Online Appendix
"The Education-Innovation Gap"

Barbara Biasi* Song Ma†

*Yale School of Management and NBER, barbara.biasi@yale.edu, +1 (203) 432-7868;
†Yale School of Management and NBER, song.ma@yale.edu, +1 (203) 436-4687.

1. University Syllabus Data

In this appendix we provide additional information on data processing, measure construction, and empirical analysis. Section 1 provides details on the syllabus data collection and processing. Section 2 provides details on the academic publication data. Section 3 provides details on the NSF/NIH grant data. Section 4 discusses the process of building the dictionaries used in our textual analysis. Section 5 provides additional details on the construction of the education-innovation gap measure.

1.1. Syllabus Data Collection and Overview

The university syllabus data are obtained from Open Syllabus Project (OSP)[1]. The dataset includes more than three million syllabi, collected from early three thousand colleges and universities in the United States, citation metrics that link against 1.8 million unique titles (papers, textbooks, articles, etc.). Additionally, each syllabus in the dataset is annotated with metadata about where and when the course was taught.

In the OSP dataset, the following variables are used in our variable construction and analysis:

Table 1. Variable List from OSP Database

Variable Name	Description
ID	The unique identifier assigned to each syllabus.
Text	The text of the syllabus.
Text md5	The md5sum of the text, which can also be used as unique identifier.
Language	The language of the document.
Year	The academic year that the syllabus was taught.
Field name	The name of the academic field most associated with the syllabus.
Institution ID	The unique identifier for the institution matched to the syllabus.
UNITID	The IPEDS unique identifier for the institution.
Country code	The ISO 3166-1 alpha-2 code of the country the syllabus was taught in.
Institution name	The name of the institution the course was taught in.

In the paper, the analysis focuses on those syllabi that satisfy the following criterion.

(i) From a four-year university based in the United States (country_code is equal to "US")

[1] https://opensyllabus.org

(ii) Syllabus language is English

(iii) Year from 1998 to 2018

(iv) Extracted syllabus length (wordlength_extracted) must be in [20, 10000] (1.58% drop rate)

(v) Number of syllabi per institution (inst_count_syl) larger than 100

(vi) remove syllabi from online-only universities

To complement the syllabus data, we also collected university course offerings from university catalog. We do so by first randomly identify 10 percent of all universities in our sample (212 universities), and then we manually search and download electronic copies (usually in the PDF format) of university catalogs for those universities for all years available. 161 out of the 212 universities have at least one such catolog books available. A total of 2,348 catalogs are downloaded and processed for those 161 universities (14.5 catalogs per university). Those universities, due to the random-selection algorithm, are representative of the full sample in all school-level characteristics. A balance test across school characteristics on the full sample and the catalog sample are provided in Table 2.

University catalog data provide the following information: course code, course name, and course level (classified into Basic, Advanced, and Graduate). Some course catalogs also provide a brief course description.

Table 2. School Characteristics of Schools In and Out of Catalog Data

	Mean for Catalog Sample # Institutions = 158	Mean for Full Sample # Institutions = 1,956	t-statistics	p-values
ln Expenditure on instruction (2013)	8.693	8.601	-1.725	0.085
ln Endowment per capita (2000)	6.857	6.483	-1.304	0.193
ln Sticker price (2013)	9.197	9.153	-0.520	0.603
ln Avg faculty salary (2013)	8.890	8.850	-1.897	0.058
ln Enrollment (2013)	8.708	8.634	-0.685	0.494
Share Black students (2000)	0.109	0.112	0.153	0.879
Share Hispanic students (2000)	0.063	0.065	0.183	0.855
Share alien students (2000)	0.025	0.022	-1.030	0.303
Share grad in Arts & Humanities (2000)	7.581	7.958	0.382	0.703
Share grad in STEM (2000)	14.861	14.050	-0.772	0.440
Share grad in Social Sciences (2000)	21.068	19.202	-1.342	0.180

Note: Balance test of universities in and out of the catalog sample.

1.2. Extracting Course Content of Syllabi

The section describes the method of processing the text of syllabus. The full text of a syllabus comes in the variable "text" from the OSP database. The steps taken to transform the variable into usable content are: (i) cleaning the text variable (e.g., removing html language from the web scraping, removing errors from the OCR process); (ii) identifying syllabus sections; and (iii) extracting course content information and removing text unrelated to course content (e.g., course policy, etc.).

1.2.1. Cleaning the Raw Text Files. The OSP syllabus text requires additional cleaning before performing textual analysis. The cleaning process involves two parts:

(i) We use Unidecode Python Package[2] to convert the Unicode text into ASCII text, including those legacy code that doesn't support Unicode, non-Roman names on a US keyboard and ASCII approximations for symbols and non-Latin alphabets.

