



Mathematics Clinic

Final Report for
PilotCity

Automating an Engine to Extract Educational Priorities for Workforce City Innovation

May 2, 2019

Team Members

Madison Hobbs (Project Manager)
Jean Selasi Adedze
Dominique Macias
Xuming Liang
Aanya Alwani

Advisor

Dr. Talithia Williams

Liaison

Derick Lee, PilotCity

Abstract

Our clinic team was tasked with developing software and algorithms to automate PilotCity programming and to extract educational insights from unstructured data sources like websites, syllabi, resumes, and more. Our solutions involved creating web interfaces for PilotCity users to engage with, designing a recommender system that facilitates the process of matching employers to high school classrooms, and employing topic modeling techniques to extract educational priorities of an institution through all its syllabi. Our team also investigated the creation of an automated study guide using insights from topic models.

In this report, we detail our approaches, implementation, results, future directions, and the impact of our work.

Acknowledgements

We wish to thank PilotCity and the Institute of Education Sciences, U.S. Department of Education for sponsoring our yearlong project through the Harvey Mudd College Mathematics Clinic Program. We also extend our gratitude to Professor Talithia Williams for being a fantastic advisor as she supported and guided our process. Thank you to the Mathematics Clinic Director Professor Weiqing Gu and to Clinic Coordinator DruAnn Thomas for facilitating and managing the Mathematics Clinic program this year.

We would also like to acknowledge our collaboration with the PilotCity team which has grown over the past year and who took on the majority of website design, implementation, and maintenance after our initial contributions. This includes Eric Reyes, Jerold Inocencio, Kura Peng, Camila Ramos, and CEO & Founder of PilotCity, Derick Lee.

Contents

Abstract	i
Acknowledgements	iii
1 Executive Summary	1
2 Introduction	3
2.1 Sponsors	3
2.2 Problem Statements	3
2.2.1 PilotCity	3
2.2.2 IES	4
3 Automation of PilotCity Programming	5
3.0.1 Background	5
3.0.2 Web Interface For PilotCity Users	5
3.0.3 Recommender System	8
3.0.4 Evaluation of Recommender System	9
3.0.5 Impact	10
3.0.6 Code	11
4 Extracting Insights From Text Data	13
4.1 Methods	14
4.2 Assessing Insights	15
4.2.1 Topic Coherence	15
4.3 LDA vs NMF	16
4.4 Extracting insights	16
4.4.1 Data	17
4.4.2 Visualization of Results	17
4.4.3 PilotCity Deliverable	18

4.5	Further work with Topic Modeling	20
5	Conclusion	21
6	Future Work	23
6.0.1	Improvements to Recommender System	23
6.0.2	Possible Extensions and Applications of Insights Engine	23
	References	25

Chapter 1

Executive Summary

The goal of this project is to optimize automation, scalability, and user engagement of PilotCity programming. Specifically, we want to automate a process which has historically been done by hand. As PilotCity grows, matching employers and classrooms by hand will be intractable. Our work will enable PilotCity to expand, reaching more classrooms and helping more students gain work-based learning experience.

The first step was to create an onboarding website, where employers and teachers could log in and input all their relevant details. These user inputs were then used to generate a list of recommendations for users, which could be further filtered. This website is also being used as a platform for students, teachers, and employers to engage with each other, and look at their milestones.

Working off of designs provided by PilotCity, we started building the frontend of this website. This required us to try our hand at HTML, CSS, and JavaScript. Upon each successive draft, we showcased our product to PilotCity users and gained valuable feedback from them. We then helped the PilotCity local team onboard and take over this section while we moved onto building the employer-classroom recommendation system. We conducted more user interviews with both teachers and employers in order to gauge what they thought would be relevant for our recommendation system.

The main algorithm behind the system involves scoring unknown input. Rather than give users a long list of predefined responses, our user feedback suggested that a limited set of open-ended questions would be better able to capture the diverse interests and skill sets of our users. Our scoring for open-end text responses is currently based on the GloVe model. This

model is designed to take in words as an input and outputs vectors reflecting semantic meaning and relationships between words based on how often they appear in similar contexts.

This algorithm is already incorporated into the website so that an employer logged in is able to see a ranked list of all the classrooms based on their preferences and filters, and likewise a logged in teacher can see a ranked list of employers for each of their classrooms. This ranking is based on the similarity score of their responses as determined by the GloVe model.

Our project also focused on extracting educational priorities from uncured data sources, such as class syllabi, school handbooks, and employer websites. We decided to use Topic Modeling to visualize high level themes represented in an input document.

Topic Modeling is a statistical model that represents documents by a specific number of topics. For each input document, a topic is represented by a list of words ranked by their relevance within the topic. We experimented with two types of topic models, Latent Dirichlet Allocation (LDA) and Non-negative Matrix Factorization (NMF).

