

Generative Language Models

PROJECT 3

DEVELOPMENT OF A CASE STUDY

WITH MODEL EVALUATION

CRISP-DM Phases 1-5

Table of Contents

1. Business Understanding.....	4
1.1 Problem Definition	4
1.2 Stakeholders.....	4
1.3 Value Creation	4
1.4 Model Performance and Practical Value.....	5
1.5 Business Understanding Summary	5
2. Data Understanding.....	6
2.1 ArXiv Scientific Articles Dataset	6
2.2.1 Initial Data Load Interpretation	6
2.2 Dtypes and Missing Values Analysis.....	7
2.2.1 Data Types, Missing Values, and Cardinality – Interpretation	7
2.3 Category Class Distribution	8
2.3.1 Distribution of Primary Subjects - Scientific Domain Imbalance	8
2.4 Abstract Length Analysis	9
2.4.1 Abstract Length - Informational Richness for NLP	9
2.5 Inspect and Clean submission_date	10
2.5.1 Submission Date – HTML Noise and Normalization	10
2.6 Datetime Convert and Year Distribution's Inspection.....	11
2.6.1 Temporal Coverage - Evolution of Scientific Submissions.....	11
2.6.2 Cumulative Coverage of primary_subject.....	12
2.6.3 Class Concentration - top-k Primary Subjects.....	12
2.6.4 Compare primary_subject vs subjects.....	13
2.6.5 Primary Subject vs. Full Subject Tagging.....	13
2.6.6 Number of Subject Tags per Paper	14
2.6.7 Number of Authors Per Paper	15
2.6.8 Number of Authors per Paper	15
2.6.9 Sample Titles and Abstracts by primary_subject.....	16
2.6.10 Qualitative Look at Titles and Abstracts Across Subjects.....	16
2.6.11 Title and Abstract Word Length Analysis	17
2.6.12 Plots with Highlighted Peak Frequency Bins	18
2.6.13 Title and Abstract Word Length Distributions	18
2.6.14 Top 20 Per-Subject Profile.....	19
2.6.15 Subject-Level Profile	19
2.6.16 Mean Abstract Length by Primary Subject	20
2.7 Inspect Text Artifacts in Abstracts	21

2.7.1 Text Artefacts in Abstracts	21
2.8 Impact of Cleaned Text Fields for Title and Abstract.....	22
2.8.1 Remarks on Original vs. Cleaned Text Fields.....	22
2.9 Linguistic Patterns in Abstracts	22
2.9.1 Linguistic Artefacts in Abstracts	22
2.10 Main Scientific Domains in primary_subject.....	23
2.10.1 Distribution of Main Scientific Domains.....	23
2.11 Top Unigrams and Bigrams in Abstracts.....	24
2.11.1 Dominant Vocabulary in Abstracts	24
2.12 Heuristic Detection of Truncated Abstracts.....	25
2.12.1 Potentially Truncated Abstracts	25
2.13 Unigram Word Cloud From Cleaned Abstracts	26
2.14 Bigram Word Cloud from Cleaned Abstracts	27
2.15 Data Understanding Summary	28
Document structure.....	28
Presence of LaTeX, HTML, and special characters	28
Scientific domains	28
Vocabulary and linguistic patterns.....	28
Key implications for Data Preparation.....	28
3. Data Preparation	29
3.1 Context.....	29
3.2 Data Preprocessing	29
4. Modeling	30
4.1 Introduction	30
4.2 SciBERT's Architecture.....	31
5. Evaluation	31
5.1 Accuracy and Loss	31
5.2 Confidence Analysis.....	32
5.3 Confusion Matrices.....	32
6. Bibliography and Supporting Sources	32

1. Business Understanding

1.1 Problem Definition

The volume of academic literature published every day continues to grow across all fields of study. Each new paper introduces a unique contribution, whether a novel solution, a methodological innovation, or a theoretical analysis requiring accurate categorization to be discoverable and useful.

Traditionally, papers are classified according to their content, methodology, contribution type, and research domain (e.g., empirical studies, reviews, theoretical work, descriptive or analytical approaches).

However, as publications scale in both volume and interdisciplinarity, manual categorization has become increasingly impractical, inconsistent, and time-consuming. The complexity of modern research means that many papers naturally span multiple categories, for instance, mathematical modeling within physics or machine learning applied to social sciences. Traditional single-label or Softmax-based classification systems are inadequate for this reality because they force a paper into a single dominant category or enforce normalized probabilities across categories.

In addition to scalability challenges, classification inconsistency among human reviewers poses a significant problem. Different experts may categorize the same paper differently or attribute varying levels of relevance to certain domains. Such subjectivity reduces reliability and introduces bias into bibliographic systems. Our model seeks to address this challenge by producing standardized, repeatable, and objective multi-label classifications, reducing human disagreement and enabling a consistent categorization framework across large-scale repositories.

The increasing complexity and volume of academic literature creates a clear need for an automated system capable of providing accurate, multi-label, domain-independent categorization at scale. Such a system must not only identify the primary category of paper but also assign independent relevance probabilities across multiple fields.

1.2 Stakeholders

Researchers and Academics - Primary users who rely on accurate categorization to quickly find relevant literature. - Benefit from reduced time spent searching through poorly organized or uncategorized papers.

Librarians and Information Specialists - Currently responsible for manually classifying and organizing academic publications. - Gain efficiency through automation and improved consistency in taxonomy management.

Research Institutions and Universities - Need to maintain organized repositories of academic output. - Benefit from streamlined workflows, increased discoverability of internal research, and reduced operational costs.

1.3 Value Creation

Our solution automates the multi-label classification of academic papers. Unlike traditional Softmax-based classifiers, which distribute probabilities across categories so that they sum to 1, our model assigns:

- A separate, independent probability to each category
- No normalization constraints a paper can strongly belong to multiple fields
- Flexible relevance scores that reflect real interdisciplinary relationships

This approach mirrors human judgment more accurately than mutually exclusive classification, while eliminating human inconsistency and subjectivity.

By standardizing and automating classification, the system reduces manual labor, improves accuracy and consistency, and increases the discoverability of relevant research. It also frees researchers and librarians to perform higher-value analytical tasks instead of repetitive sorting activities.

1.4 Model Performance and Practical Value

Classifying academic papers across 148 distinct categories is inherently challenging due to the multidisciplinary and nuanced nature of research. Even without perfect accuracy, a validated model can serve as an asset by:

- Providing a first-level clustering or triage, helping human reviewers focus their efforts more efficiently.
- Reducing initial workload by organizing large volumes of documents into broad thematic groups.
- Laying the groundwork for more refined analysis, as its output can be further validated or adjusted by experts.

This approach not only streamlines the review process but also creates a structured foundation for deeper exploration, setting the stage for insights that will be detailed in the final report.

1.5 Business Understanding Summary

The rapidly growing volume and complexity of academic publications make manual categorization increasingly inefficient, inconsistent, and error prone. Papers often span multiple disciplines, but traditional single-label or Softmax based classification systems force them into narrow categories, failing to capture their interdisciplinary nature. Human reviewers also introduce subjectivity, leading to inconsistent labeling across repositories.

The proposed solution is an automated multi-label classification system that assigns independent relevance probabilities to each of 148 academic categories. Unlike traditional methods, it does not normalize probabilities, allowing papers to strongly belong to several fields simultaneously. This creates more accurate, objective, and repeatable classifications that reflect real interdisciplinary research.

Key stakeholders including researchers, librarians, and academic institutions benefit from improved discoverability, reduced manual workload, greater consistency, and lower operational costs. Even without perfect accuracy, the model provides valuable first level clustering, streamlining document organization and enabling experts to focus on higher value tasks. This automated system establishes a scalable, reliable foundation for managing large academic repositories.

2. Data Understanding

2.1 ArXiv Scientific Articles Dataset

Perform an in-depth Data Understanding and Exploratory Data Analysis (EDA) on a large corpus of scientific articles extracted from arXiv ($\approx 2.55M$ records). The goal of this stage is to thoroughly examine the characteristics, structure, and quality of the dataset before building any machine learning or language model-based classifier.

Objectives of this Data Understanding stage:

1. Inspect the dataset structure

- a. Review columns, datatypes, and the general schema
- b. Verify consistency of key fields (title, abstract, subjects, authors)

2. Assess data quality

- a. Identify missing, inconsistent, or malformed entries
- b. Detect anomalies in abstracts, dates, author lists, and subject labels

3. Explore distributions and content characteristics

- a. Distribution of scientific subjects
- b. Abstract length statistics and text quality indicators
- c. Unique values, duplication analysis, and potential noise sources

4. Establish the suitability of the data for downstream tasks

- a. Validate the feasibility of scientific article classification using titles and abstracts
- b. Understand challenges such as class imbalance, text variability, and metadata inconsistencies

2.2.1 Initial Data Load Interpretation

The dataset successfully loaded with a total of **2,549,619 records** and **10 columns**, indicating a large-scale scientific metadata collection suitable for NLP-driven classification tasks.

The first fields inspection confirms the presence of the main elements expected for arXiv articles:

Table 2.1 - First fields inspection

Column	Observation
arxiv_id	Unique identifier present in all previewed rows.
title	Informative scientific titles, typically long and domain specific.
authors	Stored as a stringified list, indicating a need for parsing if used for author-level analysis.
submission_date	Values containing embedded HTML markup (e.g.,), which require cleaning.
comments	Heterogeneous notes not consistently populated: require filtering or optional use.
primary_subject/ subjects	Core labels for classification. Early inspection suggests clean and readable subject tags.
doi	Contains DOIs for traceability, though likely not complete for all entries.
abstract	Core textual field for modeling. Appears well populated in the sample.
file_path	Points to original PDF storage, enabling future full-text extraction if needed.

Overall, the dataset structure is consistent with expectations for arXiv metadata and appears **well-suited for downstream modeling** - particularly classification tasks based on title and/or abstract. The next steps will include:

- Inspecting missing values and data types.
- Assessing class distribution for primary_subject.
- Exploring abstract length variability to validate text richness for LLM tasks.

2.2 Dtypes and Missing Values Analysis

2.2.1 Data Types, Missing Values, and Cardinality – Interpretation

The info() summary shows that all **10 columns are stored as object type**, which is typical for text-heavy datasets but also indicates that:

- Dates (e.g., submission_date) are not yet parsed as proper datetime objects.
- Structured fields like authors and subjects are currently represented as generic Python objects or stringified lists and may require custom parsing for deeper analysis.

A key observation is that **there are no missing values reported in any column (0.0% missing)**. While this looks ideal at first glance, it also suggests that: - Any apparent “absence” of information is likely encoded as empty strings, minimal text, or placeholder content rather than NaN. - Data quality issues, if present, will be more semantic (e.g., noisy text, HTML fragments) than structural (nulls).

Regarding cardinality (number of distinct values):

- arxiv_id and file_path are **fully unique**, matching the total number of rows – each record corresponds to a unique article and PDF path.
- doi is almost fully unique as well (2549556 unique values vs 2,549,619 rows), which is consistent with each published article being mapped to a specific DOI, with a small fraction possibly missing or duplicated.
- title and abstract also show **very high uniqueness**, confirming that they can be treated as instance-level descriptors, ideal for text-based classification and retrieval tasks.
- primary_subject has **only 148 unique values**, which aligns well with its role as a **categorical label** for a multi-class classification problem.
- subjects have **85,221 unique combinations**, which reflects the multidimensional tagging of arXiv papers (multiple subjects per article). It is more complex to work with as a label space and may require simplification (e.g., using only primary_subject or deriving higher-level domain groupings).
- The authors’ column is reported as “*unhashable/array-like → needs custom handling*”, confirming that it likely contains list-like structures (arrays or Python lists). This is important for future work on collaboration networks or author-level statistics but is **not immediately suitable** for standard groupby operations without preprocessing.