(ii) We remove the browser information in the header of the syllabus raw text using the keyword list ("Internet Explorer", "Newer Browser", "JavaScript Enabled", "Cookies Are", "Download Info", "Login", "Log In", "Print", "Search").

1.2.2. Separating and Identifying Syllabus Sections. In general, a syllabus will contain the following contents that are more relevant for our analysis: instructor information, course description, course requirement, course objective, outline, homework, exam and other policies. Meanwhile, the syllabus text often include other information that are not directly related to the purpose of our study and should be ideally removed from the analysis, including honor code, policies related to disability, class room laptop and cellphone policies, among others.

We develop a supervised algorithm to divide the text into different sections, using a set of section title keywords. We first manually sort out the keywords of the section title by reading raw syllabus data. Those different section titles can be classified into several different categories—course description, requirements, course objectives, course outline, course materials, professor information, assignment-related information, grading policy,

[2] https://pypi.org/project/Unidecode/

policies regarding non-academic issues, misc notes. For example, course description includes keywords like"description","instruction","introduction", etc. Table 3 provides the entire list of keywords.

Armed with this list of section keywords, our algorithm separate different sections of the syllabus by combining those words and the formatting norm in a syllabus file. In Figure 1, we use a part of one syllabus as an example, and present step-by-step illustrations of our processing.

- First, for each syllabus, we identify the section titles based on the word list described above and the formatting features. We mark all cases when the section title phrases appear as all uppercases or consecutive initial capital letters using regular expressions.
  - In Figure 1, those phrases with underline satisfy the features of section title, such as"Advanced Recombinant Techniques" and "Course Description".
- In the Step 2 of Figure 1, the whole syllabus is divided into 8 parts and we use the Arabic numerals to mark them out. Finally, we select those sections with important section title and get the cleaned text.
  - In Figure 1, we select those sections with blue and underline section title, including"Required Text","Course Description","Objectives The" and"Course Outcomes Upon".

1.3. Identifying Additional Information From Syllabus Text

1.3.1. Instructor Names. We build a neural network model to identify instructor names from the syllabi. We use the BiLSTM-CNNs-CRF model for named entity recognition (NER) to extract instructor name.[3]

The training/test dataset is built up via the following three steps:

(i) We first select those syllabi that contains any keyword of the list ("Doctor", "Doctors", "Dr", "Professor", "Prof", "Instructor", "Instructors", "Tutor", "Tutors") in the first 3,500 characters.

[3]BiLSTM-CNNs-CRF model for named entity recognition (NER) (Ma and Hovy, ACL 2016, End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF).

Figure 1. The Structure of a Sample Syllabus

Econ 561a Yale University Fall 2005
Prof. Tony Smith (Part I) Prof. Michael Keane (Part II)
Syllabus for **COMPUTATIONAL METHODS FOR ECONOMIC DYNAMICS** ECON 561a

Course Objectives:
Most of the dynamic economic models used in modern quantitative research in economics do not have analytical (closed-form) solutions. For this reason, the computer has become an indispensable tool for conducting research in dynamic economics. The goal of this two-part course is precisely to teach students computational tools for conducting numerical analysis of dynamic economic models. The focus of the first half of the course, taught by Prof. Tony Smith, is on solving dynamic programming problems and on computing competitive equilibria of dynamic economic models. The first half of the course also provides an introduction to some of the basic tools of numerical analysis, including minimization, root-finding, interpolation, function approximation, and integration. The focus of the second half course, taught by Prof. Michael Keane, is on solving and estimating discrete-choice dynamic programming models of economic behavior. Taken together, the two halves of the course provide students with a thorough introduction to the numerical analysis of dynamic economic models in both microeconomics and macroeconomics.

Contact Information (Prof. Tony Smith)
Office: 28 Hillhouse, Room 306 Office phone: (203) 432-3583
Email address: tony.smith@yale.edu Course Web site: www.econ.yale.edu/smith/econ561a
Office hours: Thursdays from 10AM–noon, or by appointment

Class Meetings:
The course meets on Mondays and Wednesdays from 2:30PM to 3:50PM in a room to be determined.

Prerequisites:
This course is designed for graduate students in economics who have taken first-year graduate courses in microeconomics, macroeconomics, and econometrics. No prior knowledge of either numerical methods or computer programming is assumed, but some familiarity with a programming language would prove helpful.