We built a module that pre-trains a topic model on an input set of PDF documents and then extracts and visualizes the topic distribution of a single input PDF document. This module provides a structure for general data, restricted only by its format as a PDF. We tested our module on a set of course syllabi extracted from the website of Las Positas College, which contains all syllabi for all courses offered. We also investigated the creation of an auto-generated study guide using the top 3 words in the 5 most relevant topics pertaining to the syllabus of a particular classroom.

In Chapter 2, we give an introduction to this clinic project. In Chapter 3, we document the web development process, which included the recommender system that dominated the first semester of clinic. Specifically, we will discuss our approaches, our findings, our implementations, and the societal import of our work. In Chapter 4 we provide an overview of our topic modeling techniques and visualizations, in Chapter 5 we provide a conclusion, and finally, in Chapter 6 we discuss some future work.

Chapter 2

Introduction

2.1 Sponsors

Our project is unique because we have two independent stakeholders (IES and PilotCity) who have different but connected areas of interest.

- **Institute of Education Sciences (IES)** is the statistics, research, and evaluation arm of the U.S. Department of Education. Their mission is to provide scientific evidence on which to ground education practice and policy and to share this information in formats that are useful and accessible to educators, parents, policymakers, researchers, and the public (Institute of Education Science Website, 2019).
- **PilotCity**, on the other hand, is a startup that was created to help transform small to medium sized cities into innovation engines by converting local high school classrooms into workforce incubators. The motivation behind PilotCity is to generate talent within a city, rather than attract it from the outside. To realize this vision, PilotCity connects local high schools and employers thereby empowering students at an early age and promoting the idea of project based learning.

2.2 Problem Statements

2.2.1 PilotCity

In the past, PilotCity has manually performed logistical tasks like recruiting, signing up teachers and employers and matching employers to classrooms. However, manual program delivery is a bottleneck to PilotCity's capacity to

scale and thus PilotCity aims to automate most of their processes like enrollment and matchmaking with the aid of the Harvey Mudd Clinic program. Our team has thereby been tasked with improving user engagement and scalability of PilotCity programming by helping design and create a web interface that students, teachers and employers can better interface with as well as building a recommender system to facilitate matching employers to high school classrooms.

2.2.2 IES

While we help facilitate PilotCity programming, we are also tasked by IES to research and come up with new ways to gain insights on educational priorities and approaches from uncured educational resources like course websites, syllabi and teacher resumes. To fulfill this task, we decided to employ topic modeling techniques on a collection of syllabi pertaining to a college to make inferences on the nature of a college's curriculum. This exploration was motivated by work by Sekiya et al who present a similar investigation to ours but with a slightly different angle and scope (Sekiya, Matsuda, & Yamaguchi, 2017). Rather than looking at all course subjects across time at one school, they narrow the focus to computer science curriculum across multiple schools within the same time frame. Furthermore, rather than having to train a topic model themselves, they leverage the pre-existing CS2013 Body of Knowledge (BOK), produced by the ACM and IEEE Computer Society and detailing the 18 primary topics in Computer Science curriculum as of 2013. Sekiya et al. used those 18 topics to train a simplified, supervised Latent Dirichlet Allocation model (ssLDA) which then output, for a given unseen computer science syllabus, how much each of those 18 core topics were represented. However, since we lack predefined topics, our work involves training unsupervised topic models to discover the core topics across all disciplines at Las Positas College in Livermore, California.

Chapter 3

Automation of PilotCity Programming

3.0.1 Background

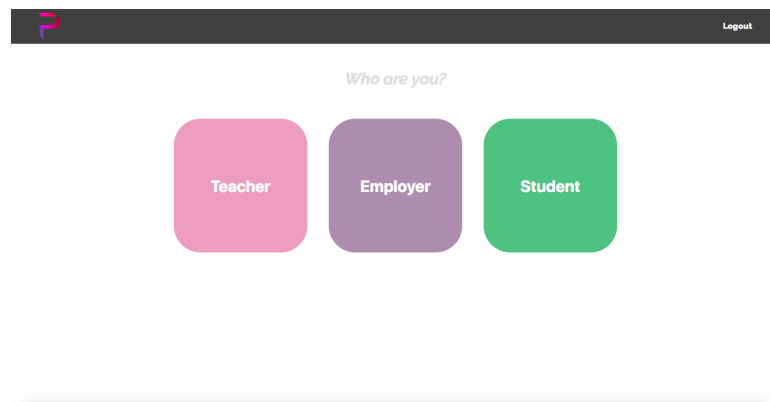
This project requires consideration of high level social problems related to education, including an analysis of the effectiveness of Project Based Learning versus Work Based Learning. It also includes analyzing more standard technical problems such as matching PilotCity users, and building out a platform with which these users can interact. The various challenges in this project combine into a unique problem space involving the analysis of uncurated data for educational advancement.

At the start of this project, we had limited access to historic user data from PilotCity. We conducted a series of user interviews to incorporate feedback into the construction of a recommender system. We interviewed students, teachers, employers, and school district administrators to gather information regarding pain points of PilotCity's past programming, the value of Project Based Learning, and ideas for an online platform for PilotCity. We compiled interview notes and accordingly planned a web application that can onboard and recommend matches between employers and classrooms.

