A Transfer Learning Pipeline for Educational Resource Discovery with Application in Survey Generation

Irene Li[1*], **Thomas George**[2], **Alex Fabbri**[1], **Tammy Liao**[1], **Benjamin Chen**[1], **Rina Kawamura**[1], **Richard Zhou**[1], **Vanessa Yan**[1], **Swapnil Hingmire**[3] **and Dragomir Radev**[1]
[1]Yale University, [2]University of Waterloo,
[3]Tata Consultancy Services Limited

Abstract

Effective human learning depends on a wide selection of educational materials that align with the learner's current understanding of the topic. While the Internet has revolutionized human learning or education, a substantial resource accessibility barrier still exists. Namely, the excess of online information can make it challenging to navigate and discover high-quality learning materials in a given subject area. In this paper, we propose an automatic pipeline for building an educational resource discovery system for new domains. The pipeline consists of three main steps: resource searching, feature extraction, and resource classification. We first collect frequent queries from a set of seed documents, and search the web with these queries to obtain candidate resources such as lecture slides and introductory blog posts. Then, we process these resources for BERT-based features and meta-features. Next, we train a tree-based classifier to decide whether they are suitable learning materials. The pipeline achieves F1 scores of 0.94 and 0.82 when evaluated on two similar but novel domains. Finally, we demonstrate how this pipeline can benefit two applications: prerequisite chain learning and leading paragraph generation for surveys. We also release a corpus of 39,728 manually labeled web resources and 659 queries from NLP, Computer Vision (CV), and Statistics (STATS).

1 Introduction

People rely on the internet for various educational activities, such as watching lectures, reading textbooks, articles, and encyclopedia pages. One may wish to develop their knowledge in a familiar subject area or to learn something entirely new. Many online tools exist that enable and promote independent learning (Montalvo et al., 2018; Romero and Ventura, 2017; Fabbri et al., 2018a; Li et al., 2019). A subset of these platforms provide primary literature resources (e.g. publications), such as Google Scholar[1] and Semantic Scholar[2]. As an alternative to these advanced materials, other educational platforms such as MOOC.org [3] deliver free online courses. Also, unstructured searching on the internet is a popular method to discover other useful resources, such as blog posts, GitHub projects, tutorials, lecture slides and textbooks. Rather than diving into the technical details, these secondary literature resources provide a broad overview of the given domain, which is more valuable for beginners. Still, sifting through this material can be challenging and time-consuming, even if the learner is simply looking for a general and reliable introduction into a new subject area.

Publicly accessible data repositories that focus on gathering a fixed number of educational resources exist currently, such as scientific papers (Tang et al., 2008, 2010), online platforms like AMiner (Sinha et al., 2015) and Semantic Scholar. Some archives also compile secondary literature materials. TutorialBank (Fabbri et al., 2018a) is a manually-collected corpus with over 6,300 NLP resources, as well as related fields in Artificial Intelligence (AI), Machine Learning (ML) and so on. LectureBank (Li et al., 2020) is also a manually-collected corpus and contains 1,717 lecture slides. MOOCCube (Yu et al., 2020) is a large-scale data repository containing 700 MOOC (Massive Open Online Courses), 100k concepts and 8 million student behaviours with an external resource. However, in their initial synthesis, these existing corpora either heavily relied on manual efforts that restricted in certain domains, or on a large volume of existing courses sourced from a certain platform. Such solutions are not practically extensible into new or evolving domains. Moreover, according to (Fabbri et al., 2018a), some web data such as blog posts, tutorials and educational web pages are

*Corresponding author: irene.li@aya.yale.edu

[1]https://scholar.google.com/
[2]https://www.semanticscholar.org/
[3]https://www.mooc.org/

Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023), pages 29–43

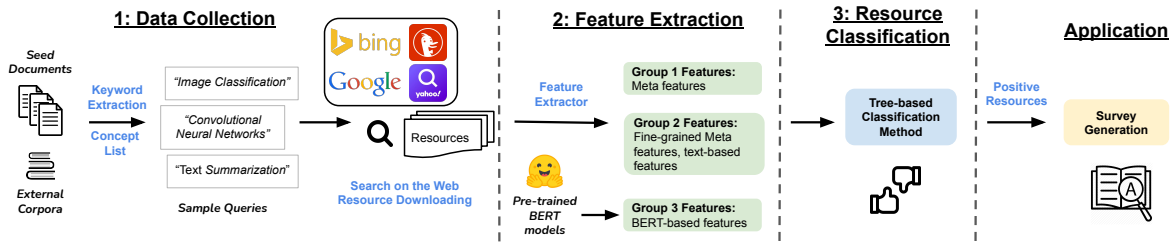

Figure 1: Pipeline Overview. The pipeline contains three steps: query generation, feature extraction, and classification & evaluation. We also show an application in this figure.

also suitable materials for learners. These rich web data are ignored by existing educational platforms such as google scholar and MOOCcube. In this paper, we wish to ease the need for human annotators by proposing a pipeline that automates resource discovery to similar unseen domains through transfer learning. Besides, such a pipeline deals with multiple resource types to take advantage of web data.

Our contributions can be summarized into three parts. First, we present a self-sustaining pipeline for educational resource discovery in close unseen subject area or domain. We apply transfer learning with a novel pre-training information retrieval (IR) model, achieving competitive performances. We show that this pipeline achieves 0.94 and 0.82 F1 scores for two arbitrary target domains on discovering high-quality resources. Second, we demonstrate an application that leverage resources discovered by our pipeline, survey generation for leading paragraph. Lastly, we release the core source code of the pipeline, as well as the training and testing datasets, comprised of 39,728 manually labelled web resources and 659 search queries. [4]

2 Educational Resource Discovery Pipeline

We propose the Educational Resource Discovery (ERD) pipeline that aims at automatically recognizing high-quality educational resources. We model this problem as a resource classification task. Given a resource r, where r can be any source type such as web page, PDF, we can obtain a list of features by feature engineering; based on these features, r is classified positive if it is a high-quality resource, otherwise negative. We illustrate the ERD pipeline in Figure 1. It consists of data collection, feature extraction and resource classification.

[4] https://github.com/IreneZihuiLi/Educational-Resource-Discovery

2.1 Data Collection

2.1.1 Queries for search

In this step, we need to conduct a list of meaningful and fine-grain search queries to start. These search queries will then be applied to online search engines for web resources. Queries can be borrowed from external corpora or extracted from existing seed documents (e.g., textbooks). We focus on three domains: NLP (natural language processing), CV (computer vision) and STATS (statistics). For NLP queries, we utilize external topic lists provided by LectureBankCD (Li et al., 2021), in which there are totally 322 NLP-based and 201 CV-based topics from crowdsourcing. For STATS, we extract a list of fine-grained terms from several seed documents, including several textbooks. These terms contain frequent keywords and phrases that are extracted by TextRank (Mihalcea and Tarau, 2004), a statistical method to keyword ranking. In total, we end up with 322, 201 and 137 queries for NLP, CV and STATS domain.