Texts:
The required textbook for this course is:
Numerical Recipes in Fortran 77: The Art of Scientific Computing, Second Edition (Volume 1 of Fortran Numerical Recipes) by William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery (Cambridge University Press, 1992). This book, as well as its (optional) companion Numerical Recipes in Fortran 90: The Art of Parallel Scientific Computing, Second Edition (Volume 2 of Fortran Numerical Recipes), is available online at: www.library.cornell.edu/nr/.
Other (optional) books that students might find useful are:
- Numerical Methods in Economics by Kenneth L. Judd (MIT Press, 1998).
- Handbook of Computational Economics (Volume 1), edited by Hans M. Amman, David A. Kendrick, and John Rust (North-Holland, 1996).
- Computational Methods for the Study of Dynamic Economies, edited by Ramon Marimon and Andrew Scott (Oxford University Press, 1999).
- Dynamic Economics: Quantitative Methods and Applications by Jérôme Adda and Russell Cooper (MIT Press, 2003).
- Applied Computational Economics and Finance by Mario J. Miranda and Paul L. Fackler (MIT Press, 2002).

Grading:
The course grade will be based on two (equally-weighted) projects, one for the first part of the course and one for the second part of the course. Each project consists of writing a program in Fortran to solve an assigned problem. Students must submit their code as well as a brief (roughly five pages) description of their numerical findings. The first project will involve solving for the competitive equilibrium of a dynamic macroeconomic model; the second project will involve solving and estimating a discrete-choice dynamic programming model. Fortran is the language of choice for most researchers in computational economics; requiring that the code for the projects be written in Fortran will help students to become proficient in this powerful and useful language. The first project is due on Monday, November 14 and the second project is due at the end of the semester. Occasional short programming problems may also be assigned as the course proceeds. The purpose of these assignments is to help students develop the skills they need to complete the projects; these assignments will not be graded.

Approximate Schedule of Lectures (Part I)
I. **INTRODUCTION**
Lecture 1 Introduction to numerical dynamic programming (built around the stochasticgrowth model and the Aiyagari (1994) model). General considerations in numerical analysis: convergence, roundoff error, truncation error. Numerical differentiation.
Readings:
- Aiyagari, S.R. (1994), "Uninsured Idiosyncratic Risk and Aggregate Saving," Quarterly Journal of Economics 109, 659–684.
- Numerical Recipes: Chapters 1 and 5.7
- Judd: Chapters 1, 2, and 7.7

II. **BASIC NUMERICAL METHODS**
Lecture 2 Root-finding in one or more dimensions: bisection, secant method, Newton's method, fixed-point iteration, Gauss-Jacobi, Gauss-Seidel, Brent's method.
Readings:
- Numerical Recipes: Chapter 9

……

Note: This figure shows a sample syllabus from the OSP database, in the original format. The subsections are identified using the algorithms as described above.

Table 3. Section Title Keywords List

Section Title Keywords of Important Content
 # Course Description
Syllabi, Syllabus, Title, Description, Method, Instruction, Content, Characteristics, Overview, Tutorial, Intro, Abstract, Methodologies, Summary, Conclusion, Appendix, Guide, Document, Module, Introduction, Approach, Lab, Background
 # Requirement
Requirement, Applicability, Required
 # Objective
Objective, Achievement, Outcome, Motivation, Purpose, Statement, Skill, Competency, Performance, Goal
 # Outline
Outline, Schedule, Timeline, Guideline
 # Material
Text, Material, Resource, Recommend, Reference, Book, Calendar, Textbook, Guidebook
 # Professor information
Instructor, About, Email, Phone, Contact, Professor, Staff, Faculty, Information
 # Project, homwork, paper, and exam
Personal, Total, Individual, Exercise, Essay, Submission, Assign, Homework, Paper, Final, Examing, Midterm, Term, Semester, Proposal, Application, Demonstration, Program, Task, Report, Pracical, Drafting, Project, Plan, Deadline, Makeup, Advising, Advisor, Survey, Assignment, Planning, Practice, Group, Participation, Team, Research, Activity, Complaint, Design, Analysis, Strategy, Procedure, Working, Work, Exam, Examination, Training, Professional, Test, Case, Discussion, Grade, Presentation, Quiz, Essay, Layout, Sample, Rewrite
 # Grade
Assessment, Point, Scope, Evaluation, Record, Grading, Composition, Review
 # Policy
Academic, Justice, Administration, Rule, Discipline, Disclaimer, Regulation, Standard, Affair, Dishonesty, Plagiarism, Misconduct, Offence, Medical, Absent, Absence, Trip, Religious, Observance, Ttendance, Honesty, Origination, Originator, Help, Technology, Attendance, Accessing, Service, Oppotunity, Administrative, Accommodation, Support, Policy, Right, Responsibility, Disability, Weather, Integrity, Copyright
 # Note
Remark, Notice, Additional, Acknowledgement, Absolutely, Absolute, Important, Note, Cannot, Can, Must, Should, Will, Please, No
 # Other Words
Course, Lecture, Catalog, Campus, Commuity, Class, Classroom, College, Univerity, Discussion, Seminar

Note: The table shows the word list used to identify subsections of a syllabus. Only the singular terms are listed, and in the implementation we also include the plural forms for each keyword.