3.0.2 Web Interface For PilotCity Users

Our initial technical work involved creating the front-end onboarding platform to start building the web application. We created a sequence of pages in which teachers and employers could fill in relevant information that allowed us to make the most optimal recommendations. Once we had a functional

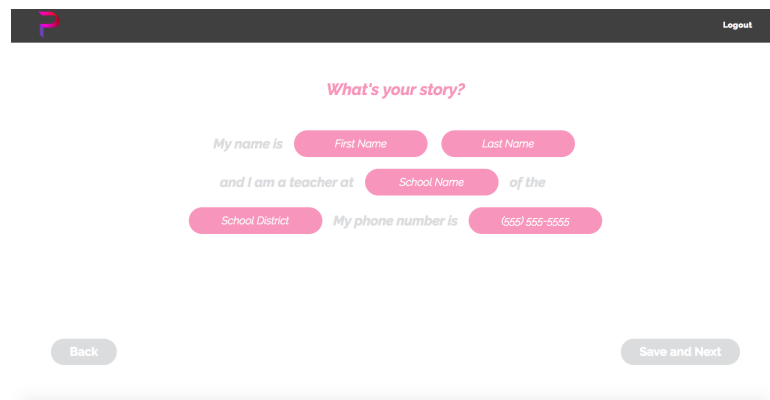
front-end product, we handed off the front-end work to PilotCity's local team. Figures 3.1-3.5 demonstrate the front-end flow built by the team, as viewed by an onboarding teacher.



Who are you?

Teacher Employer Student

Figure 3.1 Front-end view of what the user sees when beginning their PilotCity onboarding process. The user would then proceed accordingly, specifying what user category they would fall into.



What's your story?

My name is First Name Last Name

and I am a teacher at School Name of the

School District My phone number is (555) 555-5555

Back Save and Next

Figure 3.2 The next page seen by an onboarding teacher. Here, the teacher is asked about the school they teach at and their contact information.

The form is titled "Classes" and has a "Logout" link in the top right. It contains two rows of input fields. The first row is pre-filled with "P3" for Period, "AP Physics" for Course Name, "Spring" for Semester, "Selected 10, 11, 12" for Grade, and "65-70 Students" for Class Size. The second row has "Enter Course Name" for Course Name, "Select" for Semester, "Select" for Grade, and "Select" for Class Size. Below these rows is a large blue rounded rectangle with a red "+" sign in the center. At the bottom are "Back" and "Save and Next" buttons.

Figure 3.3 Front-end view where the onboarding teacher is asked about the classes they want to involve in the PilotCity program.

The form is titled "Industries" and has a "Logout" link in the top right. Below the title is a subtitle "What industries would your classrooms be excited about?". The main content area displays a grid of 14 industry tags, each with a red "+" icon and a red "x" icon. The tags are: Trades, Healthcare, Drones, Robotics, Data Science, Internet Of Things, Sustainability, Space, Artificial Intelligence, Automotive, Bioprinting, Data, Drones, Gaming, Healthcare, and Lifestyle. At the bottom are "Back" and "Save and Next" buttons.

Figure 3.4 Front-end view where the onboarding teacher is asked about the industries that their classrooms would be interested in working with.

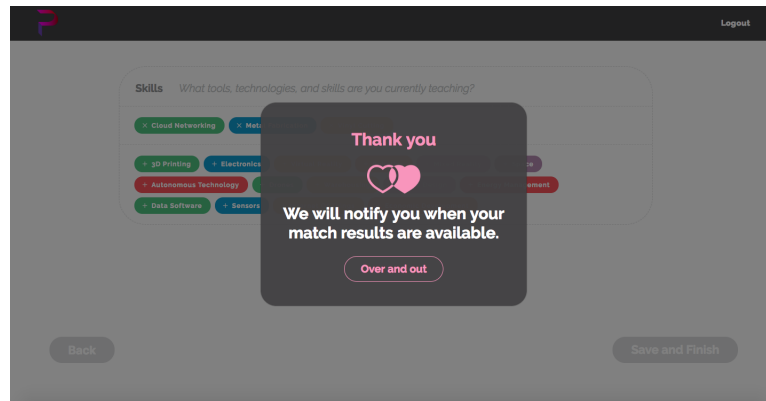


Figure 3.5 View that the teacher would see once finishing the onboarding process. This page indicates to the user that their information has been saved, and that they will be contacted as soon as employer recommendations are available for each of their classrooms.

3.0.3 Recommender System

We planned and implemented a recommender system to recommend classrooms to employers and employers to teachers based on similarities in industry and skills. We score classrooms for an employer and employers for a teacher using the GloVe Model, which represents words as vectors capturing semantic meaning. The similarity score between two words is taken to be the cosine of the angle between the vector representation of the two words. We further adapt the GloVe model's output to be able to take in two phrases with multiple words, such as "Machine Learning" and "Artificial Intelligence" and compute their similarity. This is done by taking the average of each pairwise similarity between individual words.