2.3 Category Class Distribution

2.3.1 Distribution of Primary Subjects - Scientific Domain Imbalance

The primary_subject field contains **148 distinct scientific categories**, reflecting the breadth of the arXiv taxonomy. However, the distribution is **highly imbalanced**, with a small set of domains concentrating a large fraction of all papers.

The top categories include:

- **Computer Vision and Pattern Recognition (cs.CV)** – 120,122 papers (~4.71%)
- **Quantum Physics (quant-ph)** – 120,051 (~4.71%)
- **High Energy Physics – Phenomenology (hep-ph)** – 116,920 (~4.59%)
- **Machine Learning (cs.LG)** – 106,484 (~4.18%)
- **High Energy Physics – Theory (hep-th)** – 93,277 (~3.66%)

Together, just these five labels account for more than **one-fifth of the entire dataset**, highlighting a strong concentration in a few research communities such as computer vision, machine learning, quantum physics, and high-energy physics.

Beyond the top group, we see substantial representation of:

- Condensed matter subfields (e.g., *cond-mat.mes-hall*, *cond-mat.mtrl-sci*, *cond-mat.str-el*),
- Core mathematics categories (e.g., *math.AP*, *math.CO*, *math.PR*, *math.AG*),
- Astrophysics subdomains (e.g., *astro-ph.GA*, *astro-ph.SR*, *astro-ph.HE*, *astro-ph.CO*),
- Information theory and related CS-theory areas (e.g., *cs.IT*).

This pattern confirms that:

- The classification task is a **multi-class problem with a long-tailed label distribution**.
- Any model trained to predict primary_subject must be evaluated with **metrics that are robust to class imbalance**, such as macro-average F1, rather than overall accuracy alone.
- It may be beneficial to:
 - focus on a subset of high-frequency subjects for an initial prototype,
 - or group fine-grained arXiv categories into higher-level super-domains (e.g., “Computer Science”, “Physics”, “Mathematics”) to ease the learning problem and improve interpretability.

From a data understanding perspective, this distribution strongly motivates a careful **design of the label space and evaluation strategy** before training classification models.

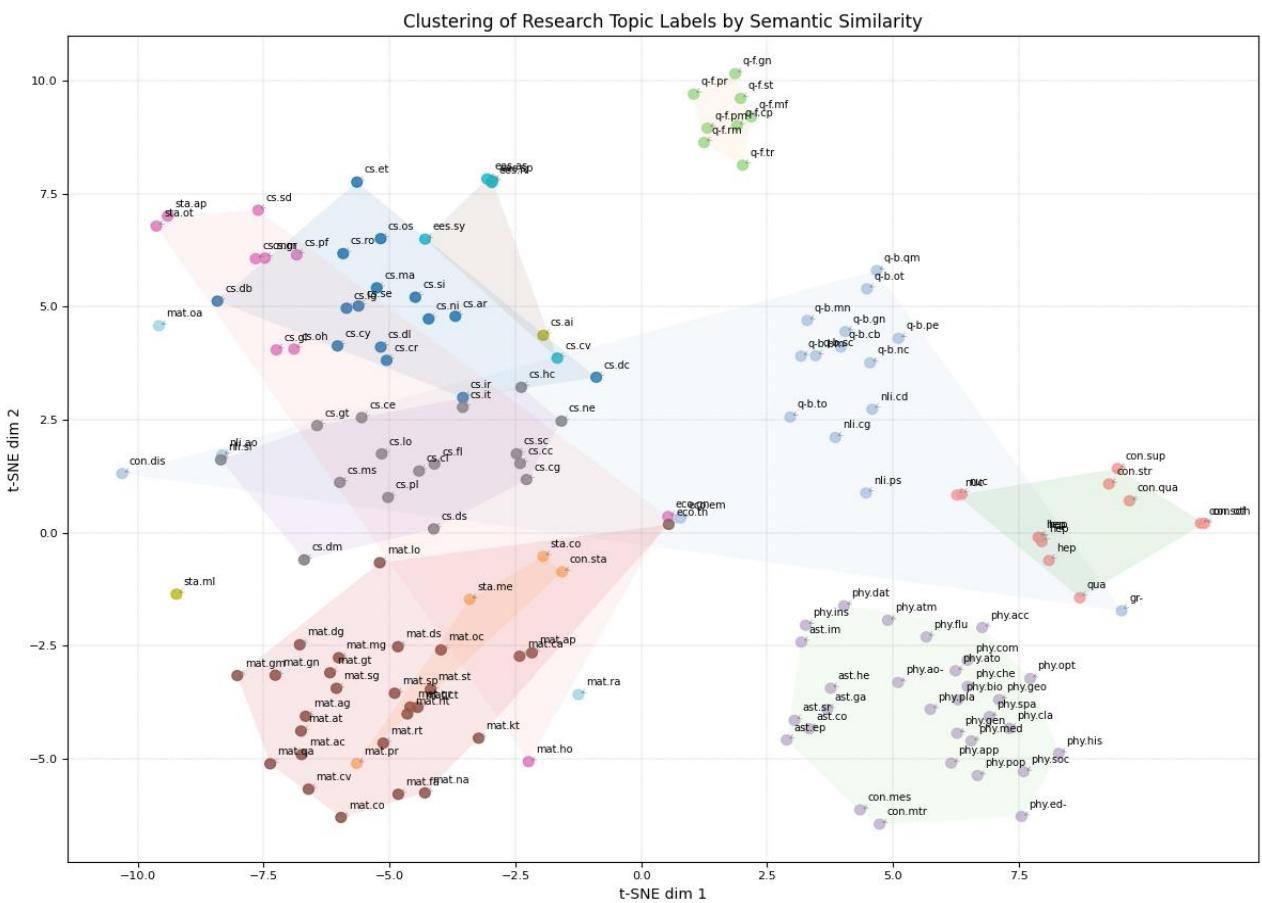


Figure 2.1 - Clustering of Research Topic Labels by Semantic Similarity

2.4 Abstract Length Analysis

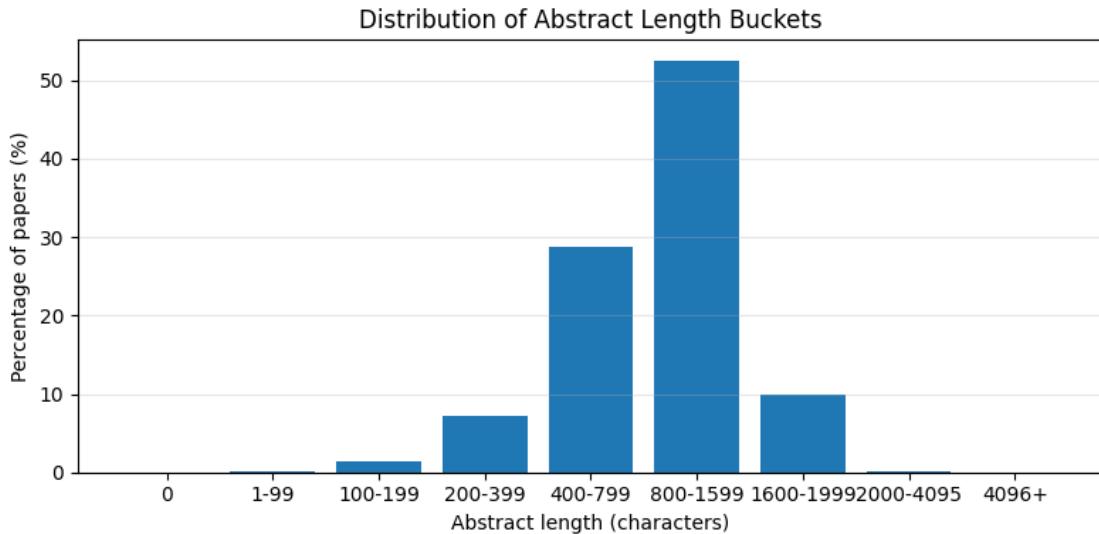


Figure 2.2 - Distribution of Abstract Length Buckets

2.4.1 Abstract Length - Informational Richness for NLP

The abstract_char_len statistics confirm that the dataset provides **substantial textual content** for each paper:

- Median length \approx **956 characters** and mean \approx **983 characters**, with a standard deviation of \approx **436**.

- The central 50% of abstracts lie between **644** (25th percentile) and **1,297** characters (75th percentile).
- Very short abstracts are rare: the 1st percentile is already at **167** characters.
- The maximum length reaches **6,554 characters**, but such extremely long abstracts are exceptional.

The bucketed distribution further clarifies the picture:

Table 2.2 - Dataset bucket distribution

Bucket	Share of dataset
400–799	28.71%
800–1,599	52.52%
1,600–1,999	9.89%
< 200	~1.61% (combined)
≥ 2,000	~0.13%

This indicates that:

- **Over 80%** of the abstracts fall in the range **400–1,600 characters**, which aligns well with typical scientific abstracts (roughly 150–300 words).
- **Extremely short abstracts (< 200 characters)** are rare and may represent edge cases, such as minimal descriptions, incomplete records, or noisy entries. They can be flagged for potential exclusion or special handling in robustness analyses.
- **Very long abstracts (≥ 2,000 characters)** are also extremely rare and unlikely to dominate model behavior, though they may slightly increase computational cost.

Overall, the distribution strongly supports the use of abstract as a **primary textual feature** for downstream NLP and LLM-based classification: the majority of records provide sufficiently detailed, informative text without being excessively long for modern language models.

2.5 Inspect and Clean `submission_date`

2.5.1 Submission Date – HTML Noise and Normalization

The raw `submission_date` field shows two distinct formats:

- **Clean dates**, such as: 3 Feb 2009, 22 Jan 2009, 12 Jan 2009.
- **Dates with embedded HTML and revision metadata**, for example: 18 Feb 2009 (v1), last revised 18 Jun 2009 (this version, v2)

Approximately **37.53% of all records** contain HTML anchor tags (<a ...>) inside `submission_date`, which means the field mixes: - the initial submission date, - version links, - and “last revised” information.

For most modeling tasks, especially classification, we only need a **single reference submission date** per paper. The `submission_date_clean` field addresses this by:

1. Stripping out any HTML content using a regular expression,
2. Keeping only the leading date portion before any parentheses.

Examples confirm that:

- 18 Feb 2009 (...) → 18 Feb 2009.
- 30 Jan 2009 (...) → 30 Jan 2009.
- Already clean dates (e.g. 3 Feb 2009) remain unchanged.

This normalized string column (submission_date_clean) is now suitable for conversion into a proper datetime type, enabling temporal analyses such as: - yearly submission trends, - time-based stratification, - or temporal drift analysis in subject distribution.