(ii) We use the Spacy[4] package to identify whether the words following those keywords are

[4] https://spacy.io/

Table 4. Summary Statistics of Open Syllabus Project

	# of records	Syllabus word length (raw)	Syllabus word length ("knowledge content")
Original data	6,852,971		
Keep syllabus based in the United States (Syllabus language is English)	3,995,483		
Keep syllabus from four-year university	1,951,933	2,725.41	1,435.09
Year from 1998 to 2018	1,937,284	2,732.09	1,436.77
Extracted syllabus length must be in [20, 10000]	1,901,367	2,279.66	1,057.35
Number of syllabi per institution larger than 100	1,882,224	2,274.55	1,056.77
Remove syllabus from online-only universities	1,752,795	2,218.08	1,010.82

names of people (entity label is "PERSON").

(iii) We process the syllabus text sentence by sentence as the training and test data of the model.

(iv) **Model performance on the testing data set:** precision 85.18%, F1 84.98.

We then apply the model to the whole OSP data. We also apply a few further filtering: (a) we remove single letter names (Professor Biasi); (2) all the words in the name are required to appear in the English first and last name data set[5]; (c) after the first two filters, we only keep the first instructor name. Model performance on the full OSP sample: 86.23% of syllabi that has a instructor name.

1.3.2. Course-Levels: Basic, Advanced, Graduate. We train a Natural Language Processing (NLP) model to identify the course level for each syllabus. We use the training

[5] https://github.com/philipperemy/name-dataset

sample data set with syllabus text as input feature, and course level extracted manually from university catalog data as labels. The model is Distilled BERT[6] (Sanh et al. 2019) via the transformers[7] library. In our training, the out-of-sample prediction precision is 85.04%.

To build the training/testing dataset: all the syllabi that appear in the university catalog data (56,831 syllabi). The matching process between the syllabi and university catalog data is with institution and course code. For those 57 thousand syllabi, we label the course as basic course if the catalog is undergraduate-level and the first digit of the course code is 1 or 2; we label the course as advanced course if the catalog is undergraduate-level and the first digit of the course code is 3 or 4; we label the course as graduate course if the catalog is graduate-level or the first digit of the course code is larger than 4. The input of the model includes all description text from OSP.[8]

1.4. Syllabus References and Recommended Readings

In addition to the text and metadata of the syllabi, there are two other datasets in the OSP, Matches and Catalog Records. Matches dataset links between syllabi and catalog records, where each records represents a single instance in which a title is assigned in a syllabus. Catalog Records is the set of 1.8 million bibliographic records that are assigned to at least one syllabus. In the Matched dataset, the following variables are kept:

- ID: The unique identifier of the match.
- Doc ID: The id of the syllabus that contains the match.
- Work ID: The work_id of the catalog record that is assigned to the match.

In the Catalog Records dataset, the following variables are kept:

- Work ID: The unique identifier of the catalog record.
- Publication type: The type of publication of the work. Possible values are "journal" and "book".

[6] https://arxiv.org/abs/1910.01108

[7] https://huggingface.co/transformers/index.html

[8] OSP extracts structured text fields from the syllabus by a token-level sequence tagging model. https://opensyllabus.github.io/osp-dataset-docs/syllabi.html#description

- Publication year: The year the work was published.

1.5. Syllabus Field

Table 5. Categorization of Course (Macro-)Fields

Macro-field	Fields
Business	Business, Accounting, Marketing
Humanities	English Literature, Media / Communications Philosophy, Theology, Criminal Justice Library Science, Classics, Women's Studies Journalism, Religion, Sign Language Music, Theatre Arts, Fine Arts, History Film and Photography, Dance, Anthropology
STEM	Mathematics, Computer Science, Biology Engineering, Chemistry, Physics Architecture, Agriculture, Earth Sciences Basic Computer Skills, Astronomy, Transportation Atmospheric Sciences
Social Sciences	Psychology, Political Science, Economics Law, Social Work, Geography Linguistics, Sociology Education
Vocational	Fitness and Leisure, Basic Skills Mechanic / Repair Tech, Cosmetology Culinary Arts, Health Technician, Public Safety

Note: Mapping between the "macro-fields" used in our analysis and syllabi's "fields" as reported in the OSP dataset.

2. Academic Publication Data

In this appendix, we describe the process of collecting and cleaning academic publication data.

2.1. Journal List

The first step is to define our sample of academic journal. The field categories and ranking of academic journal is obtain from Incites Journal Citation Reports (JCR) [9]. JCR is a resource tool published annually by Thomson Reuters (formerly ISI) to provide citation and publication data of academic journals in the science and social science fields. We choose journals that were ranked as top ten in the respective field for at least once. This yields 3,962 journals among 223 academic publication fields. A full list of those journals are provided in our data package.