The final score between an employer and a classroom is a weighted combination of the similarity scores between the inputs of employers and teachers. For each employer, we consider their industry, service, product, their vision for working with high school students, and the location of their workplace. For each classroom, we consider the course name, the industry preference of the teacher, the tools, technologies, and skills taught in the classroom, and the location of the school. The UI of the final recommendation system, as viewed by a logged in employer, is shown in figure 3.6 below. The final recommender system was meant to serve as a baseline ordering from which teachers and employers could easily choose suitable matches.

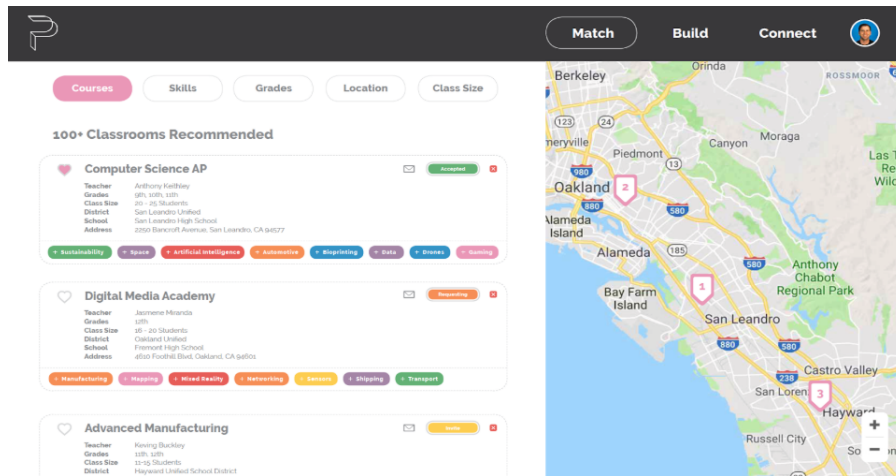


Figure 3.6 UI for the Recommender system

3.0.4 Evaluation of Recommender System

The recommender system, due to a miscommunication between ourselves and the local team, was not correctly incorporated into PilotCity's website at the time of project placement which occurred over our winter break. For this reason, PilotCity made the final matches as they had done in previous years. We then evaluated our recommender system against the final project-placements in the following way. For each employer's assigned classrooms, we recorded what rank our recommender system gave each classroom. For example, one employer, the City of Hayward, was assigned 7 classrooms for spring 2019. Our recommender gave these 7 classrooms the ranks shown in the "our ranks" row in table 3.1 below. We compare these ranks to what the "ideal ranks" would be (also shown table 3.1). Note that the lower the number, the more highly it is ranked (the classroom given a rank of 0 is the most highly-recommended classroom).

classroom	A	B	C	D	E	F	G
our ranks	13	16	14	15	30	91	29
ideal ranks	0	1	2	3	4	5	6

Table 3.1 Comparison of classroom ranks for City of Hayward

We can see from table 3.1 above that our recommender, though imperfect,

ranked the majority of the classrooms relatively high (13th, 14th, 15th, 16th place), yet some of the classrooms were ranked much deeper down the list, especially the outlier in 91st place. We create a score to encapsulate this trend.

Since all of the classrooms placed with the City of Hayward are equally suitable placements, any permutation of the ideal ranks is equally ideal. In order to account for this, our metric takes the median of our rank and compares that to the median of the ideal rank. We then subtract the median of the ideal rank from the median of our algorithm's rank to get a sense of how well, in aggregate, our recommender did for a given employer relative to that employer's ideal matching. For the above example with the City of Hayward, the score comes out to:

$$\text{median}(\text{our ranks}) - \text{median}(\text{ideal ranks}) = 16 - 3 = 13$$

Using the scoring method just described and averaging across all employers, our recommender system got a score of 34. Relative to over 150 total classrooms, this score is expected. Many of the variables that informed the final employer-classroom pairing decisions were factors we were advised to ignore to simplify the onboarding process, such as schedule and availability.

We considered tuning our recommender system's algorithm to minimize the score described above. However, we agree that at this stage with only 35 employers this approach would probably overfit our algorithm, especially as our score currently makes the strong assumption that each employer's "ideal" classrooms were the ones to which they were assigned. A more informative approach would be to gather data about how well classrooms and employers enjoyed their partnership and incorporate this into future recommender system iterations.

3.0.5 Impact

A primary goal of our project has been to improve scalability, automation, and user engagement of PilotCity's efforts. PilotCity having a website this year led to a major influx of new users. As PilotCity grows, they will be able to build off of our recommendation tool to optimize for the most successful classroom-employer partnerships. These improvements to PilotCity's programming may allow for more connections between high school students and workplace opportunities. These educational opportunities and general local connections benefit countless students and local cities which PilotCity serves.