2.6 Datetime Convert and Year Distribution's Inspection

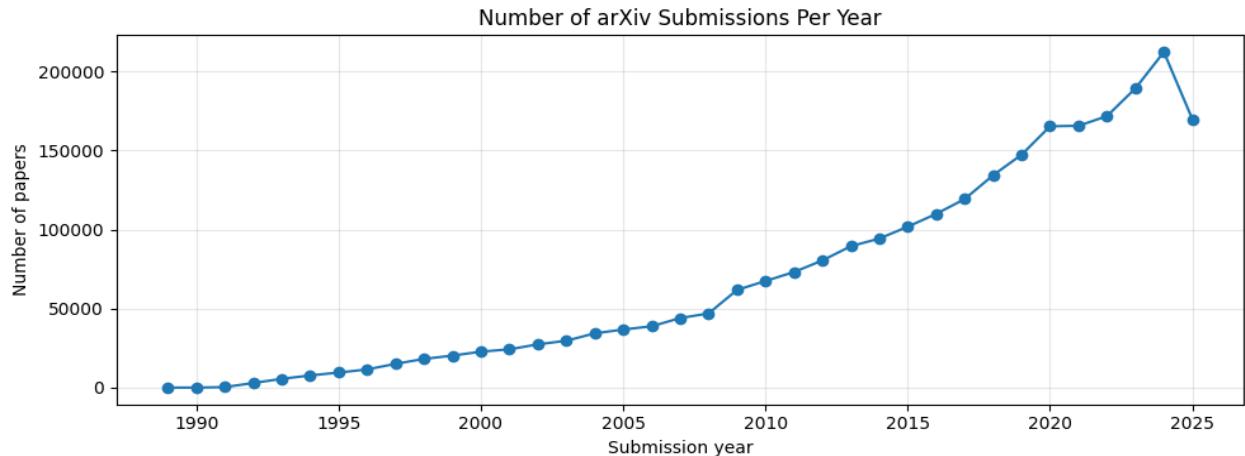


Figure 2.3 - Number of arXiv Submissions Per Year

2.6.1 Temporal Coverage - Evolution of Scientific Submissions

After cleaning the submission_date field and converting submission_date_clean into a proper datetime object (submission_datetime), all records were successfully parsed:

- **0 rows** failed the date conversion (0.00% NaT).
- The temporal range spans from **1989-11-17** to **2025-09-25**, covering more than three decades of scientific activity.

The yearly submission figures reveal a clear long-term growth trend:

- Early years (1989–1992) show very low volume (from 1 to a few thousand papers per year), reflecting the initial stages of arXiv as a preprint server.
- From the mid-1990s onwards, there is a consistent increase in yearly submissions, with steady growth across the 2000s.
- The 2010s mark a strong expansion phase, with counts surpassing **100,000 submissions per year** around the mid-decade.
- Recent years exhibit peak activity:
 1. **2018–2019** already exceed 130k–140k submissions,
 2. **2020–2022** surpass **160k–170k**,
 3. **2023** reaches **189,381** submissions,
 4. **2024** peaks at **212,376** submissions,
 5. **2025** (partial year) already accumulates **169,873** submissions up to late September.

This pattern confirms that:

1. The dataset is **dominated by recent submissions**, which is expected given the increasing popularity of arXiv across many disciplines (e.g., machine learning, computer vision, condensed matter, astrophysics).
2. Any modeling effort that samples uniformly across the full dataset will naturally emphasize **modern scientific language and topics**, especially from the last decade.
3. If desired, one could:
 - a. restrict experiments to a specific time window (e.g., 2010 onwards),
 - b. or explicitly study **temporal drift** in subject distributions and language usage over time.

From a data understanding perspective, the temporal structure is clean, well-covered, and aligns with the known expansion of arXiv as a central platform for scientific dissemination.

2.6.2 Cumulative Coverage of primary_subject

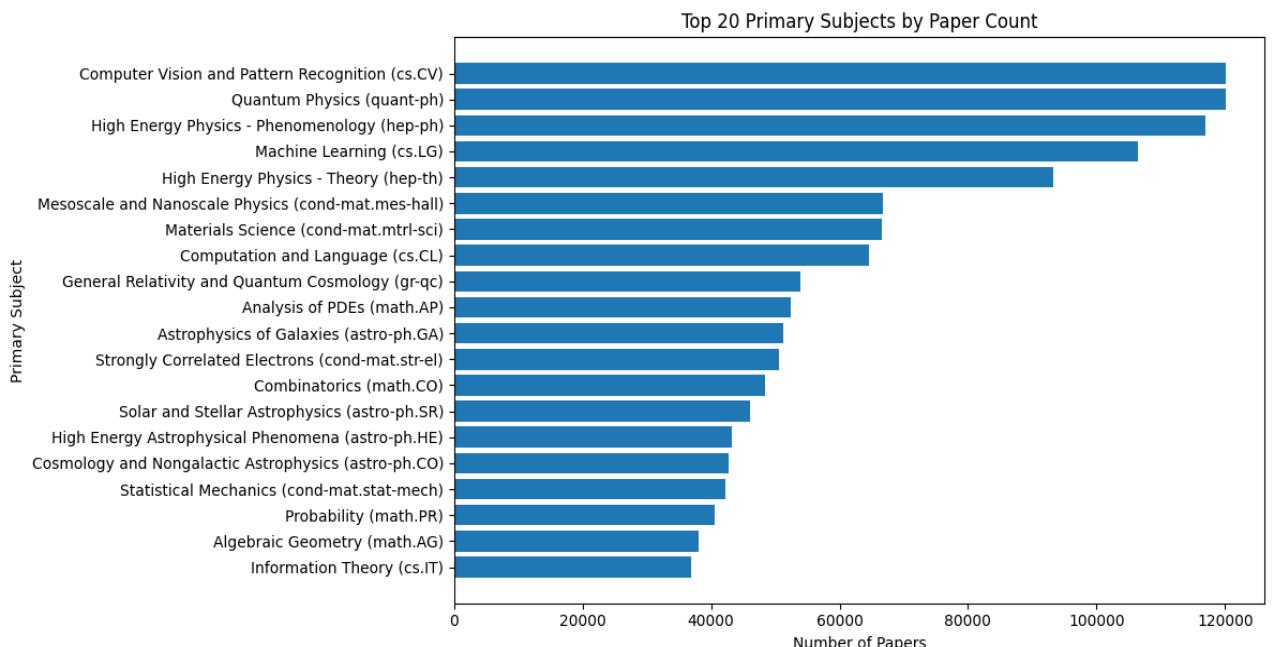


Figure 2.4 - Top 10 Primary Subjects by Paper Count

2.6.3 Class Concentration - top-k Primary Subjects

The analysis of primary_subject frequencies confirms a **strongly imbalanced label space**. Among the 148 distinct categories, only a small subset concentrates a large proportion of all papers.

From the cumulative distribution:

- The **top 10** subjects already cover **≈33.8%** of all records.
- The **top 20** subjects cover **≈51.0%** of the dataset.
- The **top 50** subjects cover **≈78.7%** of all papers.

The horizontal bar chart of the **Top 20 primary subjects** highlights this pattern visually: a handful of domains dominate the corpus, notably:

- Computer Vision and Pattern Recognition (cs.CV),
- Quantum Physics (quant-ph),

- High Energy Physics – Phenomenology (hep-ph),
- Machine Learning (cs.LG),
- High Energy Physics – Theory (hep-th),
- Condensed Matter subfields,
- Core mathematics categories (e.g., PDEs, combinatorics, probability),
- Astrophysics subdomains,
- Information Theory and related areas.

This has several implications for downstream modeling:

- The classification problem is **multi-class with a long tail**: many labels appear rarely and will be hard to model reliably with standard supervised approaches.
- Evaluation using **only accuracy would be misleading**, since a model could perform well on dominant classes while ignoring minority ones. Metrics such as **macro-F1** and per-class analysis are more appropriate.
- For a practical LLM-based case study, it may be beneficial to:
 1. restrict the problem to the **top-k most frequent labels** (e.g., Top 20 or Top 30), achieving a balance between coverage and tractability, or
 2. group fine-grained arXiv categories into a smaller set of **broader scientific domains**, simplifying both training and interpretation.

This distribution strongly motivates a **careful design of the label space** before training or evaluating any classification model on this dataset.

2.6.4 Compare primary_subject vs subjects

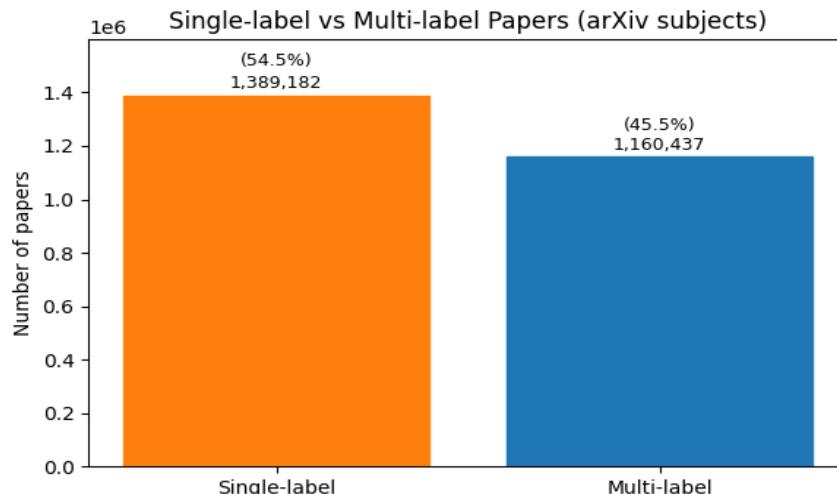


Figure 2.5 - Single-Label vs multi-label (arXiv subjects)

2.6.5 Primary Subject vs. Full Subject Tagging

The comparison between primary_subject and subjects reveals that arXiv uses a **hybrid labeling scheme**:

- In **54.49%** of the records, subjects is **identical** to primary_subject. In these cases, the paper is effectively associated with a **single subject label**.

- In **45.51%** of the records, subjects **differ** from primary_subject. Here, the article is tagged with **multiple subjects**, with primary_subject acting as the main category and subjects providing additional cross-disciplinary labels.

Examples where they differ show patterns such as:

- A primary label in *Earth and Planetary Astrophysics (astro-ph.EP)* combined with secondary tags like:
 - *Solar and Stellar Astrophysics (astro-ph.SR)*,
 - *Astrophysics of Galaxies (astro-ph.GA)*,
 - *Geophysics (physics.geo-ph)*,
 - *Biological Physics (physics.bio-ph)*,
 - *Populations and Evolution (q-bio.PE)*,
 - *Instrumentation and Methods for Astrophysics (astro-ph.IM)*.

This confirms that:

- primary_subject is a **single-label categorical target** suitable for standard multi-class classification.
- subjects encode a **richer, multi-label structure**, which could support more advanced tasks such as:
 1. multi-label classification,
 2. interdisciplinarity analysis,
 3. or recommendation systems based on overlapping subject tags.
- For an initial, well-scooped classification study, it is reasonable to **focus on primary_subject as the main label**, while keeping subjects as an optional extension for future experiments in multi-label modeling.

Understanding this distinction is crucial to clarifying the target definition and avoiding confusion between single-label and multi-label problem formulations.

2.6.6 Number of Subject Tags per Paper

The num_subject_tags feature quantifies how many labels each paper receives in the subject field.

Key descriptive statistics:

- Mean ≈ **1.65** tags per paper
- Median = **1** tag
- 75th percentile = **2** tags
- Maximum = **11** tags

The discrete distribution shows:

- **1 tag**: 1,389,182 papers (**54.49%**)
- **2 tags**: 785,941 papers (**30.83%**)
- **3 tags**: 280,778 papers (**11.01%**)
- **4 or more tags**: 93,718 papers (**3.68%**)

This confirms that:

1. Most papers are **single labelled** in practice, with just over half having only one subject tag.
2. A large fraction (~31%) has **two subject tags**, reflecting moderate interdisciplinarity.
3. Only a small minority of papers have **three or more** tags, and extreme cases with ≥ 5 tags are rare.