To craft our search engine queries, we leverage advanced search conditions: *filetype* and *site* (website). Specifically, we consider three file types: PDF, PPTX/PPT, and HTML. Moreover, according to the TutorialBank corpus (Fabbri et al., 2018b), resources clustered by the components of their URL possess highly correlated educational content. Thus, we prioritize restricting our queries to websites that consistently provide high-quality resources. We select the top sites from the manually-created TutorialBank corpus and incorporate them into our search queries, as exemplified in 1. We also include the ".edu" top-level domain as a special case for our search queries in order to capture general educational resources. Finally, we combine our query terms with the website and file-type constraints: e.g. "word embeddings filetype:pdf". We also augment the original query by generating a disjunction of its variations: e.g., "stochastic gradient

towardsdatascience.com	datahacker.rs
medium.com	hackernoon.com
www.analyticsvidhya.com	skymind.ai
www.kdnuggets.com	maelfabien.github.io
machinelearningmastery.com	rubikscode.net
paperswithcode.com	research.googleblog.com

Table 1: Top sites found in the TutorialBank corpus (Fabbri et al., 2018b).

descent" becomes "stochastic gradient descent OR SGD". Table 2 displays several sample queries.

Once the queries are generated, we leverage three well-established online search engines: DuckDuckGo (https://duckduckgo.com/), Yahoo (https://search.yahoo.com/) and Bing (https://www.bing.com/) to obtain our candidate resources. The top N URLs (where N is determined from the domain, file type and site type, varying from 20 to 100 to control the total number of resources we want to collect) for a given query are cached after checking their HTTP response status and ensuring that a URL has not already been collected as part of another query. Moving forward, the documents pointed to by all of these URLs were automatically downloaded and parsed for their features. Certain features, such as the number of authors were collected using heuristics that accounted for most of the variability within the diverse dataset. The ERD Pipeline's parsers use the pdfminer[5] and grobid[6] libraries for PDF files, Apache Tika[7] for PPTX/PPT and beautifulsoup[8] for HTML.

2.1.2 Annotation

After collecting all resources, the next step is to assign a binary label to each resource based on its quality. Our annotators consist of 7 graduate and senior college students with a solid background in NLP, CV, and STATS. A resource is annotated as positive if it is a high-quality one. Guidelines for a positive resource are:

- *Informative and relevant*: introducing basic knowledge about a specific topic. For example, tutorials, introductions, explanations, guides.
- *Papers and lecture slides*: papers and lecture notes about a topic in the correct domain.

[5] https://github.com/pdfminer/
[6] https://github.com/kermitt2/grobid
[7] https://tika.apache.org/
[8] https://crummy.com/software/BeautifulSoup/

NLP Sample Queries
" morphological disambiguation " filetype:pptx
" word embeddings " filetype:pdf
" text classification tutorial "
" summarization nlp tutorial" site:edu
CV Sample Queries
" computer graphics " site:kdnuggets.com
" texture classification " filetype:pptx
STATS Sample Queries
" conditional probability " site:kdnuggets.com
" multinomial distribution introduction " filetype:html

Table 2: Sample queries in the three domains.

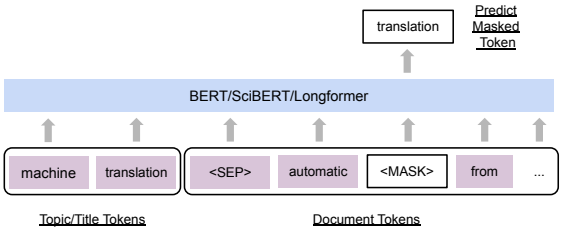

Figure 2: QD-BERT MLM pretraining.

- *Other secondary literature articles*: i.e., blog posts with informative descriptions, definitions and code blocks.

The annotation criteria for a poor resource are:

- *Not informative*: dataset/software/tool download page without introductory descriptions, such as a paper abstract page (not the paper content), a download page with links.
- *Irrelevant*: not showing correct content, broken URLs, URLs with not enough or no text (video or image only).
- *No knowledge included*: such as a course landing page, a person's personal website page.
- *A list of resources/datasets*: containing only links to other pages.

Finally, to measure the inter-coder agreement of the labels, we randomly picked 100 resources and asked each annotator to provide labels independently. Krippendorff's alpha (Krippendorff, 2011) on this sample evaluated to 0.8344, indicating a high degree of consistency amongst all annotators.

We detail statistics about our collected dataset in Table 2, providing the total counts by file type and domain. From the three domains, we collected 39,728 valid resources using 659 distinct queries and achieved a total positive rate of 69.05%.

	NLP	CV	STATS	**Total**
Query Num	322	200	137	659
PPTX	1,216	733	1,463	3,412
PDF	4,961	3,782	1,449	10,192
HTML	9,368	9,302	7,454	26,124
Total	15,545	13,817	10,366	**39,728**
Pos.Num	9,589	11,101	6,742	27,432
Pos.Rate	0.6169	0.8034	0.6501	**0.6905**

Table 3: Dataset statistics by domain and file type. *Pos.Num* is the number of positive resources. *Pos.Rate* is the fraction of resources that were labeled as positive.

2.2 Feature Extraction

To train a classifier to identify high-quality educational resources, we first focus on feature engineering. Specifically, we investigate the following three groups of classification features and summarize them in Table 4.

Group 1 Features Some of the meta-features of a document that can characterize its quality are embedded in its structure. The features encompassed by Group 1 are high-level and coarse-grained, and focus on aspects such as: the number of headings, equations, outgoing links and authors in a given resource. Heuristically, some good tutorials may tend to include more equations and paragraphs, with many details included. We list all 8 such features in Table 4, Group 1.

Group 2 Features These meta-features describe the fine-grained but statistical details of the document. The resource URL's components, such as the top-level domain name and subdomain name, correlate resources from websites that deliver consistent quality. The other Group 2 features are centered around the characteristics of the free text. For instance, *NormalizedUniqueVocab* (the size of the vocabulary divided by the total number of words) can estimate the vocabulary's complexity and *PercentTypos* (the percentage of words that are incorrectly spelled) can approximate reliability. We itemize such features in Table 4, Group 2.

Group 3 Features In addition to the above features, we propose 9 features based on pretrained language models. To achieve this, we first choose three models[9]: BERT (Devlin et al., 2019), SciBERT (Beltagy et al., 2019) and Longformer (Beltagy et al., 2020). BERT is a pretrained language model that was pretrained on Wikipedia documents. SciBERT is a BERT-based model trained on the scientific domain, making it suitable for our use case. Longformer is a BERT-based model that handles longer input sequences.

Feature Name	**Explanation**
Group 1	
NumAuthor	Number of authors
NumHeading	Number of headings
NumFig	Number of figures
NumEqu	Number of equations
NumPara	Number of paragraphs
NumSent	Number of sentences
NumLink	Number of outgoing links
BibLen	Bibliography length
Group 2	
Subdomain	Subdomain of resource URL
SecondDomain	Second-level domain of resource URL
TopDomain	Top-level domain of resource URL
NumUrlSubdirs	Number of URL subdirectories
NormalizedUniqueVocab	Number of unique words divided by total number of words
UniqueVocabMean	Mean number of occurrences of a word
UniqueVocabStdev	Stdev of number of occurrences of a word
WordLenMean	Mean number of characters per word
WordLenStdev	Stdev of number of characters per word
SentenceLenMean	Mean number of words per sentence
SentenceLenStdev	Stdev of number of words per sentence
PercentTypos	Percentage of words that were misspelled
NumGithubLinks	Number of links to GitHub
Group 3	
bert	BERT base model
scibert	SciBERT base model
longformer	Longformer base model
arXiv_bert	BERT pre-trained on arXiv
arXiv_scibert	SciBERT pre-trained on arXiv
arXiv_longformer	Longformer pre-trained on arXiv
TB_longformer	BERT pre-trained on TutorialBank
TB_bert	SciBERT pre-trained on TutorialBank
TB_scibert	Longformer pre-trained on TutorialBank

Table 4: Chosen features: we select 3 groups consist of meta features and deep learning-based features.

