent CS papers (AI, ML, Robotics)
- **Volume:** 2000+ research papers
- **Fields:** Title, Abstract, Categories, Date

## 2. Preprocessing

- Tokenization with NLTK
- Stopword removal
- Lemmatization (WordNet)
- **Bigram detection** (e.g., “machine\_learning” )

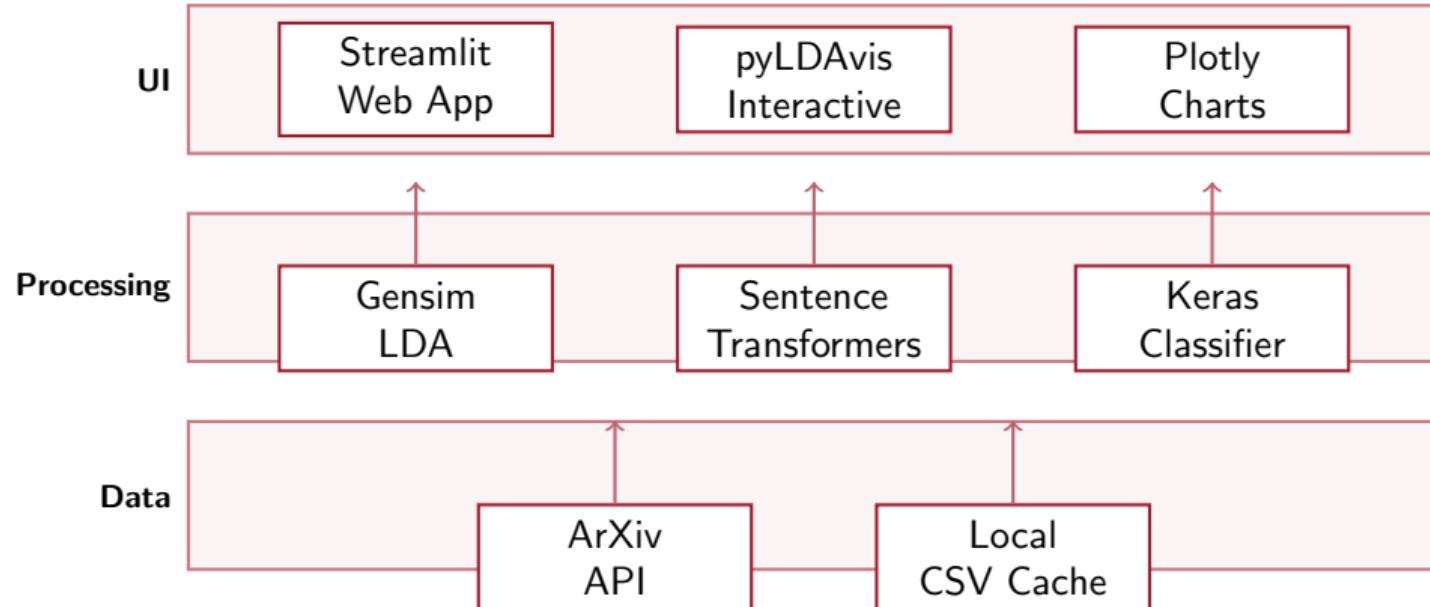
data\_loader.py

```
# Fetch from ArXiv
query = 'all:"additive\u00b7manufacturing"'
df = fetch_arxiv_papers(
    query=query,
    max_results=2000
)

# Preprocess
processed = df['abstract'].apply(
    preprocess_text
)

# Bigrams
docs = make_bigrams(processed)
```

# System Architecture



# Demo: Topic Discovery & Visualization

## Discovered Topics (ArXiv Dataset)

Topic	Top Words
0	reasoning, llm, agent, benchmark
1	data, model, performance, framework
2	policy, action, reinforcement_learning
3	dynamic, control, simulation, sensor
4	task, visual, robot, motion

## Coherence Score

$C_v = 0.4170$   
(Good topic separation)



Word Clouds for Discovered Topics

# Demo: Semantic Search & Recommendations

## Semantic Search

Search for papers using natural language. The model understands the meaning of your query, not just keywords.

Enter search query

deep learning for autonomous navigation

## Top Results

- Digitel Train Supervised Reinforcement Learning Framework for Autonomous Underwater Navigation [Score: 0.5612]

Published: 2022-12-11 18:02:40 | ID: 90008

**Abstract:** Autonomous navigation in underwater environments remains a major challenge due to the absence of GPS, degraded visibility, and the presence of submerged obstacles. This article investigates these issues through the case of the BlueROV2, an open platform widely used for scientific experiments. We propose a deep reinforcement learning approach based on the Primal Policy Optimization (PPO) algorithm, using an observation space that combines target orientation, angular velocity, and depth information. The proposed solution is evaluated in a simulated environment, the Dynamic Window Approach (DWA), commonly employed as a robust baseline for mobile robotics. The evaluation is conducted in a realistic simulation environment and complemented by validation on a physical BlueROV2 system, guided by a 3D digital twin of the test site, helping to reduce risks associated with real-world experimentation. The results show that the PPO policy consistently outperforms DWA in highly cluttered environments, notably thanks to better local adaptation and reduced collisions. Finally, the experiments demonstrate the transferability of the learned behavior from simulation to the real world, confirming the relevance of deep RL for autonomous navigation in underwater robotics.