3.0.6 Code

Our code is available on github.com/hmcmathclinic/18-19-PilotCity-Code and the most current version of the website can be viewed at pilotcity.com.

Chapter 4

Extracting Insights From Text Data

Topic modeling is an unsupervised learning algorithm that takes in a large corpus of text data and returns a specified number of representative topics found in the corpus. A topic is defined as probability distribution over fixed vocabulary. Words in a topic are sorted in descending order using the probability assigned to each word in the topic. The top k words in a topic (for some k defined by the user) reflect the overarching and related concepts of the topic. Topic modeling therefore provides a method of extracting insights about the high level meaning of a text. An example of an output of topic modeling is shown in figure 4.1.

Topic # 06	Topic # 07	Topic # 08	Topic # 09	Topic # 10	Topic # 11	Topic # 12	Topic # 13	Topic # 14	Topic # 15	Topic # 16
independent	fitness	math	theater	basketball	web	network	wine	swimming	painting	yoga
project	exercise	linear	musical	game	design	configure	winery	polo	color	relaxation
study	training	algebra	music	intercollegiate	site	cisco	grape	swim	drawing	breathing
end	strength	solve	production	team	create	routing	tasting	water	design	strength
noted	endurance	exponential	performance	competition	data	security	vineyard	backstroke	studio	flexibility
semester	aerobic	quadratic	ensemble	shooting	use	configuration	sensory	training	art	yo
develop	walking	rational	acting	participation	office	operating	production	stroke	critique	balance
form	kin	logarithmic	vocal	flag	lab	server	viticulture	butterfly	value	kin
instructor	muscular	intermediate	jazz	passing	page	lan	fermentation	kin	lighting	mat
lab	heart	learning	stage	football	user	wireless	world	competitive	composition	increase

Figure 4.1 Example distribution of topics output by a trained topic model

4.1 Methods

Our team considered and explored two well-known methods for topic modeling:

- Latent Dirichlet Allocation, commonly known as LDA, is a generative probabilistic model that prescribes each document in a corpus with a finite mixture of topics from an underlying topic distribution. LDA views each document in a corpus as a generated item from a collection in order to infer the topic distribution. The generative process that LDA uses to infer the underlying topic distribution represents the topic distribution θ_m for each document m as a random variable from a Dirichlet distribution with sparse priors, where each topic is a distribution over all of the words. For each of N words in document m , a topic $z_n \sim \text{Multinomial}(\theta)$ is chosen and a word w_n is chosen from a multinomial distribution conditioned on z_n . LDA requires the modeler to input number of topics. In our experiments, we built our LDA models from `LdaModel` in the `gensim` package.
- Non-negative Matrix Factorization (NMF for short) takes in a bag-of-words $n \times m$ matrix A whose entry A_{ij} is the number of occurrences of word j in document i . NMF seeks to find the closest approximation of A as a product of a $n \times k$ matrix W and a $k \times m$ matrix H with the condition that all entries of W and H are non-negative. In other words, NMF outputs W, H with the prescribed dimensions such that $\|A - WH\|_F$ is minimized.

To interpret the output matrices W and H , we call W the *basis matrix* and H the *coefficient matrix*. The prescribed number k here represents the number of topics we wish to extract from the text. To find out the top words are associated with the i -th topic, we examine the i -th row of the H matrix and take the words whose position in that row has a high value. In other words, the entry H_{ij} measures how relevant word j is for topic i . Next, if we want to find out what topics are the most prevalent in the i -th corpus document, we examine the i -th row of the W matrix and take the topics whose position in that row has a high value. Although there are methods to perform this task with unseen documents that were not used in training, we omit their discussions here as they were not used for our experiments.

In a later section, we discuss our preferred method and our reasons for

choosing it.

4.2 Assessing Insights

4.2.1 Topic Coherence

Topic coherence is a standard method of comparing topic models. It measures how semantically cohesive each topic is. For example, we'd say that a topic that contained all medically-related words ("doctor," "stethoscope," "surgery," etc.) has a high coherence while a topic composed of unrelated words ("city", "toothpaste", "kittens") has low coherence. Our measure of coherence is the average pairwise similarity between the GloVe vector representations of the top three words of a given topic.

By analyzing the mean topic coherence of a model compared to the number of topics in the model, we can determine the optimal number of topics for our model. As seen in figure 4.2, topic coherence peaks where the model only has two topics. However, the team judged this to be an under-fit of the model. In this case, the optimal number of topics chosen was 16, the next highest peak.