Moreover, we introduce a novel pre-training approach: QD-BERT MLM (Query-document BERT Masked Language Modeling). A query could be a single word, phrase or a paper title, indicating the **topic** or **main idea** of the document. We pair the query term with the corresponding document as the input and follow the Masked Language Modeling (MLM) method of BERT (randomly masking 15% tokens and letting the model predict them), as shown in Figure 2. We apply two external corpora for pre-training to ensure the data quality: TutorialBank (TB) [10] and arXiv [11]. The latest TutorialBank has 15,584 topic-document pairs; and arXiv has 259,050 title-abstract pairs (computer science papers only). We enumerate all models in Table 4, Group 3, naming *dataset_modelname*.

We propose an information retrieval-based scoring function to combine features from deep models with Group 1 and 2 features. This scoring function

[9] https://huggingface.co/transformers/pretrained_models.html

[10] http://aan.how/download/

[11] https://www.kaggle.com/Cornell-University/arxiv

Features	NLP→CV			NLP→STATS		
	F1	Precision	Recall	F1	Precision	Recall
Group 1	0.7238	0.5802	0.9617	0.6508	0.5405	0.8177
Group 1 + 2	0.8579	0.7772	0.9571	0.7990	0.8141	0.7845
Group 3, BERT Only*	0.7764	0.7522	0.8497	0.7923	0.7903	0.7944
Group 1 + 2 + 3	**0.9402**	0.9849	0.8994	**0.8225**	0.9965	0.7002

Table 5: Classification Results in two target domains: CV and STATS. For Group 3, BERT Only*, we report the best model: CV (`scibert`), STATS (`TB_scibert`).

calculates a score of each resource, showing the relevancy of the resource to all the searching queries. Relevancy is one of the most indicators that the resource is annotated as positive. The score is higher if it is more relevant to the queries. In Section 2.1.1, we apply a list of queries ($q \in Q$) to download resources, we compute a cosine-similarity based ranking score $score_r$ for resource r:

$$score_r = \sum_{q \in Q} cosine\,(V_q, V_r)$$

where V_q and V_r are BERT-based model embeddings for the query term and resource respectively. We compute scores on each pre-trained BERT models of each resource.

2.3 Resource Classification

Since there are various feature types, we conduct prepossessing before applying the classifiers. Numerical values are binned into groups, and categorical features are converted into integer codes. We evaluate four traditional classifiers: Random Forest (RF), Decision Tree (DT), Support Vector Machine (SVM) and Logistic Regression (LR). We find that RF performs the best and has a slight edge over DT, but SVM and LR significantly lag behind. Thus, we report the Random Forest's performance, summarized in Table 5. Specifically, we include precision, recall and F1 scores on different feature groups: Group 1, Group 1+2, and Group 1+2+3. The last setting achieves the best performance. Additionally, since it is also possible to solely apply BERT models (Group 3) for the classification task, we include a special setting: Group 3, BERT only. While BERT's results in isolation are good, Group 1+2+3 still remains the winner.

In general, performance on the CV domain is better than on STATS. This is expected given that the corpus distance between NLP and CV is smaller than the one between NLP and STATS. We give detailed data analysis in the next section.

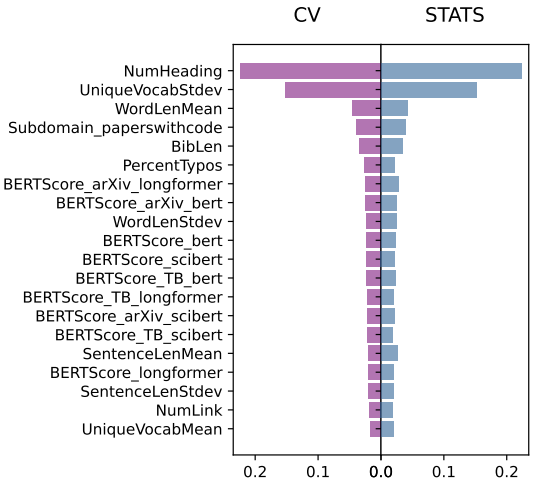

Figure 3: Top 20 features on two target domains.

3 Data Analysis

To better understand the collected data and our classifier's performance, we conduct a study on the features and corpus differences between the three experimental domains.

Feature Importance Score We take the best-performed model of NLP→CV domain (Group 1+2+3), and take the Gini Index calculated by Decision Trees as the feature importance score. Overall, we extract 8746 features in CV and 8525 features of STATS after binning numerical values and encoding categorical features. In Figure 3, we list the top 20 features of CV and STATS. Some Group 1+2 features rank in the top 5, since they are main indicators that the resource is informative (i.e., more heading numbers, longer contents). Additionally, Group 3 features (starting with *BERTScore*) also play an important role. In fact, all 9 BERT-based feature scores rank top 20, suggesting that our scoring function that adds these BERT-based semantic features into the pipeline is very helpful when doing classification for resource discovery.

Corpus Differences Our pipeline performs better on CV topics, which can be attributed to cor-

Domain	Top 10 Sites
NLP	www.cs.cmu.edu , web.stanford.edu , www.cs.toronto.edu , www.paperswithcode.com , maelfabien.github.io , www.academia.edu , courses.cs.washington.edu , nlp.stanford.edu, ocw.mit.edu , www.cs.cornell.edu
CV	www.kdnuggets.com, maelfabien.github.io , www.paperswithcode.com , www.academia.edu , www.cs.toronto.edu , www.cs.cmu.edu , web.stanford.edu , courses.cs.washington.edu, cseweb.ucsd.edu, www.cs.cornell.edu
STATS	www.kdnuggets.com, maelfabien.github.io , www.paperswithcode.com , web.stanford.edu , ocw.mit.edu , online.stat.psu.edu, www.hackernoon.com, www.sjsu.edu, research.googleblog.com,www.cpp.edu

Table 6: Comparison of the top 10 sites. **Gray** means overlapped in both CV and STATS domain; **Purple** means overlapping between NLP and CV; **Blue** means overlapping between NLP and STATS.

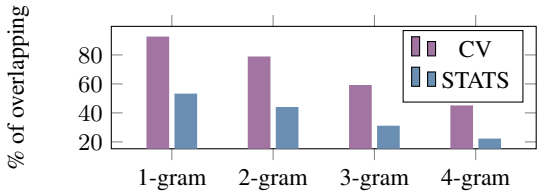

Figure 4: Percentage of overlapping n-grams.

pus differences relative to NLP. In Figure 4, we plot the percentage of overlapping n-grams of the {NLP, CV} and {NLP, STATS} domain pairs. This shows that NLP and CV have a larger overlap than {NLP, STATS} with respect to all of the n-grams ($n \in \{1, 2, 3, 4\}$). From this, we uphold that the classifiers trained on semantic features based on BERT models are valuable for bridging more distant domains with transfer learning.

To further contrast our findings, we enumerate the top 10 URLs in Table 6. Although the websites are ranked in different orders, there are still common URLs across the domains (highlighted in the table). Once again, CV shares a larger overlap with NLP in comparison to STATS. Along with the feature importance score, this cross-domain consistency further illustrates that the URL meta-features will benefit our model's out-of-domain classification. We show more feature statistics in the Appendix.