From a modeling perspective, this supports the choice of primary_subject as a **single-label target** for the core classification task, while subjects can be used later to explore **multi-label extensions** or to analyze cross-disciplinary patterns in the corpus.

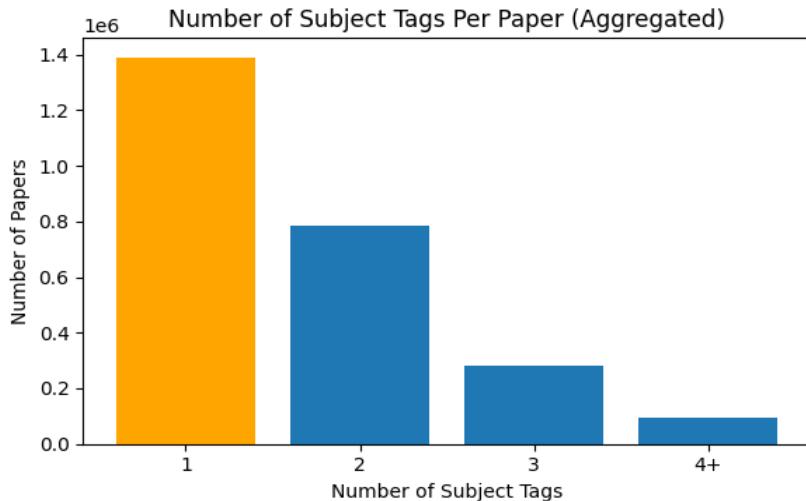


Figure 2.6 - Number of Subject Tags Per Paper

2.6.7 Number of Authors Per Paper

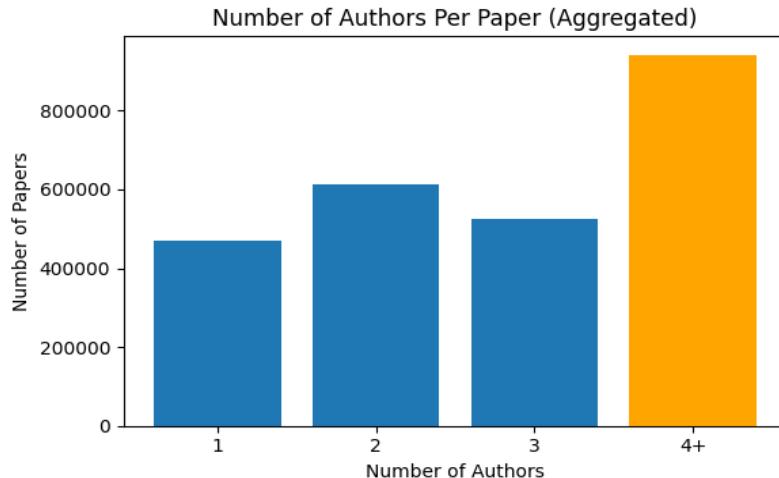


Figure 2.7 - Number of Authors Per Paper

2.6.8 Number of Authors per Paper

The num_authors feature captures the size of the author team for each article. The first rows confirm that authors is stored as an array-like object, and the counting function now correctly handles those structures.

Key statistics:

- Mean ≈ **4.71** authors per paper.
- Median = **3** authors.
- 25th percentile = **2** authors.
- 75th percentile = **4** authors.
- Maximum = **3,301** authors (extreme large-collaboration papers, typical in high-energy physics and similar fields).

The discrete counts for small team sizes are:

- 1 author: **469,799** papers (**18.43%**)
- 2 authors: **611,577** papers (**23.99%**)
- 3 authors: **526,695** papers (**20.66%**)
- 4 or more authors: **941,545** papers (**36.93%**)

This indicates that:

1. Single-author papers are relatively common but not dominant.
2. Most of the corpus consists of **small to medium-sized collaborations (2–4 authors)**.
3. A substantial fraction (~37%) involves **larger teams (4+ authors)**, reflecting big collaborations that are typical in certain scientific domains.

From a modeling perspective, the number of authors is **not directly needed** for text classification but can be used for: - descriptive statistics about collaboration patterns across subjects, - potential correlation analyses between team size and subject area, - or as an optional metadata feature in extended models.

2.6.9 Sample Titles and Abstracts by primary_subject

2.6.10 Qualitative Look at Titles and Abstracts Across Subjects

The sampled titles and abstracts illustrate clear stylistic and lexical differences across primary_subject categories:

- **Computer Vision and Pattern Recognition (cs.CV)**

The language is highly technical but application-driven, focusing on concrete tasks such as image inpainting, data efficiency, and text-to-video generation. Abstracts frequently reference:

1. deep learning architectures (CNNs, transformers, diffusion models),
2. practical challenges (data requirements, visual quality, real-time constraints),
3. and benchmark-style evaluation.

- **Machine Learning (cs.LG)**

Abstracts tend to emphasize:

1. general-purpose learning frameworks (meta-RL, actor-critic, world models),
2. theoretical or algorithmic contributions (off-policy learning, constrained inference),
3. and broad applicability across tasks or hardware platforms. The style is a mix of theoretical and system-oriented writing, often highlighting scalability and generalization.

- **Quantum Physics (quant-ph)**

Here the language shifts toward:

1. highly specialized physical concepts (holonomy, exceptional points, spectral gaps, entangled photons),
2. mathematical formalisms (Riemann surfaces, renormalization methods),
3. and experimental setups (Hong–Ou–Mandel interferometers). Abstracts are dense, concept-heavy, and less focused on “datasets” or “benchmarks”, reflecting a more traditional theoretical physics style.

- **High Energy Physics – Phenomenology (hep-ph)**

Abstracts are strongly domain-specific, referring to:

1. supersymmetric inflation, heavy-quark expansions, beauty hadron lifetimes,
2. particle production cross sections and collider processes. The writing is technical, formula-rich (often including LaTeX notation), and closely aligned with phenomenological modeling and experimental observables.

These qualitative differences support the idea that title and abstract contain enough **semantic signals** for a model to distinguish between scientific areas. At the same time, they highlight the challenge for any classifier (including LLMs) to handle very heterogeneous writing styles, from application-oriented ML papers to mathematically dense quantum and high-energy physics articles.

2.6.11 Title and Abstract Word Length Analysis

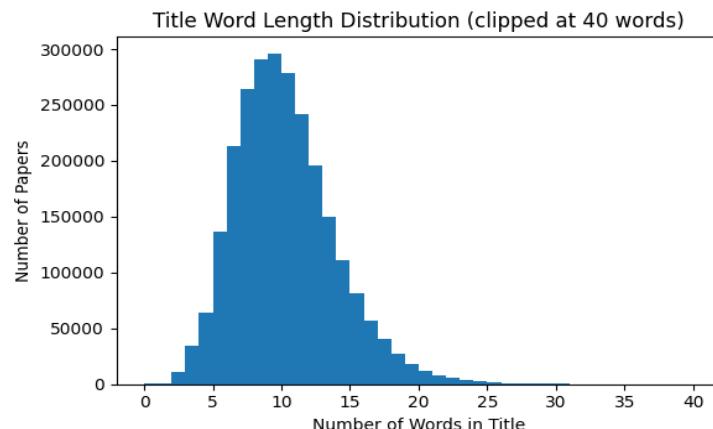


Figure 2.8 - Title Word Length Distribution

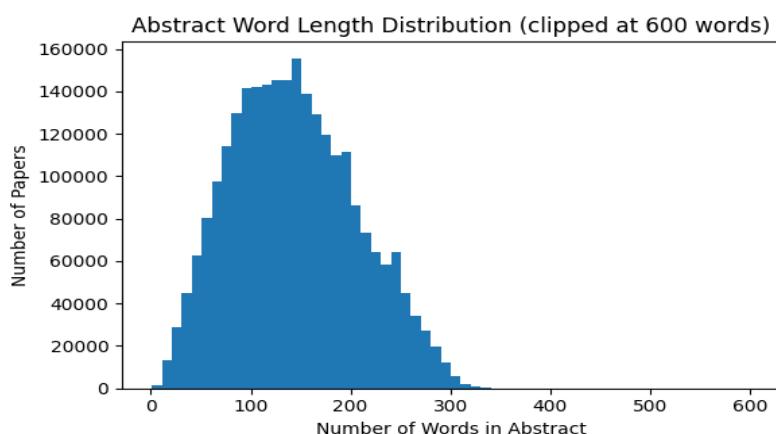


Figure 2.9 - Abstract Word Length Distribution

Title Word Length Distribution:

- The distribution is right-skewed, concentrated mainly between 6 and 14 words.
- Median title length = 9 words, mean \approx 9.79 words.
- Titles longer than 20 words are rare, and extremely short titles (below 3 words) are also uncommon.
- This suggests typical scientific titles in arXiv are concise, descriptive, and optimized for searchability.

Abstract Word Length Distribution:

- Abstracts show a wider spread, centered approximately between 90 and 200 words.
- Median abstract length = 139 words, mean \approx 143 words.
- Few cases exceed 300+ words, indicating informal adherence to typical journal/conference limits.
- Most abstracts appear sufficiently long to convey motivation, method, and contribution without excessive verbosity.

These characteristics are highly relevant for downstream text modeling, token budgeting, embedding cost estimation, and classification modeling, especially when working with LLMs that have input size constraints.

2.6.12 Plots with Highlighted Peak Frequency Bins

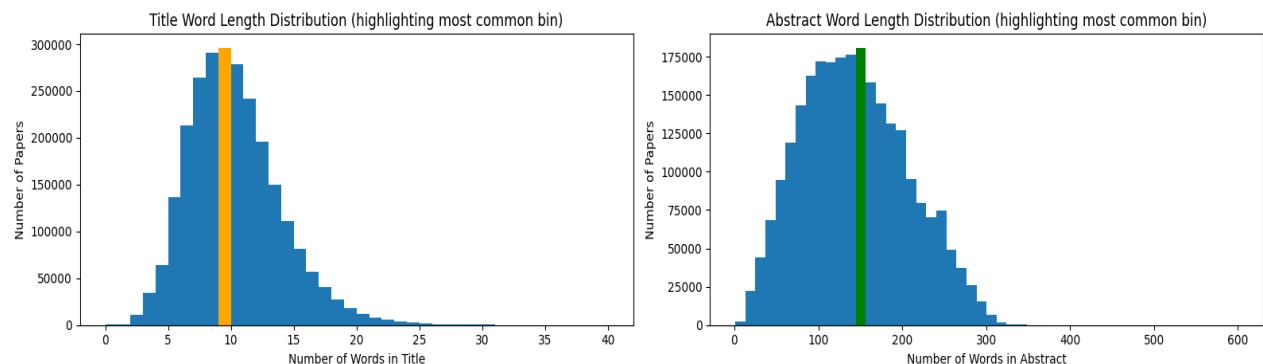


Figure 2.10 - Title and Abstract Word Length Distributions

2.6.13 Title and Abstract Word Length Distributions

The first histogram shows the distribution of **title length in words**, with the most frequent bin highlighted in orange.