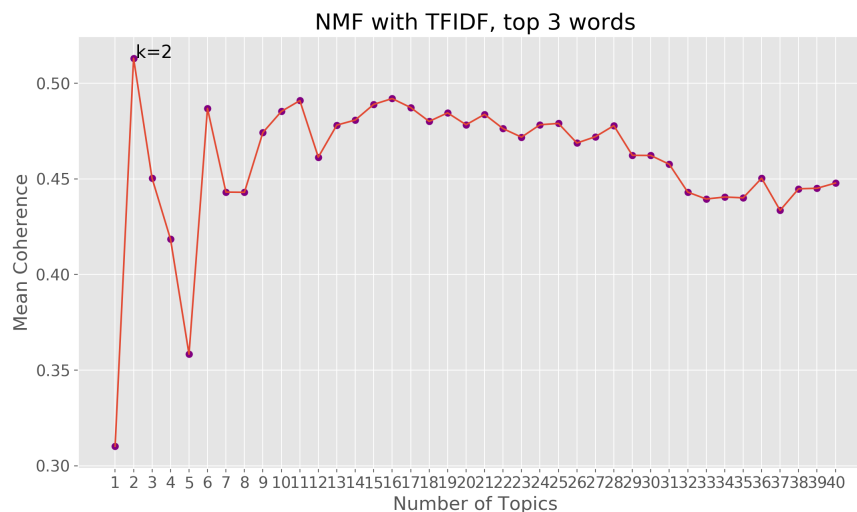


Figure 4.2 Topic Coherence vs Number of topics using the first 3 words of every topic

4.3 LDA vs NMF

We ran experiments to determine which topic modeling technique would perform well for our purposes, specifically on Los Positas Syllabi. We trained models using both techniques and graphed the mean topic coherence of the outputs from the models to judge which technique performed better. From the graph in figure 4.3, we see that for the range of number of topics considered, NMF consistently had higher mean topic coherence than LDA, and hence chose it for our model and the rest of our analysis. This was expected since NMF is said to perform better on sparser and smaller data sets (Chen, Zhang, Liu, Ye, & Lin, 2019). However, we did provide flexibility in the module so that the modeler could use an LDA model if they prefer.

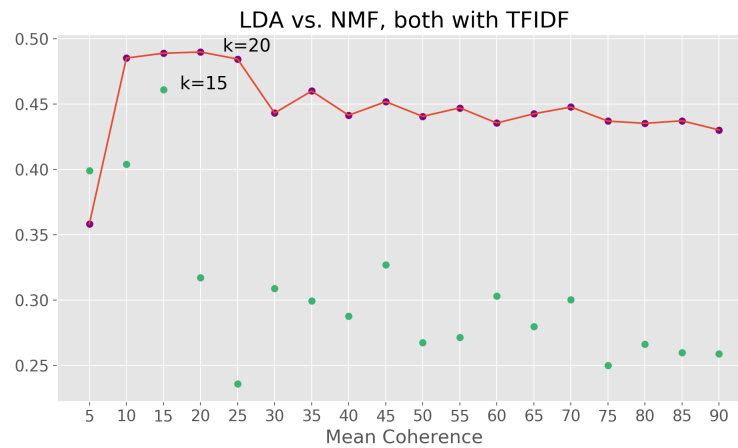


Figure 4.3 Topic Coherence of LDA compared to NMF. Models with NMF are shown with the line graph, and LDA are shown with dots

4.4 Extracting insights

Our goal is to leverage topic modeling to visualize high level themes represented in a corpus containing all the syllabi of an educational institution. By doing so, we can estimate the emphasis of each topic in the curriculum of the institution, as well as the overall educational priorities of the institution.

4.4.1 Data

Our dataset consists of 2,841 syllabi from Las Positas Community College in Livermore, California, which we obtained by scraping course syllabi publicly available on their website. http://www.curricunet.com/laspositas/search/course/course_search_result.cfm

4.4.2 Visualization of Results

The output of our trained NMF topic model is shown in figure 4.4. Each dot in this figure represents the 2D vector representation of a syllabus in the corpus. The color of these dots are determined by the topic they relate the most to. The top 3 words in each topic are printed alongside the syllabi-dots that are most related to it. Also, inter-dot distance is a measure of how similar the two syllabi are.