2.2. Data Collection

After defining our list of journals, we next collect information on all the papers that were published in those journals. These data come from Scopus.[10] Scopus is Elsevier's abstract and citation database. Using Scopus's API, we obtain the metadata of each article. We send data requests through the API. Specifically, we use the ISSN of one journal as query ("ISSN(0022-1082)"), and capture the metadata of all articles of the specific journal for all years. In the Scopus data, Table 6 lists all variables that are kept [11]:

Table 6. Scopus Variable List

Variable Name	Description
EID	Electronic ID, used as the unique identifier of each article.
Title	Title of the article.
Issn	The ISSN of publisher.
Cover Date	Publication date.
Description	Abstract of the article.
Authkeywords	Kewords of the article.

[9] https://jcr.clarivate.com/

[10] https://www.scopus.com

[11] For a more detailed and comprehensive list of variables, please see https://dev.elsevier.com/guides/ScopusSearchViews.htm

In Table 7 we describe the coverage of the sample.

Table 7. Sample Description of the Academic Publication Data

	# of records	average word length
Original data	20,779,713	2184.630
		(86.625)
Filter out those syllabus without abstract text	16,617,641	2184.630
		(86.625)
Remove copyright information	16,617,641	94.568
		(47.178)

2.3. Data Cleaning

The key textual information used in our analysis is the abstract of each article from the "Description" variable from Scopus. The data cleaning process of publication abstract mainly focuses on the removal of non-related copyright disclaimers. At the beginning or end of the abstract, there is usually a copyright disclaimer from the journal or author. We use keyword recognition techniques to clean up the data. Specifically, for the first sentence, once a keyword such as"copyright" (the keywords include"copyright","©","published","publisher","all right","all rights reserved"), we define it as a copyright disclaimer and remove the whole sentence. Repeat this step until no new sentences are identified as copyright notices. The same is true for the ending sentence.

2.4. Microsoft Academics Data

We use information from Microsoft Academic (MA) to measure the research output of all people listed as instructors in the syllabi. The Microsoft Academic data are downloaded from Microsoft Academic Knowledge Graph (MAKG)[12]. The MAKG is a large RDF knowledge graph with over eight billion triples containing information about scientific publications and related entities, including authors, institutions, journals, and fields of study. The data set is based on the Microsoft Academic Graph and licensed under the Open Data Attributions license. For each researcher, Microsoft Academic lists publications, working papers, other

[12]We downloaded the data based on the Microsoft Academic Graph data as of 2020-05-29 from https://zenodo.org/record/3936556#.YFndr2Qza3J

manuscripts, and patents, together with the counts of citations to each of these documents. Due to differences in counting citations, Microsoft Academic's citation history does not match to other similar services like Web of Science or Google Scholar. But those services' citations numbers are highly correlated with each other.

We link instructor records from the text of the syllabi to Microsoft Academic records using names, history of institutions and research fields. After restricting the sample of syllabi in which the instructor can be matched to a unique MA researcher, we are able to successfully match 38.93% percent of all instructors identified from the syllabus data. Out of all the syllabi that has a instructor name (2,433,683), 34.77% (846,287) of them can be matched to MA, 25.86% (629,365) of them have duplicated matching. In our selected syllabus sample, the matching ratio is 38.93% (= 682,286 / 1,752,795).

3. NSF/NIH Grant Data

In this appendix, we describe the process of collecting and cleaning grant data from the National Science Foundation (NSF) and the National Institutes of Health (NIH). These data are used to generate measures at the school level and instructor level related to the level of government grants. We first describe the data source, and then we will describe the matching process between the University Syllabus Data and NSF/NIH grant data respectively.

3.1. Data Source

Grant data are officially provided by the National Science Foundation (NSF)[13] and the National Institutes of Health (NIH)[14]. The versions used in the paper are dated as of May 25, 2021.

The NSF grant data includes 480,633 grants with effective year ranging from 1960 to 2022. The NIH grant data includes 2,566,358 grants with effective year ranging from 1978 to 2021. Both NSF and NIH grant data contain the institution information (institution name, country, state, and city) and investigator information (investigator name and role). For investigator role, there are four types of roles in NSF data, which are principal investigator (PI), co-PI, former PI, and former co-PI, and there are two types of roles in NIH data, i.e., contact and non-contact.

3.2. Match NSF/NIH Institutions to Schools

We first build school-year level grant measures. To do so, we match the NSF/NIH institutions to IPEDS schools. We take advantage of the detailed information from both NSF/NIH grant data and IPEDS, including the school name and location (country, state, and city).