Read PDF

- Kinetic Energy-Efficient Navigation for Surface Vehicles in Vortical Flow Fields [Score: 0.5410]

- TrajNet: Scene Adaptive Trajectory Planning with Mixture of Experts and Reinforcement Learning [Score: 0.5300]

- RICNet: An end-to-end deep learning framework for simultaneous online calibration of USAR, RADAR, and Camera [Score: 0.5132]

- A New Trajectory-Oriented Approach to Enhancing Comprehensive Crowd Navigation Performance [Score: 0.4986]

## Semantic Search Results

## AI Summarization

Uses **DistilBART** to generate concise summaries of paper abstracts.

## Smart Recommender

"If you liked this paper, you might also like..."

Select a paper from the list below, and our AI will recommend 5 other papers that are semantically similar.

Select a paper

Particulate Feed Forward 3D Object Articulation

## Selected Paper

Particulate Feed Forward 3D Object Articulation

We present Particulate, a feed-forward approach that, given a single static 3D mesh of an everyday object, directly infers all attributes of the underlying articulated structure, including its 3D parts, kinematic structure, and median constraints. At its core is a transformer network, Part Articulation Transformer, which processes a point cloud of the input mesh using a flexible and trainable architecture to predict all the aforementioned attributes with native multi-point support. We train the network end-to-end on a diverse collection of articulated 3D assets from public datasets. During inference, Particulate lifts the network's feed-forward prediction to the input mesh, yielding a fully articulated 3D mesh in seconds, much faster than prior approaches that require per-object optimization. Particulate can also accurately infer the articulated structure of AI-generated 3D assets, enabling full-fledged extraction of articulated 3D objects from a single (real or synthetic) image when combined with an off-the-shelf image-to-3D generator. We further introduce a new challenging benchmark for 3D articulation estimation curated from high-quality public 3D assets, and redesign the evaluation protocol to be more consistent with human preferences. Quantitative and qualitative results show that Particulate significantly outperforms state-of-the-art approaches.

Generate AI Summary

**All Summary:** We present Particulate, a feed-forward approach that, given a single static 3D mesh of an everyday object, infers all attributes of the underlying articulated structure. At its core is a transformer network, Part Articulation Transformer, which processes a point cloud of the

## Recommended Papers

- Sim2Real: Accurate Articulated Object Modeling from a Single Video Using Synthetic Training Data Only (Similarity: 0.79)

- SDT-4D: Fully Sparse Depth Transformer for Staged End-to-End 4D Pose Estimation in Industrial Multi-View Bin Picking (Similarity: 0.63)

## Smart Recommender Demo



# Research Trend Analysis

Data Explorer Topic Modeling Trend Analysis Research Direction Semantic Search Smart Recommender Neural Classifier

## Topic Trends Over Time



*Topic Trends Over Time (Daily)*

Data Explorer Topic Modeling Trend Analysis Research Direction Semantic Search Smart Recommender Neural Classifier

## Research Direction Dashboard

This dashboard helps identify emerging trends in the field. It calculates the growth rate of each topic over time to tell you which areas are heating up.

Emerging Topics  
2

Pending Review  
Topic 0

Total Papers Analyzed  
1003

### Topic Growth Analysis

Topic Label	Trend Status	Growth Score (Slope)	Paper Count
Topic 0: evaluation, framework, domain	Hot	0.288	301
Topic 2: agent, task, control	Hot	0.267	148
Topic 1: model, reasoning, lit	Stable	-0.08	93
Topic 4: video, image, spatial	Cooling	-0.128	7
Topic 3: ontology, metrics, translation	Cooling	-0.40	402

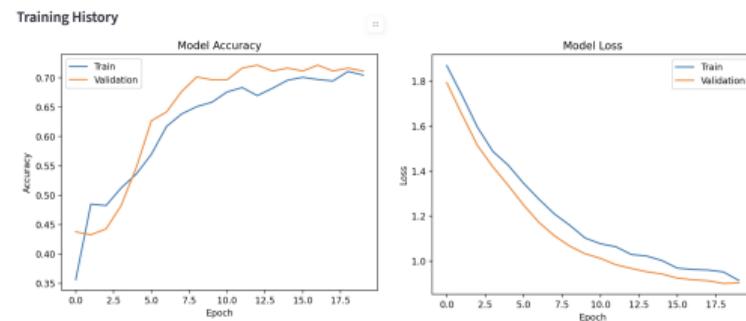
*Tip: 'Emerging' topics (Positive Slope) represent good opportunities for new research.*

*Research Direction Dashboard*

# Neural Classifier Performance

## Training Results

Metric	Value
Test Accuracy	<b>78.5%</b>
Training Epochs	20
Batch Size	32
Learning Rate	0.001
Dropout	0.3



## Key Findings

- Embeddings capture topic semantics well
- L2 regularization prevents overfitting
- Dropout improves generalization

*Training & Validation Curves (1000 papers)*

# Pros, Cons & Future Work

## Pros

- **End-to-end pipeline:** Data to insights
- **Interactive:** Streamlit web interface
- **Reproducible:** Clear documentation
- **Extensible:** Modular architecture

## Cons

- LDA requires manual topic count tuning
- API rate limits for large datasets
- Classifier accuracy could improve

## Future Work

- Use **BERTopic** for neural topic modeling
- Add **citation network** analysis
- Implement **real-time** paper alerts
- Fine-tune **domain-specific BERT**

# Thank You!

Questions?



Code: Search Engine Educational Project



Contact: [shasanov@g.harvard.edu](mailto:shasanov@g.harvard.edu)