Comparison With Similar Datasets We compare a number of existing NLP educational datasets in Table 7, emphasizing the resource type, human effort for annotations, and corpus scale. Note that in this table, we only concentrate on human annotation efforts for free-text resources. This is because these free-text resources are the primary goal of the ERD Pipeline, as opposed to other tasks (e.g. learning concept relations, concept mining). We can see that MOOCcube (Yu et al., 2020) has a massive quantities of a single resource type (papers). They obtained the metadata from a third-party platform, AMiner, without a full round of human annotations. TutorialBank (Fabbri et al., 2018b) has a larger number of resources than LectureBank (Li et al., 2020), and it consists of diverse resource types. Our pipeline is very similar to TutorialBank in terms of resource type, but ours extends to more resources and subject areas, enabling us to research transfer learning across domains.

4 Application: Survey Generation for Lead Paragraphs

In this section, we demonstrate an interesting application that applies the resources discovered using our ERD Pipeline, Leading Paragraph Generation for Surveys.

Novel concepts are being introduced and evolving at a rate that creates high-quality surveys for web resources, such as Wikipedia pages, challenging. Moreover, such existing surveys like Wikipedia still needs human efforts on collecting relevant resources and writing accurate content on a given topic. Researchers have been investigating automatic ways to generate surveys using machine learning and deep learning methods. Survey generation is a way to generate concise introductory content for a query topic (Zhao et al., 2021). While most of the existing work focuses on utilizing Wikipedia to achieve this (Liu et al., 2018), little has been done for the web content. Since our ERD pipeline provides sufficient web data, we propose a two-stage approach for generating the lead paragraph that applies these web data selected from the ERD pipeline.

Name	Resource Type (with texts)	Domain Number	Annotation	Size
TutorialBank	**Lecture sides, papers, blog posts**	NLP only	**Manually**	6,300
LectureBank	Lecture sides only	NLP only	**Manually**	1,717
MOOCcube	Papers only	**Multiple**	Scrape from third-party	679,790
ERD (ours)	**Lecture sides, papers, blog posts**	**Multiple**	**Manually**	39,728

Table 7: Comparison with similar datasets.

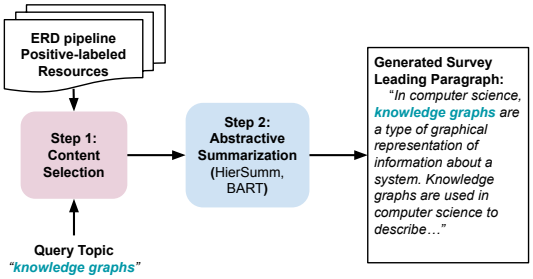

Figure 5: Two-stage Survey Generation Method.

4.1 Two stage method

We illustrate the two stage method in Figure 5. Given a query topic and high-quality web resources selected by ERD pipeline, we wish to generate the leading introductory paragraph for the query topic. This approach consists of content selection (step 1) and abstractive summarization (step 2). Content selection is the process of selecting the most relevant materials (including documents or sentences) according to the given query. Abstractive summarization generates the accurate lead paragraph from the selected materials.

Content Selection ERD pipeline is supposed to identify massive resources with broad coverage of the topics, so the first step is to select related content with the query topic.

While there is no suitable pretrained data for this task, and we do not collect survey data for training, we utilize the WikiSum dataset (Liu et al., 2018).

Methods	L=5	L=10	L=20	L=40
LSTM-Rank	39.38	46.74	53.84	60.42
Semantic Search	34.87	48.60	61.87	74.54
RoBERTa-Rank	**64.12**	**72.49**	**79.17**	**84.28**

(a) ROUGE-L (Lin, 2004) Recall scores for WikiSum content selection, varying the number of paragraphs returned.

Methods	R-1	R-2	R-L
HierSumm (Liu and Lapata, 2019)	41.53	26.52	35.76
BART (Lewis et al., 2019)	**46.61**	**26.82**	**43.25**

(b) ROUGE scores for intro generation.

Table 8: Two-stage method evaluation using WikiSum.

WikiSum contains 1.5 million Wikipedia pages, their references and their associated Google Search results. WikiSum includes many well-established topics and comprehensive reference documents, making it suitable for survey generation. We first evaluate content selection models using WikiSum. We experiment with three approaches in this step. Liu and Lapata (2019) undertake query-based content selection as a regression problem of predicting the ROUGE-2 recall of a given paragraph-topic pair (`LSTM-Rank`). Reimers and Gurevych (2019) fine-tune BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) to produce fixed-length vectors which can be compared using cosine similarity. We embed the topic of each Wikipedia page and candidate paragraph using this method, and select the paragraphs with the closest vectors to the title (`Semantic Search`). Additionally, we train RoBERTa in a similar manner as (Liu and Lapata, 2019). Then, we compare the query topic and paragraphs as sentence pairs and use the resultant relevance scores to for the paragraph ranking (`RoBERTa-Rank`). As shown in Table 8a, RoBERTa-Rank is the highest-scoring content selector, so we employ it for the abstractive summarization's input.

Abstractive Summarization This step is to generate summarization from the content selected previously. As a sequence-to-sequence task, there are many existing pretrained models to use. We experiment with `BART` (Lewis et al., 2019), a pre-trained model for text generation, as well as `HierSumm`, a hierarchical model from Liu and Lapata (2019). We show the summarization results on the WikiSum data in Table 8b, and observe that BART achieves the higher performance.

Evaluation	NLP	CV	STATS
Avg. Readability	3.45	2.90	2.75
Avg. Relevance	2.80	1.85	1.50
Avg. Non-redundancy	2.45	2.15	1.60

Table 9: Human Evaluation

NLP: knowledge graphs In computer science, knowledge graphs are a type of graphical representation of information about a system. Knowledge graphs are used in computer science to describe the structure of a system, such as a graph of knowledge about a particular system, ... or a list of items that can be used to describe a system 's structure. For example, a knowledge graph may contain information about an object 's shape, its size, its shape relative to other objects, its orientation, and its orientation relative to a point in space, or its position relative to another point in time, or any combination of these things. Knowledge Graphs can also be used as a way of describing the structure and structure of systems, such that a system can be described by a set of knowledge graphs that describe its structure and its properties.
CV: image retrieval In computer science, a Convolutional neural network is a type of neural network used in image analysis and image synthesis. It is a computer program that uses a neural network to learn from a large number of images, and can be used to predict the appearance of a given image in real-world situations, such as the shape of a surface, or the color of an object in a 3D environment such as an image of a person's face. It can also be used for image synthesis and image analysis, as well as image processing and image processing for computer vision and image recognition. It has been used in a number of applications such as computer vision, image processing, image recognition, and computer vision for image recognition and image rendering.
STATS: normal distributions In physics, normal distributions are a family of mathematical models that describe the distribution of normal distributions. They are used in the fields of statistics, physics , and computer science, and have been used in a wide variety of applications, including computer vision, image analysis, computer graphics, computer vision, computer graphics processing units (CPGs), as well as in the field of computer vision. They have also been used to study the visual appearance of real-world surfaces, such as the Phong Reflection Model, the Oren-Nayar model, the Koenderink et al. representation, and the Shading of the Sphere Model, and in the study of light sources and light sources in OpenGL and OpenGL-based graphics renderers.