Most titles cluster tightly around **9–10 words**, confirming that arXiv titles are generally short, dense and highly informative. Very short titles (< 4 words) and very long ones (> 20 words) are rare, which suggests a standardized “style” across scientific fields.

The second histogram shows the **abstract length in words**, with the most frequent bin highlighted in green.

The peak lies around **140–160 words**, and the bulk of abstracts fall roughly between **100 and 200 words**. Longer abstracts (> 300 words) are uncommon, and extremely short abstracts are also rare. This indicates that abstracts typically provide a reasonably rich description (motivation, method, results) while remaining within the usual constraints of scientific publishing.

For the classification task, these distributions imply that:

- Titles provide **compact, high-signal context** (≈ 10 tokens), useful as an auxiliary feature.
- Abstracts are **consistently long enough** to support robust text-based models, but still short enough to fit comfortably within standard LLM context windows.

2.6.14 Top 20 Per-Subject Profile

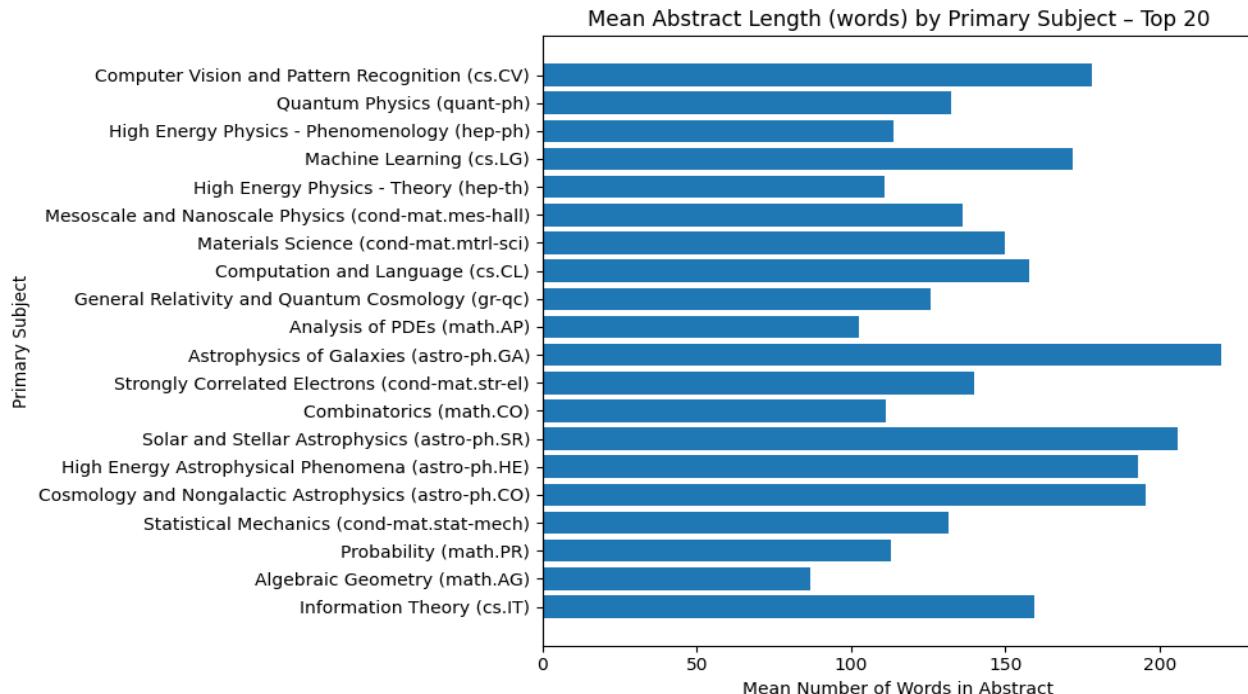


Figure 2.11 - Top 20 of Mean Abstract Word Length by Primary Subject

2.6.15 Subject-Level Profile

For the 20 most frequent primary_subject categories, we computed the mean title length, abstract length and number of authors per paper.

Several clear patterns emerge:

- **Computer Vision (cs.CV) and Machine Learning (cs.LG)**
 - Long abstracts on average (≈ 178 and ≈ 172 words).
 - Moderate title length (≈ 9 – 10 words).
 - Relatively large author teams (≈ 4 – 5 authors per paper). These fields match the “modern ML paper” style: detailed abstracts and collaborative work.
- **Quantum Physics (quant-ph) and High Energy Physics – Theory/Phenomenology (hep-th, hep-ph)**
 - Abstracts are shorter than ML/CV on average (≈ 111 – 133 words).
 - Titles tend to be slightly shorter as well (≈ 8 – 9 words).
 - Teams are smaller (≈ 2 – 3 authors per paper). The writing style is more compact and theoretically oriented, with fewer co-authors.
- **Astrophysics and cosmology (astro-ph.GA, astro-ph.SR, astro-ph.HE, astro-ph.CO)**

- Some of the longest abstracts in the dataset (\approx 194–220 words).
 - Very large collaborations, especially in high-energy astrophysics:
 - *Astrophysics of Galaxies (astro-ph.GA)*: \approx 9 authors on average.
 - *High Energy Astrophysical Phenomena (astro-ph.HE)*: \approx 14 authors on average.
- These categories show the typical pattern of big observational projects and survey papers.
- **Pure mathematics (math.AG, math.AP, math.CO, math.PR)**
- Shorter abstracts on average (\approx 87–112 words).
 - Short to medium titles (\approx 8–11 words).
 - Small teams (\approx 2 authors, often single-author papers).
- This reflects the more individual and concise style of mathematical research articles.

Overall, the subject-level profiles confirm that different scientific communities have **distinct textual and collaboration patterns**, which reinforces the feasibility of using titles and abstracts as inputs to classify papers into primary_subject categories.

2.6.16 Mean Abstract Length by Primary Subject

The chart presents the **average number of words in abstracts** across the Top 20 most frequent primary_subject categories. Several important patterns emerge:

1) Clear subject-driven variation in abstract size

Research areas differ substantially in how much textual context is typically needed to describe contributions. Some domains systematically produce **longer abstracts**, notably:

Table 2.3 - Mean Abstract Length by Primary Subject (Long Abstracts)

Subject Area (Examples)	Approx. Mean Words	Interpretation
Astrophysics of Galaxies (astro-ph.GA)	\approx 220	Complex physical systems often require extensive contextualization.
Solar & Stellar Astrophysics (astro-ph.SR)	\approx 205	Descriptive and observational content tends to require longer explanations.
High Energy Astrophysical Phenomena (astro-ph.HE)	\approx 195	Phenomena involve multi-scale theories and many experimental constraints.
Cosmology & Nongalactic Astrophysics (astro-ph.CO)	\approx 195	Typically strong theoretical foundations + empirical discussion.

2) More concise abstracts in mathematical and theoretical fields

Domains with more **formal/derivational contributions** tend to exhibit shorter abstracts:

Table 2.4 - Mean Abstract Length by Primary Subject (Short Abstracts)

Subject Area	Approx. Mean Words
Algebraic Geometry (math.AG)	\approx 87
Analysis of PDEs (math.AP)	\approx 102
Probability (math.PR)	\approx 113
Combinatorics (math.CO)	\approx 111

This aligns with writing norms where contributions are often stated more formally and compactly.

3) Computer Science & Machine Learning in the middle range

Table 2.5 - Mean Abstract Length by Primary Subject (Computer Science & Machine Learning)

Subject	Mean Words	Implication
Computer Vision (cs.CV)	~178	Heavily experimental, often describing pipeline + datasets + metrics.
Machine Learning (cs.LG)	~172	Increasingly standardized abstract structure (problem → method → results).
Computation & Language (cs.CL)	~158	NLP abstracts remain relatively dense due to task/method/result summaries.

4) Implications for downstream classification

These differences matter for modeling:

- Domains with long abstracts give **richer semantic signals** → beneficial for transformer-based classifiers.
- Shorter, theory-driven fields may require **stronger title features** or **citation/metadata augmentation**.
- Subject-aware text normalization (e.g., truncation vs. summarization) should consider this length variance.

Key Takeaway: Abstract length is **not random noise** - it is a **domain-dependent linguistic property** that can support feature engineering and model design for scientific document classification.

2.7 Inspect Text Artifacts in Abstracts

2.7.1 Text Artefacts in Abstracts

A targeted scan of the abstract field shows that a non-negligible fraction of records contains markup or mathematical notation:

- **HTML tags** (e.g.
,): present in **≈ 8.7%** of abstracts.
- **HTML entities** (e.g. &, <, '): present in **≈ 7.1%** of abstracts.
- **Inline LaTeX math using dollar signs** (\$...\$): present in **≈ 28.8%** of abstracts.
- **Alternative LaTeX math syntaxes** (\(...\)) or \begin{equation}...\end{equation}): rare but present in a small minority (**≈ 0.04%**).

Qualitative examples confirm that: - HTML tags and entities mostly appear in **legacy or scraped metadata** (e.g. line breaks, links to the original arXiv page, encoded apostrophes). - LaTeX expressions encode **mathematical formulas and symbols** which are important scientifically but often add noise for high-level **subject classification** and can confuse tokenization.

For the downstream classification task, this motivates a preprocessing strategy that: 1. **Removes HTML tags** and **decodes HTML entities** into normal characters; 2. **Replaces LaTeX math fragments** by a neutral placeholder token (e.g. [MATH]), preserving

the information that “some math is here” without keeping the raw syntax;
3. Normalizes spacing to produce clean, model-friendly input text.

2.8 Impact of Cleaned Text Fields for Title and Abstract

2.8.1 Remarks on Original vs. Cleaned Text Fields

A comparison between the original and cleaned versions of *title* and *abstract* shows that the cleaning process successfully removes markup noise such as HTML entities (&) and normalizes text without altering its scientific meaning. Although this normalization step belongs formally to the **Data Preparation** stage, its inspection at this point is relevant to Data Understanding because it highlights important characteristics of the corpus:

- A non-negligible portion of abstracts contain HTML artifacts and LaTeX fragments.
- These elements do not contribute semantic value for classification tasks and may negatively affect tokenization and language modeling.
- Cleaning ensures that downstream NLP models (e.g., TF-IDF, SciBERT, Longformer) receive more consistent and less sparse representations.

Therefore, while no transformation is applied globally at this stage, the preliminary review confirms that text normalization will likely improve vocabulary coherence, reduce feature sparsity, and contribute to a more stable modeling phase.