Since there is no unique institution id in the NSF/NIH data, we perform a many-to-one matching between NSF/NIH institutions and IPEDS schools, that is, we assign a unique IPEDS school ID to each NSF/NIH institution. We proceed in two steps. First, we adopt a exact match approach where the names of NSF/NIH institutions and IPEDS school are exactly

[13] https://www.nsf.gov/awardsearch/download.jsp
[14] https://exporter.nih.gov/ExPORTER_Catalog.aspx

the same after we remove the punctuation and special stop words (including "and" and "the"). Next, for the remaining NSF/NIH institutions, we use a fuzzy matching approach based on the name and location. We require the matching to meet the following two conditions: (1) the two institutions must be in the same city; (2) the fuzzy matching ratio must be larget than a certain threshold—specifically, we use partial ratio and token set ratio in the FuzzyWuzzy Package.[15] Naturally, this leads to repeated institutions in some cases. Therefore, we adopt a "refined" matching approach. We keep the IPEDS school with the largest average matching ratio for each NSF/NIH institution.

After the matching, for NSF grant data, we observe 21,265 unique institution name, and 11.30% (2,402) of them are matched to 2,088 IPEDS schools, which covers 82.05% (= 394,383 / 480,633) NSF grants. For NIH grant data, we observe 23,388 unique institution name, and 6.73% (1,573) of them are matched to 1,236 IPEDS schools, which covers 66.53% (= 1,707,498/2,566,358) NIH grants. The unmatched NSF grant institutions are often firms, non-university research institute, etc.

3.3. Match NSF/NIH Investigators to Instructors

We next match investigators of grants with course instructors in the syllabus data. There is no unique person identifier across data sets, so the matching is mainly performed through fuzzy matching. The NSF and NIH data provide different investigator information that are used in the fuzzy matching, so the matching methods differ slightly.

First, we describe the matching process between NSF investigators and syllabus instructors. In this process, we first perform a name disambiguation process within NSF based on the first name, last name, email, and institution of the investigators. We label two investigators as the same person if (1) they share the same email, or (2) they have exactly the same first name and last name in the same school in a certain year. Next, we perform a many-to-one fuzzy matching between NSF investigators and syllabus instructors based on the names and history of institutions the researcher was employed at. We proceed in three consecutive steps:

(i) Each NSF investigator name is matched with syllabus instructor names after removing

[15]The package uses Levenshtein Distance to calculate the differences between sequences, and its homepage is https://github.com/seatgeek/fuzzywuzzy, and we use a threshold of 80.

the punctuation. The matching score is calculated using the Whoswho Package[16], which is a python library for determining whether two names describe the same person. If an unique exact match is identified where the matching score is equal to 100, we consider this as a successful matching.

(ii) We consider those matching with matching score larger than 95 in the same schools.

   (a) If an NSF investigator and a syllabus instructor have overlapped time in the same school, we choose the matching with the maximum matching score.

   (b) Otherwise, if there is not overlapped time in the same school between the two people, we choose the matching with shortest time distance and maximum matching score.

(iii) If still unmatched, we consider those matching with matching score larger than 95 in the different schools.

   (a) We first consider those matching without overlapped time between the two people. We choose the matching with shortest time distance and maximum matching score.

   (b) Otherwise, we choose the matching with the maximum matching score if they have overlapped time.

Thus, we assign a unique syllabus instructor to each NSF investigator.

For NIH investigators, we use the unique identifier provided by the NIH grant data and require a one-to-one matching between the NIH investigators and syllabus instructors. We use the similar process as the matching process of NSF grant data. Out of all the syllabus instructors, 15.46% (= 68,236 / 441,452) instructors at least have one grant from NSF or NIH grant database, and these instructors come from 2,237 IPEDS schools. These instructors have 190,738 syllabi, accounting for about 12.82% (= 190,738 / 1,487,820) of the syllabi with instructor. For NSF grant data, after processing, we observe 232,206 disambiguated investigator, and 23.31% (= 54,118 / 232,206) of them can be matched to one syllabus instructor, which covers 44.28% (= 208,857 / 471,646) grants. For NIH grant data, processing,

[16] https://github.com/rliebz/whoswho

we observe 298,687 unique investigator, and 10.07% (= 30,087 / 298,687) of them can be matched to one syllabus instructor, which covers 17.69% (= 450,339 / 2,546,123) grants.

4. Term Extraction

In this appendix, we explain in details the term extraction process in our textual analysis. In summary, this process identifies and extracts the content words that are later used in the analysis, and remove words that are less useful in describing the syllabus content. To do so, the first step is to build a dictionary of these words. We do so by compiling a list of all unique words and terms that were ever entered as a key word in academic publications. We collect these keywords using the same method in Section 2.