Table 10: Examples of generated leading paragraphs.

4.2 Human Evaluation and Case Studies

So far we have shown that applying RoBERTa-Rank and BART as a two-step method gives promising results evaluated on the WikiSum dataset. We connect our pipeline with this method to generate the leading paragraph. We choose 10 queries randomly as survey topics in each domain, for example, "sentiment analysis "in NLP. A full query topic list is in the Appendix. Since we do not have ground truth, we conduct human evaluation and case studies.

We evaluate the model outputs on a 1-5 Likert scale based on the following qualities:

- *Readability*: attains a maximum score of 5 if the output is readable with a high degree of fluency and coherency.
- *Relevancy*: attains a maximum score of 5 if the output is perfectly relevant to the current topic with no hallucinations.
- *Non-redundancy*: attains a maximum score of 5 if the output has no repeating phrases/concepts.

We report average scores among 2 human judges of all topics by domain, shown in Table 9. The scores of NLP are the highest for all qualities, and STATS performed most poorly. This discrepancy may be caused by data collection bias, as more NLP resources were included.

We randomly pick one case study from each domain in Table 10. The model is able to generate leading paragraphs in a similar Wikipedia article style by giving a definition of a certain concept, following by descriptions of possible applications. Overall, while these surveys contains some facts, the quality can still be improved. For instance, the STATS paragraph exhibits some redundancy (e.g., "computer graphics","computer vision"). As an initial experiment, we have demonstrated the opportunities of extending our ERD Pipeline to produce survey paragraphs. In the future, we aim to enhance the generated lead paragraphs and extend the model for generating complete surveys.

5 Conclusion

In this paper, we proposed a pipeline for automatic knowledge discovery in novel domains. We applied transfer learning with a novel MLM pre-training method and achieved competitive classification performances. Moreover, we demonstrated two applications that take advantage of resource discovered by our pipeline. Finally, we released our source code and the datasets that we collected, including the 39,728 manually labelled web resources and 659 search queries. We plan to make this pipeline an online live educational tool for the public.

References

Iz Beltagy, Arman Cohan, and Kyle Lo. 2019. Scibert: Pretrained contextualized embeddings for scientific text. *CoRR*, abs/1903.10676.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *CoRR*, abs/2004.05150.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Uderstanding. *NAACL*.

Alexander Fabbri, Irene Li, Prawat Trairatvorakul, Yijiao He, Weitai Ting, Robert Tung, Caitlin Westerfield, and Dragomir Radev. 2018a. TutorialBank: A manually-collected corpus for prerequisite chains, survey extraction and resource recommendation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 611–620, Melbourne, Australia. Association for Computational Linguistics.

Alexander R Fabbri, Irene Li, Prawat Trairatvorakul, Yijiao He, Wei Tai Ting, Robert Tung, Caitlin Westerfield, and Dragomir R Radev. 2018b. Tutorialbank: A manually-collected corpus for prerequisite chains, survey extraction and resource recommendation. In *Proceedings of ACL*. Association for Computational Linguistics.

Klaus Krippendorff. 2011. Computing krippendorff's alpha-reliability.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising Sequence-to-sequence Pre-training for Natural Language Generation, Translation, and Comprehension. *arXiv preprint arXiv:1910.13461*.

Irene Li, Alexander Fabbri, Swapnil Hingmire, and Dragomir Radev. 2020. R-VGAE: Relational-variational graph autoencoder for unsupervised prerequisite chain learning. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1147–1157, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Irene Li, Alexander R. Fabbri, Robert R. Tung, and Dragomir R. Radev. 2019. What should I learn first: Introducing lecturebank for NLP education and prerequisite chain learning. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6674–6681. AAAI Press.

Irene Li, Vanessa Yan, Tianxiao Li, Rihao Qu, and Dragomir Radev. 2021. Unsupervised cross-domain prerequisite chain learning using variational graph autoencoders. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences. *arXiv preprint arXiv:1801.10198*.

Yang Liu and Mirella Lapata. 2019. Hierarchical Transformers for Multi-document Summarization. *ACL*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A Robustly Optimized Bert Pretraining Approach. *arXiv preprint arXiv:1907.11692*.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Soto Montalvo, Jesus Palomo, and Carmen de la Orden. 2018. Building an educational platform using nlp: A case study in teaching finance. *J. UCS*, 24(10):1403–1423.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Cristóbal Romero and Sebastián Ventura. 2017. Educational data science in massive open online courses. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 7(1):e1187.

Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-june Paul Hsu, and Kuansan Wang. 2015. An overview of microsoft academic service (mas) and applications. In *Proceedings of the 24th international conference on world wide web*, pages 243–246. ACM.

Jie Tang, Limin Yao, Duo Zhang, and Jing Zhang. 2010. A combination approach to web user profiling. *ACM TKDD*, 5(1):1–44.

Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. Arnetminer: Extraction and mining of academic social networks. In *KDD'08*, pages 990–998.

Jifan Yu, Gan Luo, Tong Xiao, Qingyang Zhong, Yuquan Wang, Wenzheng Feng, Junyi Luo, Chenyu Wang, Lei Hou, Juanzi Li, Zhiyuan Liu, and Jie Tang. 2020. MOOCCube: A large-scale data repository for NLP applications in MOOCs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3135–3142, Online. Association for Computational Linguistics.

Mingjun Zhao, Shengli Yan, Bang Liu, Xinwang Zhong, Qian Hao, Haolan Chen, Di Niu, Bowei Long, and Weidong Guo. 2021. QBSUM: A large-scale query-based document summarization dataset from real-world applications. *Comput. Speech Lang.*, 66:101166.

A Chosen topics for Human Evaluation in Survey Generation

Table 11 shows the randomly selected topics for survey generation, 10 from each domain.

NLP
adam optimizer
lstm model
dropout neural networks
recursive neural network
convolutional neural network
automatic summarization
sentiment analysis
attention mechanism deep learning
Pre-trained Language Models NLP
knowledge graphs
CV
transfer learning
convolutional neural network
image retrieval
image classification
feature learning
seq2seq
transformers
visual question answering
conditional probability
k means
STATS
linear regression
hypothesis testing
conditional probability
multinomial distribution
probability density
density estimation
normal distributions
bernoulli distribution
standard deviation
z-score

Table 11: Topics selected for human evaluation.

B More Sample Queries

We list more sample queries in Table 12, such queries are applied in the Data Collection step of the proposed pipeline.

NLP Sample Queries
“markov decision processes" site:.edu filetype:.pdf
“sentiment analysis" site:.edu filetype:.pptx
“unlexicalized parsing" site:kdnuggets.com filetype:.html
“semantic parsing" site:.edu filetype:.pdf
“information retrieval" site:.edu filetype:.pptx
“monte carlo methods" site:rubikscode.net filetype:.html
“natural language processing intro" site:.edu filetype:.pdf
“sequence to sequence" site:.edu filetype:.pptx
“naive bayes" site:paperswithcode.com filetype:.html
“latent dirichlet allocation" site:.edu filetype:.pdf
CV Sample Queries
“epipolar geometry" site:.edu filetype:.pptx
“particle filters" site:hackernoon.com filetype:.html
“image registration" site:.edu filetype:.pdf
“reflectance model" site:.edu filetype:.pptx
“shading analysis" site:skymind.ai filetype:.html
“imaging geometry and physics" site:.edu filetype:.pdf
“texture classification" site:.edu filetype:.pptx
“gibbs sampling" site:kdnuggets.com filetype:.html
“image thresholding" site:.edu filetype:.pdf
“region adjacency graphs" site:.edu filetype:.pptx
STATS Sample Queries
“linear regression" site:rubikscode.net filetype:.html
“hypothesis testing" site:.edu filetype:.pdf
“heteroscedasticity" site:.edu filetype:.pptx
“random event" site:paperswithcode.com filetype:.html
“maximum liklihood" site:.edu filetype:.pdf
“granger causality" site:.edu filetype:.pptx
“probability" site:hackernoon.com filetype:.html
“random sampling" site:.edu filetype:.pdf
“correlation coefficient" site:.edu filetype:.pptx
“chi-squared statistic" site:skymind.ai filetype:.html

Table 12: More sample queries used in the three selected domains, varying site and file type.