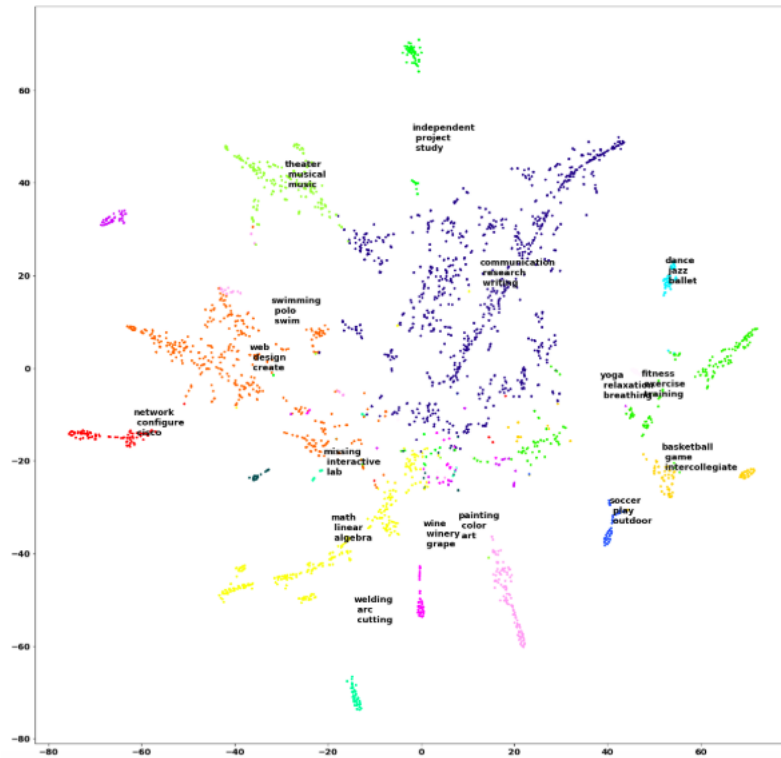


Figure 4.4 Visualization of NMF results of Las Positas data. Here we see that the topic related to communication, research and writing is prevalent across many syllabi in the corpus. These syllabi-dots are also spread out, implying that the syllabi related to this topic are similar to other syllabi in the corpus as well. In contrast, the topic related to dance, jazz, and ballet is less prevalent, and is dominant in syllabi that are only similar to other syllabi within the same topic.

4.4.3 PilotCity Deliverable

Our deliverable is a fully-functioning topic modeling package for PilotCity's use. The structure of our module is seen in figure 4.5.

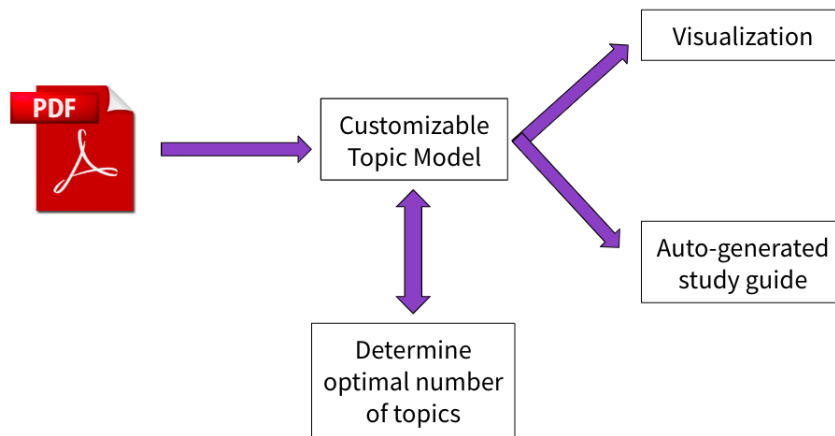


Figure 4.5 Pipeline for topic modeling module

The module takes a folder of PDFs as input and cleans and tokenizes the text in each document. It also filters out the most common words across all documents using an algorithm called 'term frequency inverse data frequency', or TFIDF. These cleaned documents are used to train an NMF topic model, where the optimal number of topics is programmatically determined based on topic coherence. The model is saved, and visualized as in figure 4.4. Given an unseen syllabus and a trained topic model, our module can also extract the most representative topics from the syllabus and search Wikipedia for word definitions to automatically generate a study guide prototype. An example of an auto-generated study-guide is seen in figure 4.6. We delivered the module to the PilotCity team and lead a training session to help them understand the potential use cases for the module during our spring site visit.



Study Guide

math: Mathematics (from Greek μάθημα *máthēma*, "knowledge, study, learning") includes the study of such topics as quantity, structure, space, and change. Mathematicians seek and use patterns to formulate new conjectures; they resolve the truth or falsity of conjectures by mathematical proof.

linear: Linearity is the property of a mathematical relationship or function which means that it can be graphically represented as a straight line.

algebra: Algebra (from Arabic "al-jabr", literally meaning "reunion of broken parts") is one of the broad parts of mathematics, together with number theory, geometry and analysis.

interactive: Across the many fields concerned with interactivity, including information science, computer science, human-computer interaction, communication, and industrial design, there is little agreement over the meaning of the term "interactivity", although all are related to interaction with computers and other machines with a user interface.

computer: A computer is a device that can be instructed to carry out sequences of arithmetic or logical operations automatically via computer programming.

electronics: Electronics comprises the physics, engineering, technology and applications that deal with the emission, flow and control of electrons in vacuum and matter.

writing: Writing is a medium of human communication that represents language and emotion with signs and symbols.

research: Research comprises "creative and systematic work undertaken to increase the stock of knowledge, including knowledge of humans, culture and society, and the use of this stock of knowledge to devise new applications." It is used to establish or confirm facts, reaffirm the results of previous work, solve new or existing problems, support theorems, or develop new theories.

reading: Reading is the complex cognitive process of decoding symbols to derive meaning.

dance: Dance is a performing art form consisting of purposefully selected sequences of human movement.

training: Training is teaching, or developing in oneself or others, any skills and knowledge that relate to specific useful competencies.

exercise: Exercise is any bodily activity that enhances or maintains physical fitness and overall health and wellness.

welding: Welding is a fabrication or sculptural process that joins materials, usually metals or thermoplastics, by using high heat to melt the parts together and allowing them to cool causing fusion.