Next we convert the text content in textual documents into numerical data for statistical analysis. We use an term extraction algorithm, NGramMatch, to parse text of both the syllabus data and the academic publication data. Before using the term extraction algorithm, we use Unidecode Python Package[17] to convert the Unicode text into ASCII text, including thost legacy code that doesn't support Unicode, non-Roman names on a US keyboard and ASCII approximations for symbols and non-Latin alphabets. Then we convert all capitalized characters to lowercase and use the NLTK Python Toolkit to strip out all non-word text elements, such as punctuation, numbers, and HTML tags. Next, we remove all occurrences of 280 "stop word", which include propositions, punctuation, pronouns, and other words that carry little semantic content.[18]

Then we apply the NGramMatch algorithm to the remaining list of "unstemmed" (that is, withour removing suffixes). NGramMatch algorithm performs exact string matching based on N-grams extracted from text. First, the algorithm extracts N-grams (N ranges from 1 to 7) from text, and takes the union into a basic term set. Secondly, we filter out those terms which cannot be found in the specific dictionary. Thus, we can get a term set of text and its frequency vector.

[17] https://pypi.org/project/Unidecode/

[18] We use the stopwords list using the union of all single letters and Stanford CoreNLP package: https://github.com/stanfordnlp/CoreNLP/blob/master/data/edu/stanford/nlp/patterns/surface/stopwords.txt.

5. Additional Details in the Education-Innovation Gap

5.1. Cosine Similarity

In this section, we provide additional details in the algorithm of calculating the textual similarity based on the modified-version of term-frequency-inverse-document-frequency ($TFIDF$). The similarity algorithm is performed between the syllabus textual data and the publication textual data.

First of all, the modified-version of $TFIDF$ can be decomposed into two matrix, document term matrix (DTM) and document inverse matrix (DIM).

<u>DTM.</u> For the DTM, donated as C, each row corresponds to each document in the corpus. Each element of C, denoted c_{dw}, is the number of times a given term (indexed by w) appears in document d. In specific, We donated C^S as DTM of the syllabus data and C^B as DTM of benchmark data of academic publication. Each row of C^S corresponds to a syllabus with a syllabus year, and each row of C^B corresponds to the set of all the documents in the benchmark data in the particular year. So we can calculate the DTM for each text data source using the formula as:

$$TF_{dw} \equiv \frac{c_{dw}}{\sum_k c_{dk}}. \tag{1}$$

<u>DIM.</u> For the DIM, donated I, each row corresponds to the vector of backward inverse document frequency (backward-IDF, $BIDF$) in the particular year. More specifically, backward-IDF is defined as:

$$BIDF_w \equiv \log(\frac{\text{\# of documents in sample prior to } d}{\text{\# of documents prior to } d \text{ that include term w}}). \tag{2}$$

So our modified version of $TFIDF$ is defined as:

$$TFBIDF_{w,d,t} = TF_{w,d} \times BIDF_{w,t} \tag{3}$$

and its normalized version that has unit length,

$$V_{d,t} = \frac{TFBIDF_{d,t}}{\|TFBIDF_{d,t}\|}. \tag{4}$$

Then we can calculate the textual similarity. Specifically, for each document pair $(d; d')$, where d is a syllabus in University Syllabus Data with a syllabus year and d' is the set of all the documents in benchmark data in the particular year, the cosine similarity is defined as:

$$\rho_{d,d'} = V_{d,t} \cdot V_{d',t}; \quad t \equiv \min(d, d') \tag{5}$$

This calculation can be done using the DTM and DIM of University Syllabus Data and benchmark data.

5.2. Defining the Education-Innovation Gap Measure

We construct the education-innovation gap as the ratio between the average similarity of a syllabus with older technologies (published in τ) and the similarity with more recent ones $(\tau' < \tau)$:

$$Gap_d \equiv \left(\frac{S_d^{\tau}}{S_d^{\tau'}} \right)$$

It follows that a syllabus published in t has a lower education-innovation gap if its text is more similar to more recent research than older research. In our analysis, we set $\tau = 13$ and $\tau' = 1$, and we scale the measure by a factor of 100 for readability.

It is worth emphasizing the advantage of a ratio measure over a simple measure of similarity, say for example, S_d^1. In particular, the latter could be sensitive to idiosyncratic differences in the "style" of language across syllabi in different fields, or even within the same field. Take the example of two syllabi that teach (nearly) identical materials in regression analysis. One syllabus, say A, explains things in greater details and more intense use of terminologies than the other one, say B. As a result, syllabus A has a closer similarity to scientific publications, near or far (i.e., both τ and τ'), due to the detail-oriented style, i.e., $S_A > S_B$.

A ratio of similarity measures *for the same syllabus* is instead free of any time-invariant, syllabus-specific attributes. The ratio definition would correct this problem because the style of syllabus are net out, and the ratio itself only captures the new terms in the syllabus relative to the old ones. Intuitively, we use the similarity with the old corpus to net out the

syllabus style fixed effect.