2.9 Linguistic Patterns in Abstracts

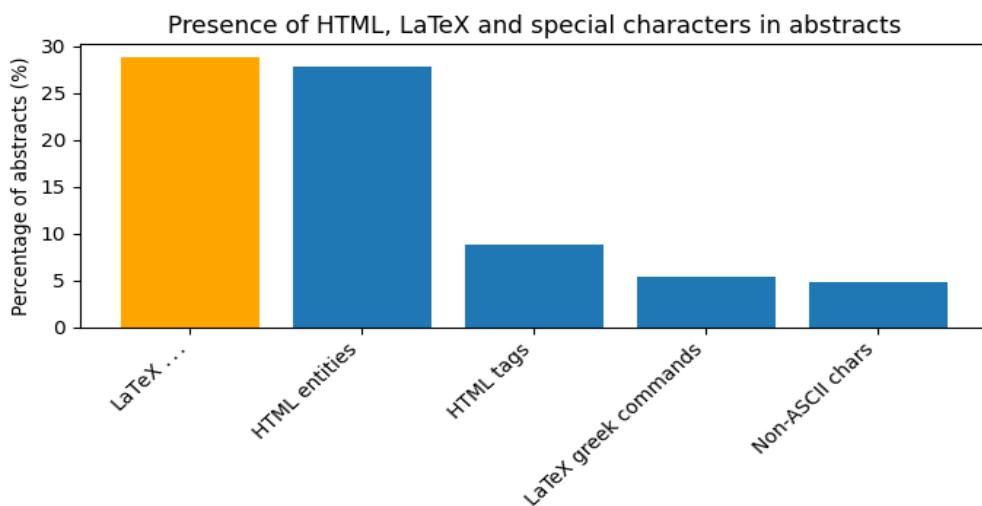


Figure 2.12 - Presence of HTML, LaTeX and Special Characters in Abstracts

2.9.1 Linguistic Artefacts in Abstracts

The inspection of linguistic patterns in the abstract field shows that non-trivial markup and mathematical notation are very common:

- **LaTeX \$...\$ math syntax** appears in about **28.8%** of all abstracts.
- **HTML entities** (e.g. &, ') appear in about **27.8%** of abstracts.
- **HTML tags** (e.g.
,) occur in about **8.7%** of abstracts.
- **LaTeX Greek commands** such as \alpha, \beta, \lambda appear in roughly **5.4%** of cases.

- **Non-ASCII characters** (accents, special symbols) are present in around **4.9%** of abstracts.

The bar chart highlights **LaTeX math** as the most frequent artefact, followed closely by HTML entities. Example abstracts confirm that these elements encode either:

- **Scientific notation** (mathematical formulas in LaTeX).
- **Formatting noise** introduced during scraping (HTML tags and encoded punctuation).

From a data understanding perspective, this indicates that the corpus is:

- **Mathematically dense**, especially in physics and mathematics papers.
- **Technically noisy** from an NLP standpoint, due to HTML/encoding artefacts.

For the later modeling phase, this motivates:

- Dedicated text-cleaning (removal/normalization of HTML and entities)
- Possibly special handling or masking of LaTeX to avoid fragmenting the vocabulary and harming tokenization.

2.10 Main Scientific Domains in primary_subject

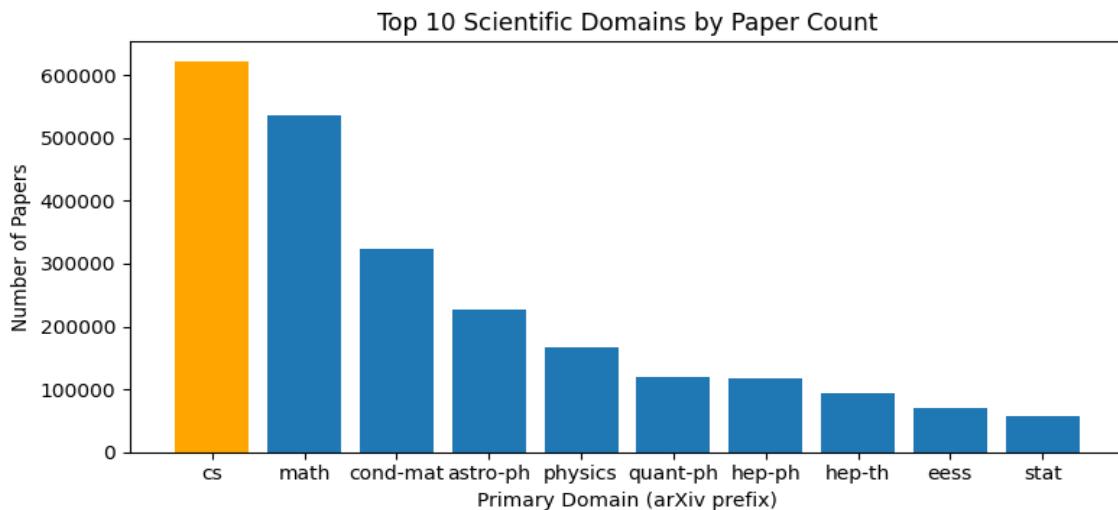


Figure 2.13 - Top 10 Scientific Domains by Paper Count

2.10.1 Distribution of Main Scientific Domains

By extracting the arXiv prefix from primary_subject (e.g. cs.CV → cs, astro-ph.GA → astro-ph) we obtain a high-level view of the scientific domains represented in the dataset.

The distribution is clearly dominated by a few large areas:

- **Computer Science (cs)**: 622,419 papers (**24.4%** of the corpus)
- **Mathematics (math)**: 535,240 papers (**21.0%**)
- **Condensed Matter Physics (cond-mat)**: 322,843 papers (**12.7%**)
- **Astrophysics (astro-ph)**: 227,757 papers (**8.9%**)
- **General Physics (physics)**: 166,441 papers (**6.5%**)

Smaller but still substantial contributions come from **Quantum Physics (quant-ph)**, **High Energy Physics (hep-ph, hep-th)**, **Electrical Engineering and Systems Science (eess)**, and **Statistics (stat)**, among others.

The bar chart for the Top 10 domains highlights cs as the largest block, followed closely by math. This confirms that the dataset is **heavily skewed towards computer science and mathematics**, with strong representation of physics-related fields. For downstream modeling, this implies that:

- Any classifier trained on the full corpus will naturally see many more examples from cs and math than from smaller domains such as econ or q-fin.
- It is important to keep this domain imbalance in mind when designing both **label sets** and **evaluation strategies**, especially if the goal is to build models that generalize beyond the dominant areas.

2.11 Top Unigrams and Bigrams in Abstracts

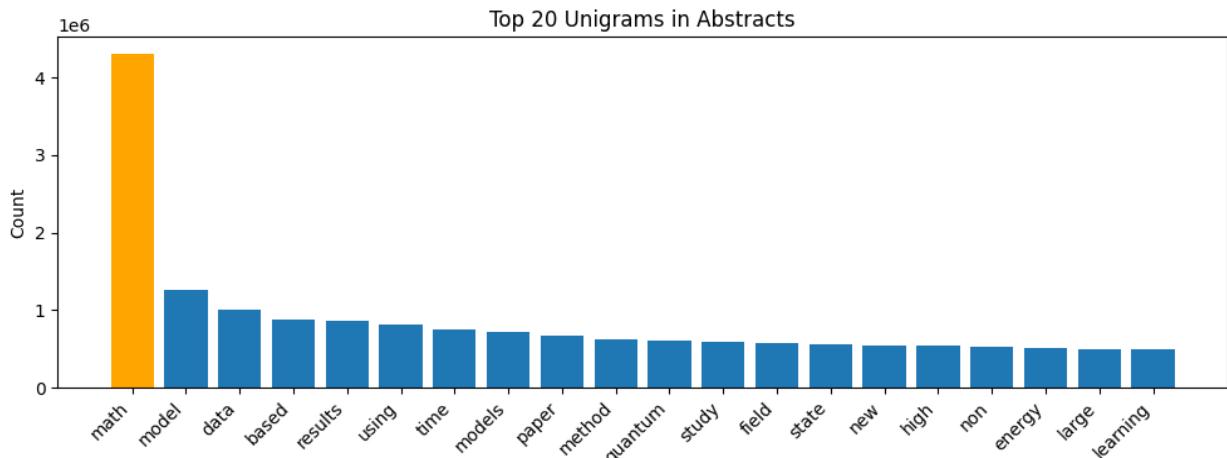


Figure 2.14 - Top 20 Unigrams in Abstracts

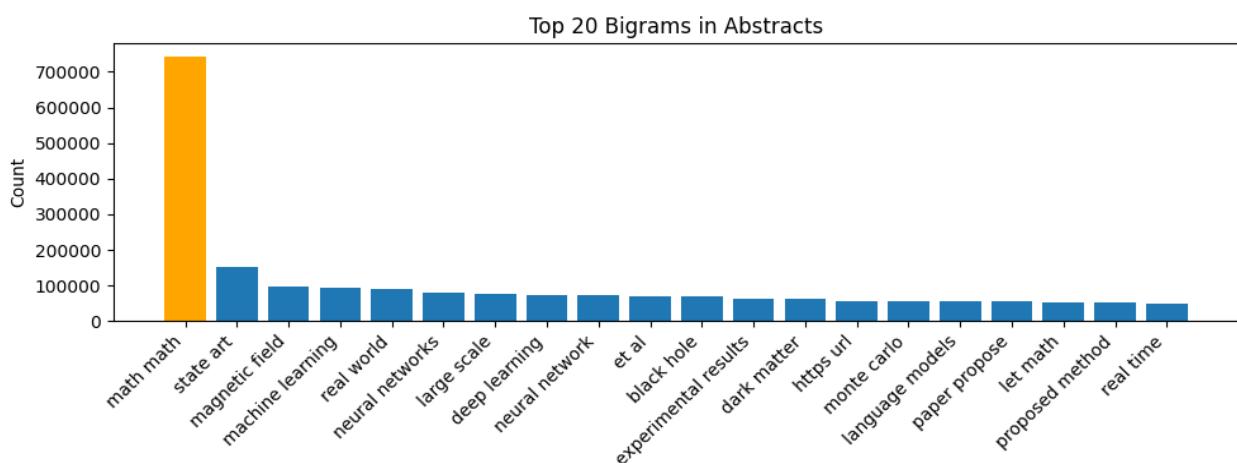


Figure 2.15 - Top 20 Bigrams in Abstracts

2.11.1 Dominant Vocabulary in Abstracts

Using a bag-of-words Count Vectorizer (English stop words removed, $\text{min_df}=5$), the most frequent single words in the cleaned abstracts are:

- **math** ($\approx 4.3\text{M}$ occurrences) – by far the most frequent token in the corpus.

- **model, models, methods, data, results, using, study, field, theory, performance, etc.**

This list reflects a mixture of:

- **Generic scientific terminology:** *model, data, results, using, study, field, method(s), systems, problem, present.*
- **Domain-specific keywords** that are common across several areas: *quantum, energy, space.*
- **“Meta-words” like math**, which appear extremely often because arXiv injects the word *math* into some metadata/structural context when scraping TeX sources.

From a modeling perspective, these findings confirm that:

- The corpus exhibits a strong **scientific register**, dominated by words that describe methods, models and results rather than informal language.
- High-frequency terms such as *math* or very generic methodological words may carry **limited discriminative power** for subject classification and might be down-weighted (e.g., via TF-IDF) or even removed if necessary.
- More specific terms and n-grams (e.g. bigrams like *machine learning, neural network, quantum field*) are likely to provide richer signal for distinguishing between domains such as cs.LG, cs.CV, quant-ph, hep-th, etc.

2.12 Heuristic Detection of Truncated Abstracts

2.12.1 Potentially Truncated Abstracts

To assess the textual quality of the corpus, we applied a simple heuristic to detect potentially truncated or incomplete abstracts. An abstract was flagged as “suspicious” if it either:

- Explicitly ended with an ellipsis "..."
- It was relatively long (more than 600 characters) but did **not** end with a final punctuation mark (., ?, !).

The results are:

- Total abstracts: **2,549,619**
- Suspicious abstracts (union of both conditions): **36,828 ($\approx 1.44\%$)**
- Ending with "...": **1,016 ($\approx 0.04\%$)**
- Long and without final punctuation: **35,812 ($\approx 1.40\%$)**

Manual inspection of sample cases shows that many of these flagged abstracts look like they were **cut mid-sentence**, often around the point where LaTeX, HTML or special characters appear (e.g., line breaks, quotes, references to instruments or telescopes). In such cases the scientific content is partially preserved but the narrative is incomplete.