Figure 4.6 Auto-generated study guide

4.5 Further work with Topic Modeling

In addition to running our topic model on syllabi of an institution to gauge current educational priorities, we also ran our topic model on syllabi of classes taught across many years to judge the change of an institution's educational priorities over time. This application is discussed further in the paper attached.

Chapter 5

Conclusion

We conclude the year with the following accomplishments. We successfully built:

- A launched website, created with the support of PilotCity's local team, which is already being used by PilotCity's academic and industry partners.
- A maintainable, scalable Google Cloud database with secure log-in and optimized data structures.
- A recommendation engine to match classrooms and employers for projects. This engine:
 - harnesses Natural Language Processing tools like the GloVe model to place similarity scores on user inputs such as course names, industry preferences and project descriptions.
 - outputs a ranked list of candidate classrooms/employers who are closely related. This list enables employers and teachers to easily discover each other for future partnerships.
- A module for visualization of educational priorities using topic modeling techniques. Features of this module include:
 - Training a topic model on any collection of PDF documents.
 - Flexibility in the type of topic modelling techniques used. Currently, the module supports LDA and NMF.
 - Automated selection of optimal number of topics based on topic coherence.

- Visualizations of the topic distributions of an input PDF document.
- An engine that generates a guide for students which enumerates and defines the most important keywords contained in a classroom syllabus.

Each stage of this project has been an iterative process involving multiple rounds of interviews and conversations with tech companies, school district administrators, high school teachers and other stakeholders to fully understand the impact of our work. From these interactions, our team has gained many insights about the educational system and the application of Natural Language Processing and automation to improve educational experiences.

Chapter 6

Future Work

6.0.1 Improvements to Recommender System

The current recommender algorithm requires that we specify weights to determine how much each input to the system contributes to a match result. This manual aspect of our algorithm could be removed by employing machine learning techniques to learn weights that lead to optimal matchings. This however would only be feasible after PilotCity has accumulated large enough employer-classroom matches and some measure of the success of the matchings.

6.0.2 Possible Extensions and Applications of Insights Engine

This report detailed how we investigated the potential to extract insights in the form of topic distributions from syllabi. This idea could be used by PilotCity to assist in the onboarding process where teachers have to supply keywords that describe the classrooms they teach. That is, PilotCity can have teachers upload course syllabus for a specific classroom and our trained insights engine would generate the top keywords that describe the nature of the class rather than having the teacher manually fill out this information.

One application of our insights engine that we explored in the report was the generation of a guide with definitions of overarching topics pertaining to a syllabus. We could improve these automatically generated guides to provide more contextualized information by performing network analysis on Wikipedia's clickstream data to gain insights on what pages people visit right before and right after going to the Wikipedia page pertaining to a topic.

References

- Chen, Y., Zhang, H., Liu, R., Ye, Z., & Lin, J. (2019, Jan). Experimental explorations on short text topic mining between LDA and NMF based Schemes. *Knowledge-Based Systems*, 163, 1–13.
- David M. Blei, A. Y. N., & Jordan, M. I. (2003). Latent dirichlet allocation. In *Journal of machine learning research*. Retrieved from <http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf>
- Greene, D. (2017). *Parameter Selection for NMF*. Available online at [gitHub.com](https://github.com).
- Hall, D. L., Jurafsky, D., & Manning, C. D. (2008). Studying the history of ideas using topic models. In *Empirical methods in natural language processing (emnlp)*. Retrieved from [pubs/hall-emnlp08.pdf](https://pubs.hall-emnlp08.pdf)
- Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. *International Conference on Machine Learning*, 1188–1196.
- McCormick, C. (2018). *Word2Vec Tutorial - The Skip-Gram Model*. Available online at mccormickml.com.
- Pedregosa, F. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12., 2825–2830.
- Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.
- Rehurek, R., & Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora.
- Sekiya, T., Matsuda, Y., & Yamaguchi, K. (2017, Oct). A web-based curriculum engineering tool for investigating syllabi in topic space of

standard computer science curricula. In *2017 IEEE Frontiers in Education Conference (FIE)* (p. 1-9). doi: 10.1109/FIE.2017.8190598

Shperber, G. (2017). *A Gentle Introduction to Doc2Vec - ScaleAbout - Medium*. Available online at medium.com.

Shuai, W. (2016). Topic Modeling and t-SNE Visualization. *Shuai's AI Data Blog*.

Institute of Education Science Website. (2019). Retrieved from <https://ies.ed.gov/aboutus/>

Las Positas College. (2019). *Las Positas College CurricUNET*. Available online at http://www.laspositascollege.edu/onlinelearning/faculty/distance_education/handbook/syllabus.php.

Xu, J. (2018). *Topic modeling with LSA, PSLA, LDA Lda2Vec - NanoNets - medium*. Available online at medium.com.