C BERT models for Group 3 features

The three main deep features were extracted using the following pre-trained models:

BERT-base
https://huggingface.co/bert-base-uncased.
SciBERT
https://huggingface.co/allenai/scibert_scivocab_uncased.
Longformer
https://huggingface.co/allenai/longformer-base-4096.

D More Data Statistics

In Table 13, we show token-level and sentence-level statistics of our collected data.

	NLP	CV	STATS
Token Number/per sentence			
Mean	18.28	26.37	23.28
Median	12	19	18
Max	2,302	458,363	20,066
Sentence Number			
Mean	161.60	122.49	107.32
Median	55	46	52
Max	5,929	21,301	52,793

Table 13: Free text statistics by domain.

E Meta-Feature Distributions

In the following pages, we show the histograms of the 18 quantitative meta-features collected for each data point. Recall from Table 4 that these features were segregated into two groups. Group 1 features are higher-level and generally pertain to the document layout. Group 2 features focus on more specific aspects of the resource's URL and free text.

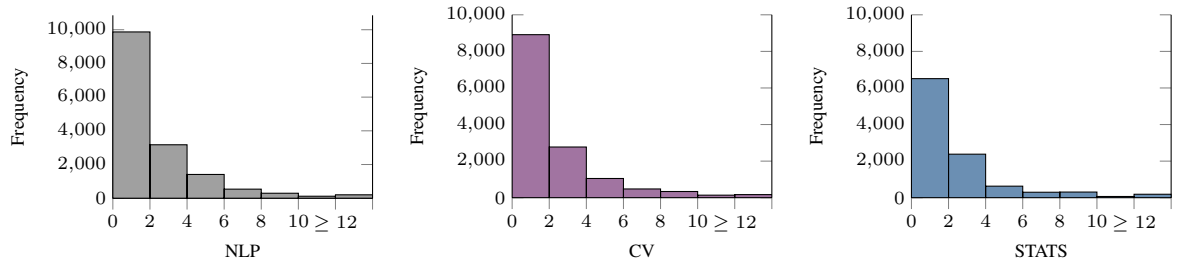

Figure 6: *NumAuthor* Distribution

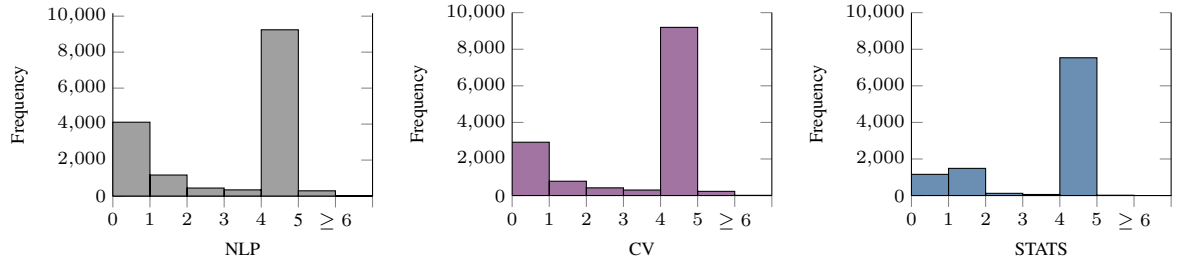

Figure 7: *NumHeading* Distribution

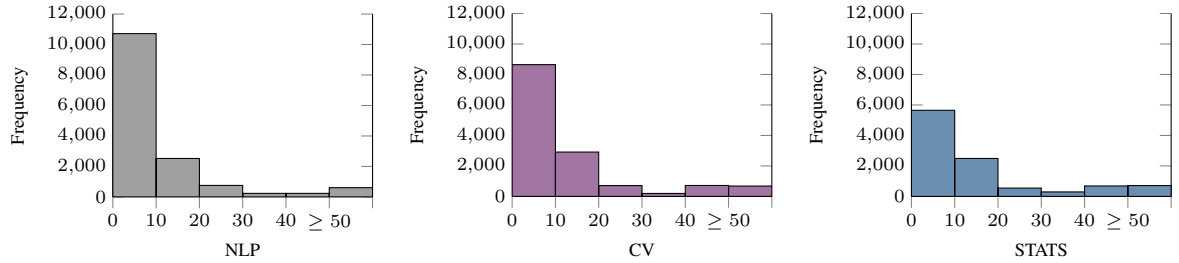

Figure 8: *NumFig* Distribution

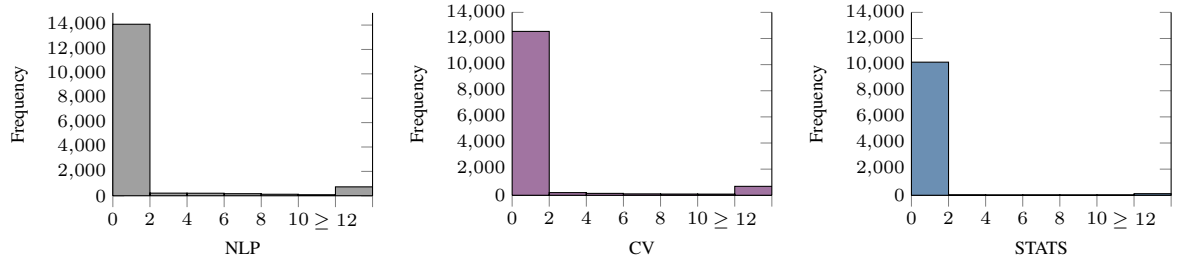

Figure 9: *NumEqu* Distribution

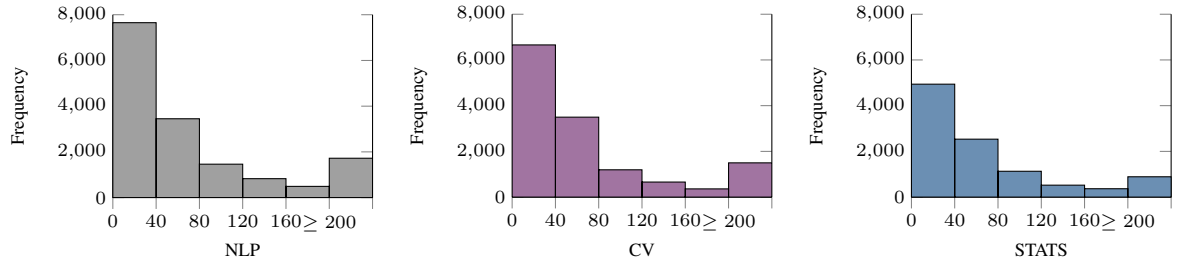

Figure 10: *NumPara* Distribution

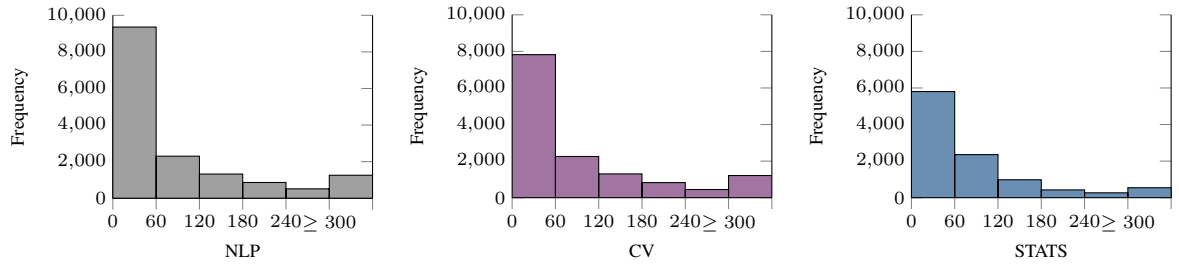

Figure 11: *NumSent* Distribution

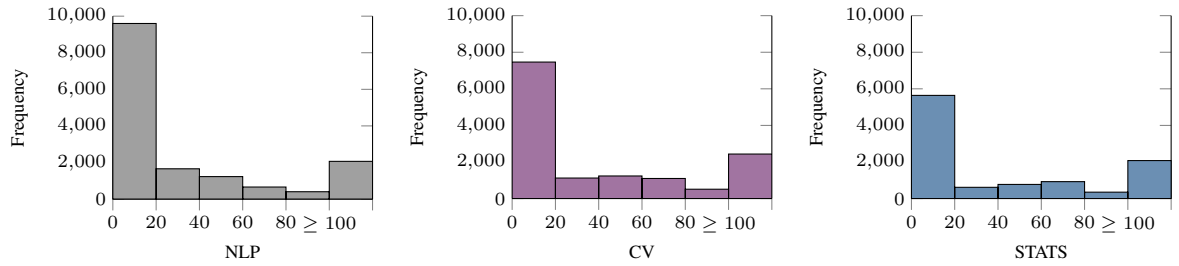

Figure 12: *NumLink* Distribution

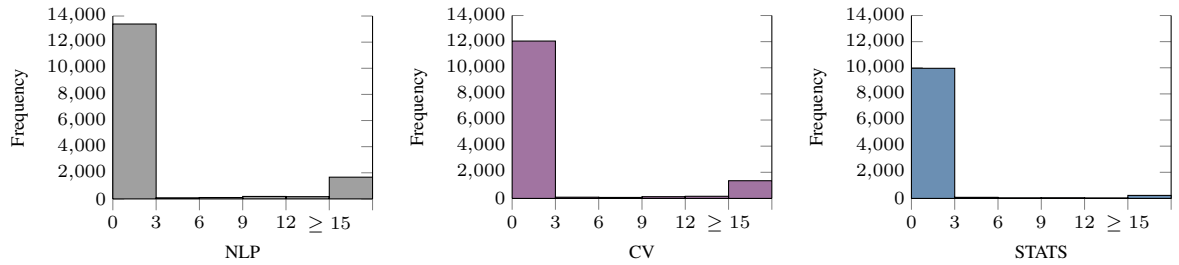

Figure 13: *BibLen* Distribution