From a data understanding perspective, this suggests that:

- The **vast majority** of abstracts are intact; potential truncation affects only a **small minority ($\sim 1-1.5\%$)** of the corpus.
- Truncated abstracts may still carry enough information for high-level subject classification, but they can introduce some noise and slightly distort length statistics.

- During the later **Data Preparation** phase, we may choose either:
 - Keep these records (accepting a small amount of noise).
 - Apply additional filtering / quality flags if the modeling task requires very clean text.

Overall, the dataset appears to be of **high-quality and largely complete**, with only a limited fraction of abstracts showing signs of truncation.

2.13 Unigram Word Cloud From Cleaned Abstracts

The unigram word cloud provides a visual summary of the most frequent individual words in the abstracts. The largest tokens are *MATH*, *model*, *show*, *paper*, *using*, *results*, *system*, *method*, *propose*, *study*, among others. This picture is consistent with the quantitative unigram counts:

- It confirms a strong **methodological and result-oriented vocabulary** (e.g., *model, method, results, algorithm, show, demonstrate*).
 - The extremely prominent *MATH* token reflects arXiv’s TeX-based origin and the strong mathematical orientation of the corpus.
 - Content words such as *system, solution, problem, data, structure, network* highlight the prevalence of computational and physical sciences.

Overall, the unigram cloud reinforces that the dataset is dominated by **formal scientific language**, dominated by generic research verbs (“show”, “propose”, “present”) and methodology-related nouns (“model”, “method”, “results”).

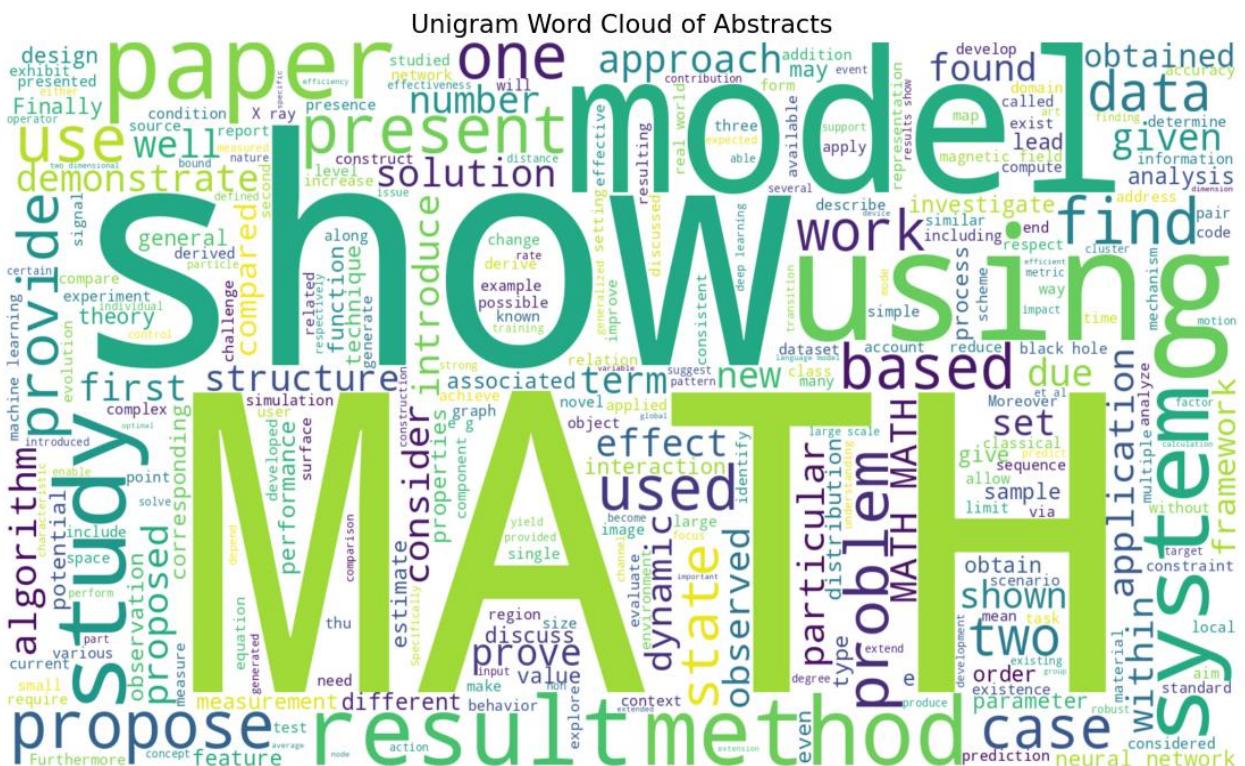


Figure 2.16 - Unigram Word Cloud of Abstracts

2.14 Bigram Word Cloud from Cleaned Abstracts

The bigram word cloud provides a more semantic view of the corpus by focusing on frequent word pairs instead of isolated tokens. Several patterns stand out:

- **Machine learning & AI:** phrases such as *machine learning*, *neural networks*, *deep learning*, *large language*, *language models*, *reinforcement learning*, *time series* highlight the strong presence of modern AI research.
 - **Astrophysics & cosmology:** bigrams like *black holes*, *dark matter*, *star formation* capture the astrophysical component of the corpus.
 - **Condensed matter & physics:** expressions such as *magnetic field*, *phase transition*, *ground state*, *standard model*, *monte carlo*, *cross section* are characteristic of condensed matter and high-energy physics papers.
 - **Generic scientific phrasing:** bigrams like *experimental results*, *proposed method*, *state art*, *real world*, *large scale* reflect standard ways of framing contributions in scientific writing.
 - **Artefacts:** tokens such as *math math*, *https url*, *et al* appear due to TeX/metadata conventions and scraping artefacts, reinforcing the need for light cleaning or down-weighting of such pairs during modeling.

Compared to the unigram cloud, the bigram cloud surfaces more domain-specific concepts (e.g., *black hole* vs. *black / hole* separately), confirming that n-gram features capture richer semantic information that can be useful for downstream subject classification and for interpreting what an NLP model might learn from the abstracts.

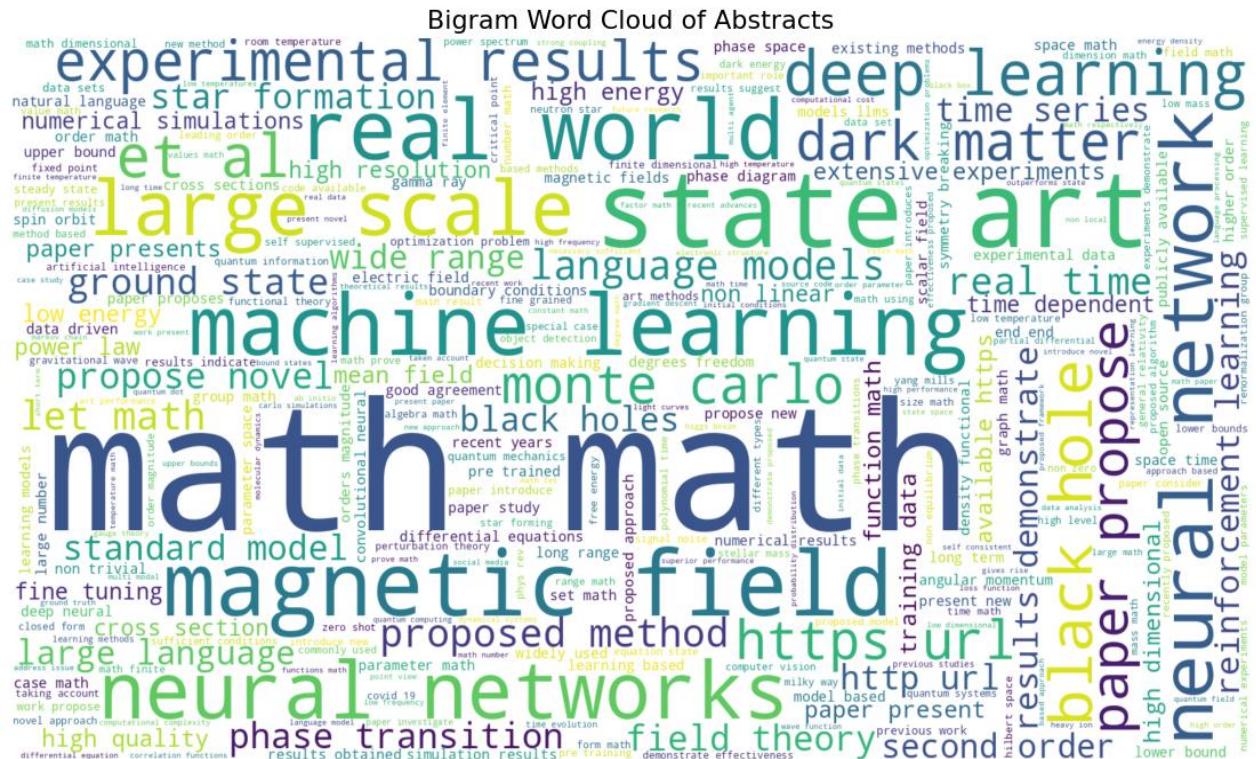


Figure 2.17 - Bigram Word Cloud of Abstracts

2.15 Data Understanding Summary

The exploration analysis of ~2.55M arXiv papers revealed clear structural and linguistic patterns that will directly influence the modeling stage:

Document structure

- Most titles contain **8–12 words**, and most abstracts fall in the **100–200 words** range, aligning with academic writing norms.
- A small but relevant portion of abstracts (~1.4%) appears **truncated or incomplete**, indicating potential data quality issues that may require filtering or repair.

Presence of LaTeX, HTML, and special characters

- A significant fraction of abstracts contains scientific markup:
 - **LaTeX inline math** (\$...\$): **~28.8%**
 - **HTML entities** (e.g., &): **~27.8%**
 - **HTML tags**: **~8.7%**
 - **LaTeX Greek** and other commands: **~5%**
- This confirms the scientific nature of the corpus and suggests benefits from dedicated cleaning steps for downstream NLP tasks.

Scientific domains

- The dataset spans all major arXiv research areas, with the largest domains being:
 - **cs (24.4%)**
 - **math (21.0%)**
 - **cond-mat (12.7%)**
 - **astro-ph (8.9%)**
- This distribution is imbalanced and may require class grouping or stratified sampling depending on the modeling objective.

Vocabulary and linguistic patterns

- Frequent unigrams and bigrams highlight dominant themes across scientific fields:
 - Core concepts: *model, data, results, method, using*
 - AI/ML signals: *machine learning, neural networks, deep learning, language models*
 - Physics/astro signals: *black hole(s), dark matter, magnetic field, monte carlo*
- Word clouds confirmed strong semantic clustering consistent with major arXiv communities.

Key implications for Data Preparation

Based on these findings, the following actions are recommended in the next stage:

Table 2.6 - Implications of Structural and Linguistic Patterns

Requirement / Need	Reason
Remove/normalize LaTeX, HTML, Greek symbols, and entities	Improve tokenization and semantic modeling
Handle truncated/low-quality abstracts	Reduce noise and potential misleading training examples

Consider domain balancing or class aggregation	Mitigate classification bias toward dominant areas
Decide between unigram vs n-gram or contextual embeddings	Vocabulary indicates value in capturing multi-word expressions

This understanding provides a solid foundation before proceeding to data cleaning, feature engineering, and modeling.