A Simulation Exercise. We illustrate this point using a simulation exercise. The idea of the simulation exercise is the following. We construct simulated syllabi with a known gap but with different "styles" (to be defined below). We then construct the similarity and gap measures for those simulated syllabi. By comparing those measures and the real known gap measure used for the simulation, we are able to evaluate the best measure to recover the true gap value.

Simulated syllabus and its "true" gap. Each simulated syllabus is determined by three parameters: a syllabus year (t), a real education-innovation gap (gap), and a parameter governing the style of the syllabus ($style$). These parameters govern the way the syllabus draws terms from three different buckets of words: new knowledge terms, old knowledge terms, and style words.

- New knowledge terms are those that are (1) among the top 5 percent in the new publication corpus between $t-3$ and $t-1$ or (2) in the new publication corpus between $t-3$ and $t-1$ but not in the old publication corpus between $t-15$ and $t-13$.

- Old knowledge terms are those knowledge terms that are (1) among the top 5 percent in the old publication corpus between $t-15$ and $t-13$ or (2) in the old publication corpus between $t-15$ and $t-13$ but not in the new publication corpus between $t-3$ and $t-1$.

- Style words are those terms that are in the knowledge corpus but not among the top knowledge terms.

We generate each simulated syllabus through the following algorithm, using parameters gap and $style$:

- Each syllabus is given the length of L words, and L is generated using a uniform distribution between 10 and 500, with an increment of 10

- We draw $L^{style}(= L \times style)$ words from the style words bucket

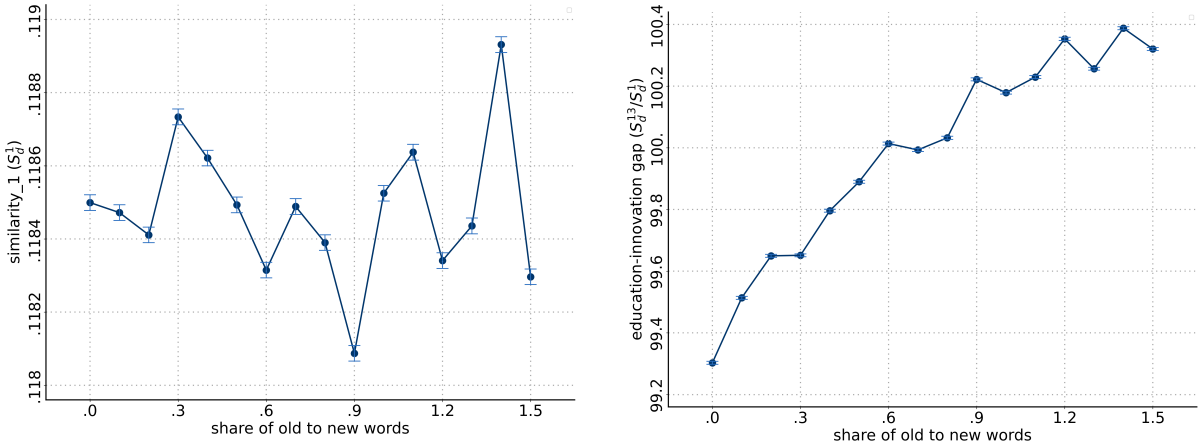
- The remaining $L^{knowledge}(= L \times (1 - style))$ words are drawn from the new knowledge terms bucket $L^{knowledge} \times (1 + gap)^{-1}$, and from the old knowledge terms bucket $L^{knowledge} \times gap \times (1 + gap)^{-1}$

- Parameters $t \in \{t : Range(1998, 2018, 1)\}$, $L \in \{L : Range(10, 500, 10)\}$, $style \in \{style : Range(0.01, 0.10, 0.01)\}$, $gap \in \{gap : Range(0, 1.5, 0.1)\}$

We generate 1,700,000 such simulated syllabi (close to the total number of syllabi used in our analysis). For each parameter pair, $(t, L, style, gap)$, we generate 10 syllabi, so the total number of simulated syllabi is $1,700,160 = 10 \times 21\ t \times 46\ L \times 11\ style \times 16\ gap$.

Figure 2. Simulated Syllabi and Their "True" Gap Measure

(a) Similarity with New Publication (S_d^1) **(b)** Ratio Gap (G_d)



Note: The figure show the relationship between different gap measures and the "true" gap measure in the simulation exercise of Section 5.2. The gap measures S_d^1 in Panel (a) and Gap_d in Panel (b) are defined in Section 5.2. We measure the "true" gap measure, *gap* as the relative ratio of old vs. new word counts.

Figure 2 shows the simulated syllabi's S_d^1 and G_d with respect to the real parameter *gap*, which governs the relative ratio of old vs. new word counts. We show that the G_d ratio performs significantly better in capturing the underlying *gap* parameter, while the S_d^1 does a poor job, likely due to the noises from the style parameter.