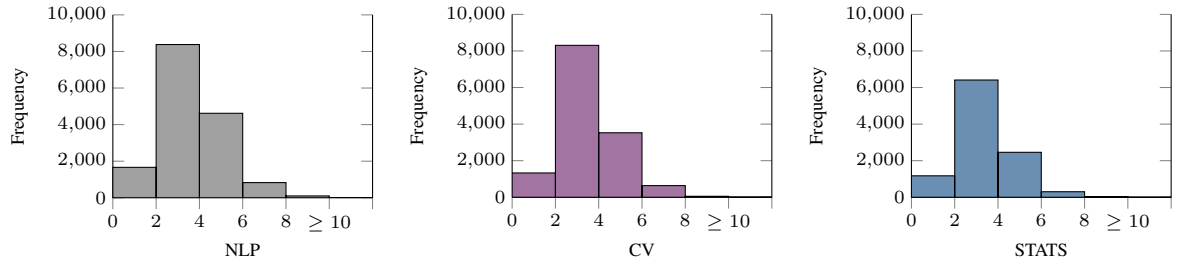

Figure 14: *NumUrlSubdir* Distribution

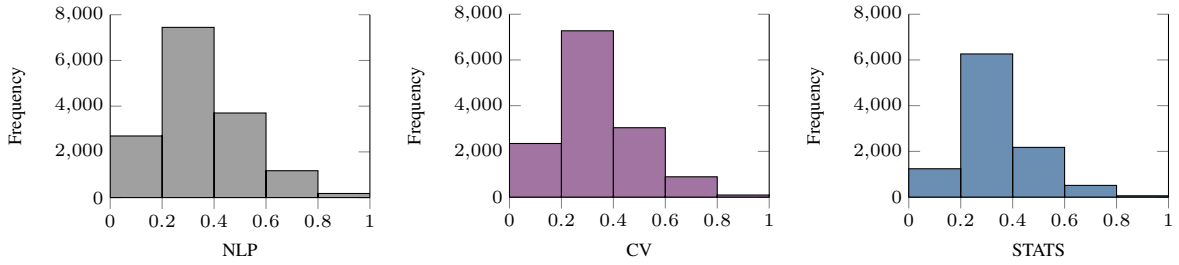

Figure 15: *NormalizedUniqueVocab* Distribution

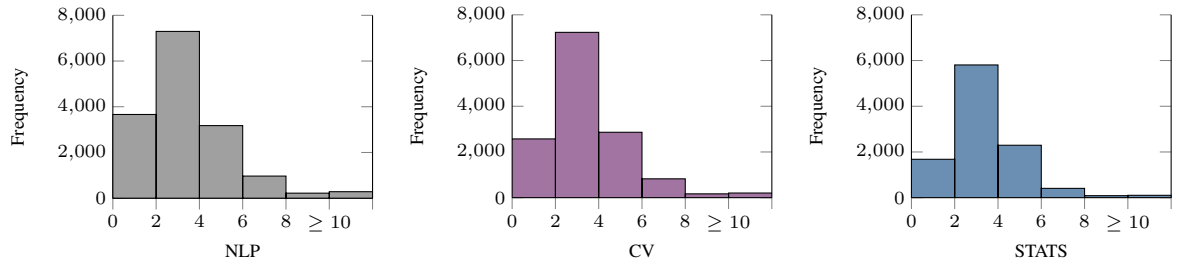

Figure 16: *UniqueVocabMean* Distribution

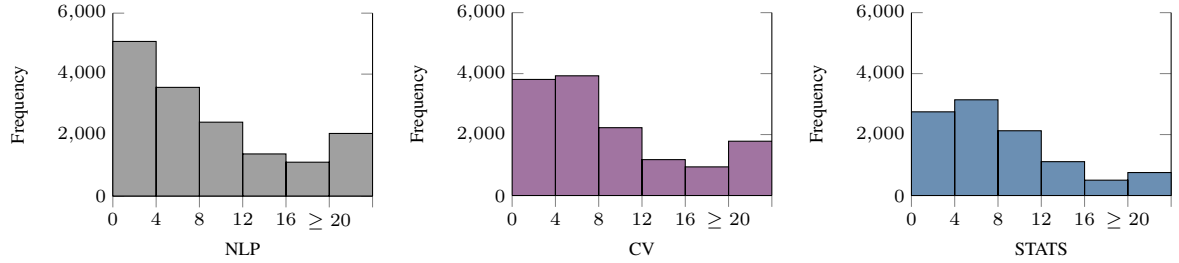

Figure 17: *UniqueVocabStdev* Distribution

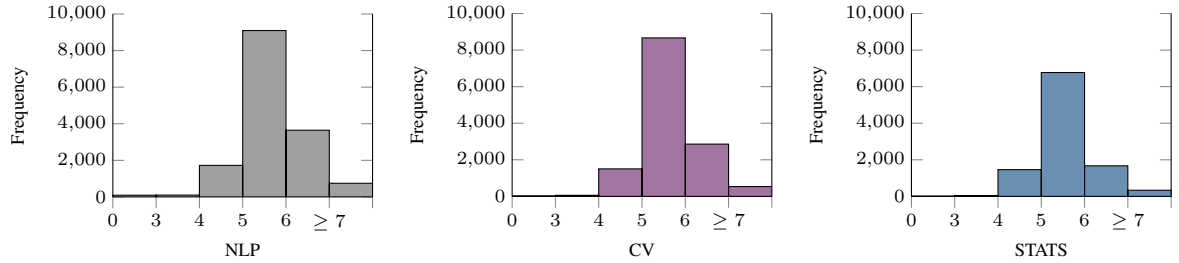

Figure 18: *WordLenMean* Distribution

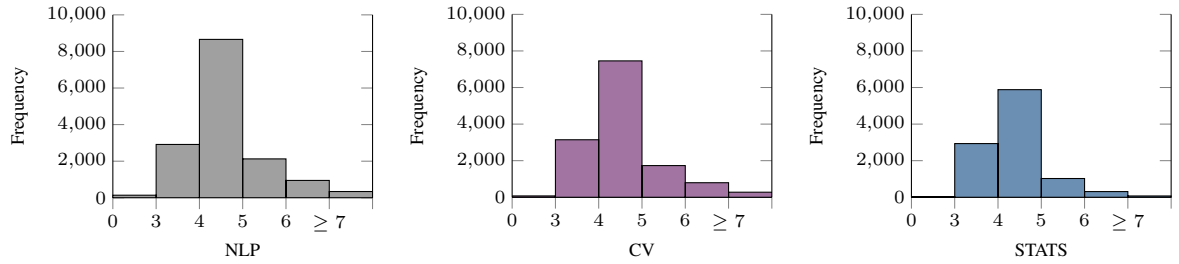

Figure 19: *WordLenStdev* Distribution

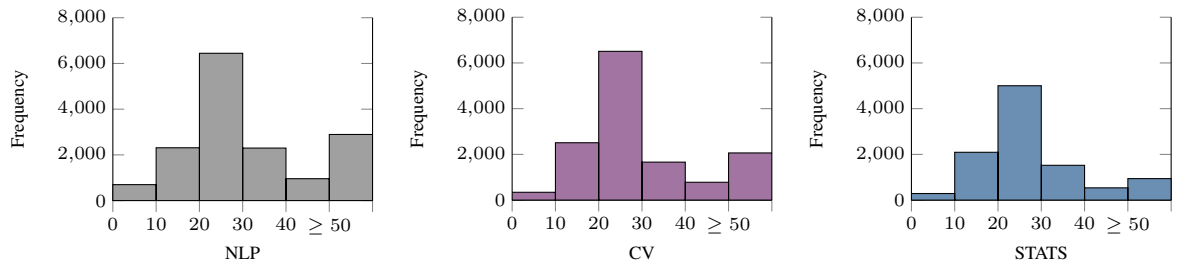

Figure 20: *SentLenMean* Distribution

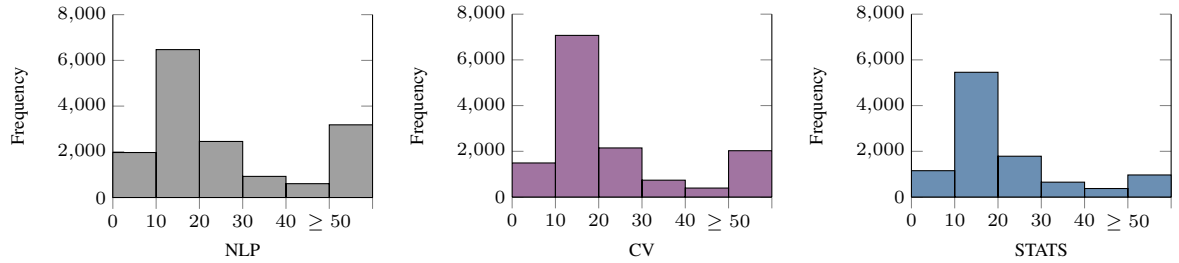

Figure 21: *SentLenStdev* Distribution

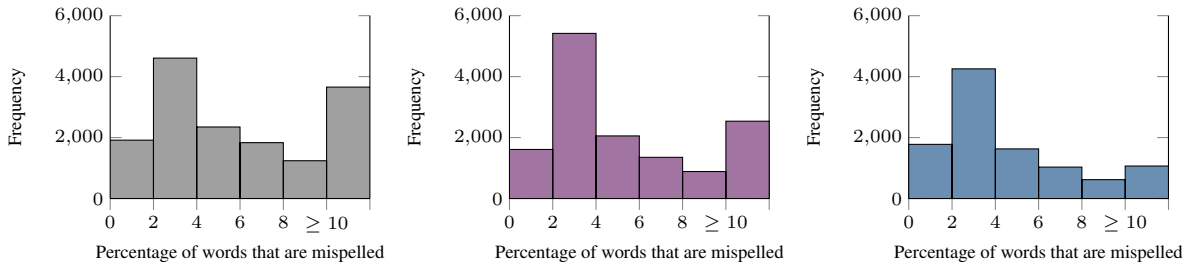

Figure 22: *PercentTypo* Distribution

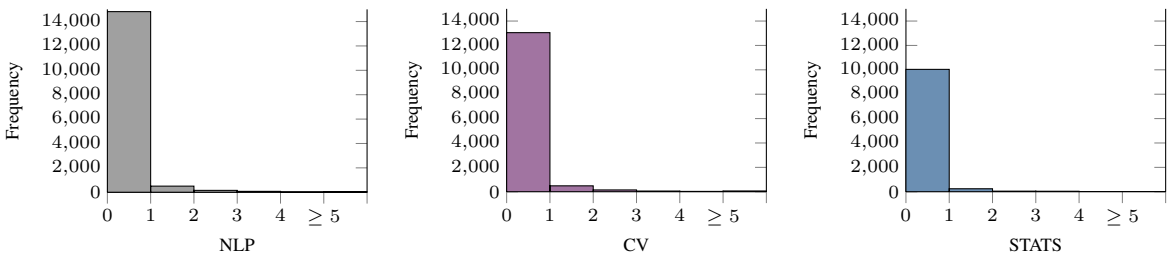

Figure 23: *NumGithubLink* Distribution