3. Data Preparation

3.1 Context

Unlike traditional methods (like TF-IDF or Word2Vec) which require aggressive cleaning to reduce vocabulary size, encoder architectures generally perform best with minimal intervention on the raw text.

- **Stemming and Lemmatization:** Encoder models often use Byte-Pair Encoding (BPE) for tokenization, which breaks words into sub-word units (e.g., ‘running’ might become ‘run’ and ‘##ing’). This sub-word tokenization implicitly handles morphological variations, automatically grouping inflections of the same root word. Forcing stemming or lemmatization before tokenization can inadvertently remove necessary linguistic information, often leading to reduced performance.
- **Stop Word Management:** It is generally not necessary to remove high-frequency words (stop words). Transformer models, which are built on the attention principle, automatically learn to focus only on the words that impact the classification output, effectively down weighting common words. Removing stop words can be problematic in cases like sentiment analysis where the absence of words like “not” or “too” changes the context.
- **Case Handling:** Case normalization (lowercasing) is typically managed by selecting the appropriate pre-trained model. If you choose an uncased model (e.g., bert-base-uncased), the model will internally convert all input text to lowercase. If you choose a cased model (e.g., bert-base-cased), capitalization is preserved, which can be important for tasks where proper nouns are critical. You do not need to manually perform case folding.

3.2 Data Preprocessing

This code filters the dataset to retain only the primary subjects with a sufficient number of samples. It first counts how many papers belong to each primary_subject, then identifies the subjects that have at least 20,000 examples. Finally, it filters the dataset to include only papers whose primary subject is in this set of frequent labels. This ensures the resulting dataset is focused on well-represented categories, reducing class imbalance and providing enough data for meaningful analysis or model training.

This code processes a dataset of abstracts by tokenizing each text using the Longformer tokenizer, collecting the lengths of the resulting token sequences, and analyzing their distribution. It computes key statistics such as the median, average, minimum, and maximum token lengths, while also visualizing the distribution with a histogram to identify patterns like long or short documents. This helps ensure the dataset is well understood and

suitable for model training, highlighting sequences that may be too short to provide meaningful context or too long to handle efficiently.

The code filters the dataset to keep only tokenized texts whose lengths are between MIN_LEN and MAX_LEN tokens. This ensures that very short texts (too little context) and very long texts (memory-heavy) are removed, making the dataset more consistent and efficient for model training.

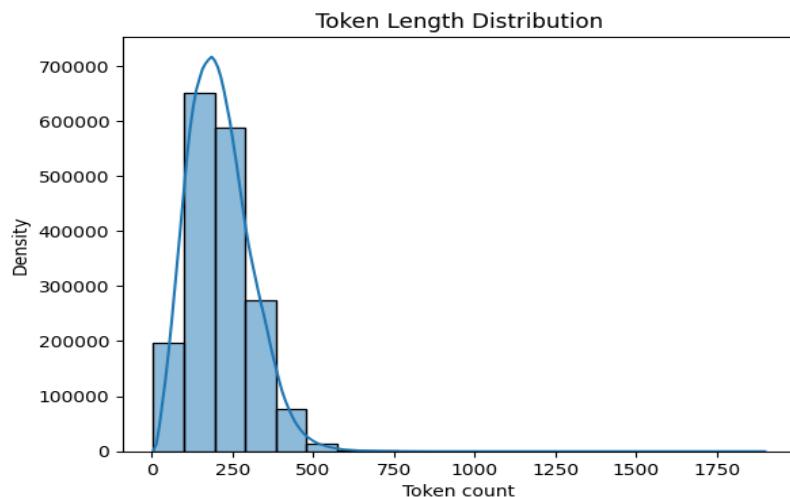


Figure 3.1 - Token Length Distribution

4. Modeling

4.1 Introduction

The development of **SCIBERT**, a pretrained language model based on BERT, was primarily motivated by the challenges of obtaining high-quality, large-scale annotated data for NLP tasks in scientific domains. Since prior language models like BERT and ELMO were trained on general domain corpora such as Wikipedia and news articles, they were less effective at capturing the unique vocabulary and structure of scientific text. SCIBERT was created to address this gap by leveraging unsupervised pretraining on a large multi-domain corpus of scientific publications, aiming to transfer knowledge embedded in the literature to improve performance on a suite of downstream scientific NLP tasks.

SCIBERT is highly suitable for fine-tuning a model to classify papers, including those from Arxiv, based on their abstracts. The model was trained on a corpus of 1.14 million papers, with a substantial portion (18%) originating from the **computer science domain**, which is a primary focus of the Arxiv repository.

The model's superior performance was demonstrated across several relevant tasks, including Named Entity Recognition (NER) from computer science abstracts (SciERC) and various text classification tasks. Specifically, SCIBERT achieved new State-of-the-Art (SOTA) results on classification tasks like ACL-ARC and SciCite, which involve analyzing citation intent and text in scientific publications.

These results indicate that the model's domain-specific pretraining successfully captures the necessary linguistic features in scientific discourse for high-accuracy classification.

SCIBERT is fundamentally an application of the **Transformer** architecture, placing it directly within the family of **Large Language Models (LLMs)**. The model uses the exact same core multilayer bidirectional **Transformer** architecture as its predecessor, BERT. Therefore, SCIBERT can be characterized as a **domain-adapted LLM**, maintaining the structural configuration and pretraining objectives (masked token prediction and next sentence prediction) of the original BERT-Base model.

Its key innovation as an LLM is its specialization: retraining the architecture on a massive scientific corpus and using an in-domain vocabulary, **SCI VOCAB**, which proved essential for its improved performance across scientific tasks.

4.2 SciBERT's Architecture

Transformer-Based:

- Built on BERT (Bidirectional Encoder Representations from Transformers)
- Uses the transformer architecture from the [Attention is All You Need](#) paper

It has key components:

- Multi-head self-attention mechanisms
- Position embeddings
- Layer normalization
- Feed-forward networks

Model Size:

- 110M parameters (BERT-base size)
- 12 transformer layers
- 768 hidden dimensions
- 12 attention heads

5. Evaluation

5.1 Accuracy and Loss

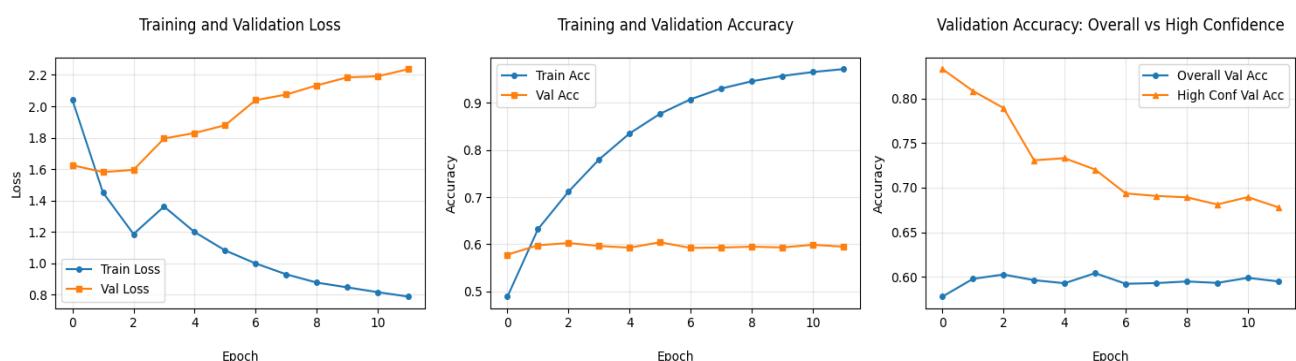


Figure 5.1 – Accuracy and Loss

5.2 Confidence Analysis

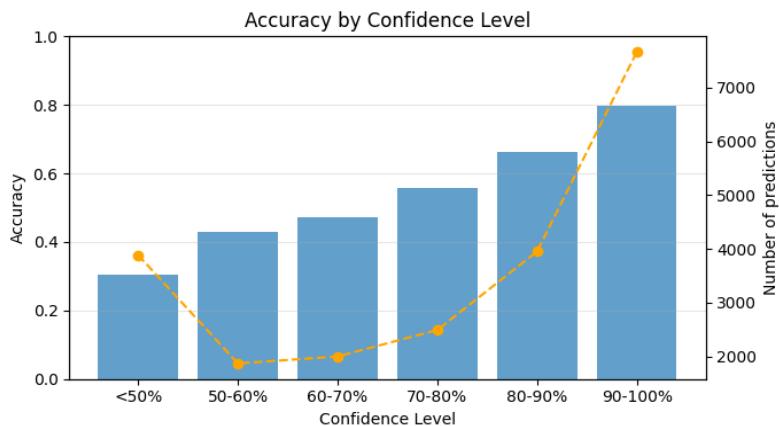


Figure 5.2 – Confidence Analysis

5.3 Confusion Matrices

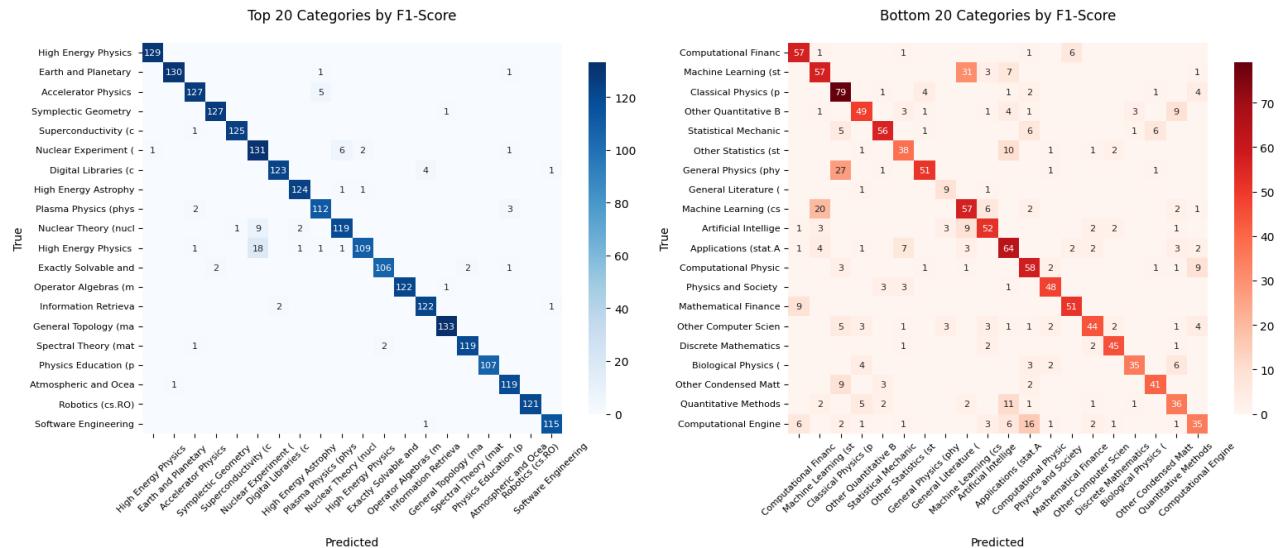


Figure 5.3 – Confusion Matrices

6. Bibliography and Supporting Sources

- [Longformer Base 4096](#)
 - [SciBERT](#)
 - [Arxiv dataset](#)
 - [Direct link to Arxiv dataset metadata file \(contains all of Arxiv abstracts\)](#)