



**Mathematics Clinic**

Midyear Report for  
*PilotCity*

## Automating an Engine to Extract Educational Priorities for Worforce City Innovation

December 14, 2018

### **Team Members**

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

### **Advisor**

Dr. Talithia Williams

### **Liaison**

Derick Lee



# Abstract

Our clinic team is tasked with developing software and algorithms to automate PilotCity programming and to also extract educational insights from less well-structured data sources like websites, syllabi, resumes, and more. Our solutions involve creating web interfaces that teachers, students and employers involved in the PilotCity Program can engage with, designing recommender systems that facilitate the process of matching employers to high school classrooms and employing Natural Language Processing (NLP) techniques to extract teacher skills and interests as well as employer priorities. In this report, we detail our approaches, implementation, results, future directions, and the impact of our work.



# Contents

<b>Abstract</b>	<b>i</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Executive Summary</b>	<b>3</b>
<b>3 Initial Work</b>	<b>5</b>
3.0.1 Background . . . . .	5
3.0.2 WebDev . . . . .	5
3.0.3 Recommender System . . . . .	7
3.0.4 Impact . . . . .	7
3.0.5 Code . . . . .	7
<b>4 Conclusion</b>	<b>9</b>
<b>5 Future Work</b>	<b>11</b>
5.0.1 Improving the GloVe Model . . . . .	11
5.0.2 Topic Modeling . . . . .	13
<b>Bibliography</b>	<b>17</b>



# Chapter 1

## Introduction

PilotCity 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. PilotCity is connecting local schools and employers and promoting the idea of project based learning by empowering students from an early age, as the first step to building an innovation ecosystem from scratch.

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 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. Our project is unique because we have two independent stakeholders (IES and PilotCity) with different but connected areas of interest. Thus, 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. Our project therefore involves incorporating NLP techniques like text mining and topic modelling in our PilotCity solution that enable us to extract and make inferences on educational priorities and approaches.

In Chapter 2, we give an executive summary for those who desire a

## 2 Introduction

---

quick overview of this clinic project. In Chapter 3, we document the web development process that dominated the first semester of clinic. Specifically, we will discuss our approaches, our findings, our implementations, and the societal import of our work. We provide a conclusion in Chapter 4, and finally, in Chapter 5, we outline several paths of works for next semester.



## Chapter 2

# Executive Summary

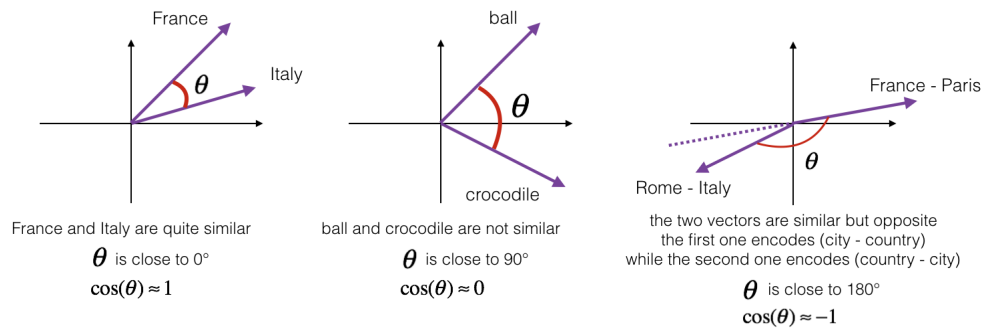
The goal of this project is to optimize automation, scalability, and user engagement of PilotCity programming. Specifically, we want to automate a process which has historically been done by hand. As PilotCity grows, matching employers and classrooms by hand will be intractable. Our work will enable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

The first step in automating was to create an onboarding website, where employers and teachers could log in and input all their relevant details. These details were then used to generate a list of recommendations for them. These recommendations would be modifiable based on certain filters. This website will eventually also be a platform for students, teachers, and employers to engage with each other, and look at their milestones.

Working off of designs provided by Derick, our PilotCity liaison, 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 some more user interviews, with both teachers and employers, and gauged 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. As demonstrated in Figure 2.1, we use the angle between the two vectors to represent the similarity between the two words.



**Figure 2.1**

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.

This algorithm is ready to be incorporated into the website so that an employer logged in is able to see a ranked list of all the classrooms based 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 their similarity score as determined by the GloVe model.

Our project also focuses on increasing the user engagement of PilotCity programming, which correlates with reducing the burden of sign-up on employers and teachers. To further this goal, our next step will be to "auto-fill" certain fields. For example, in a regular sign up procedure, when an employer signs up on the PilotCity website, they input their company name, and then their company's physical address. However, our plan is to web-scrape and obtain the address from the company's website and "auto-fill" it in the questionnaire. Web-scraping and keyword extraction can also be extended to employers' resumes and teachers' course syllabi to further assist and enhance the recommendation system. This future work is further described in Chapter 5.

## Chapter 3

# Initial Work

### 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, and more standard technical problems, including 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 uncured 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 WebDev

Our initial technical work involved creating the frontend onboarding platform to start building the web application. We created a sequence of pages in which teachers and employers can fill in relevant information that allows us to make the most optimal recommendations. Once we had a functional frontend product, we handed off the frontend work to PilotCity's local team.

## 6 Initial Work




Logout

## Who are you?

Teacher

Employer

Student



Logout

### What's your story?


My name is

and I am a teacher at  of the

My phone number is

Back

Save and Next




Logout

Period	Course Name	Semester	Grade	Class Size
<input type="text" value="P1"/>	<input type="text" value="AP Physics"/>	<input type="text" value="Spring"/>	<input type="text" value="Selected at 12:30"/>	<input type="text" value="15-20 Students"/>
<input type="text" value="P2"/>	<input type="text" value="Enter Course Name"/>	<input type="text" value=""/>	<input type="text" value="Select"/>	<input type="text" value="Select"/>

Back

Save and Next



Logout

#### Industries

What industries would your classrooms be excited about?

Tech

Healthcare

Arts

Robotics

Data Science

Internet Of Things

Sustainability

Space

Artificial Intelligence

Biotechnology

Gaming

Auto

Energy


Design

Healthcare

Lifestyle

Back

Save and Next



Logout

#### Skills

What tools, technologies, and skills are you currently teaching?

Cloud Networking

Python

3D Printing

Robotics

Artificial Intelligence

Augmented Reality


Data Science

Security

Back

Save and Finish

Thank you



We will notify you when your match results are available.

Over and out

**Figure 3.1** The onboarding flow for a teacher participating in PilotCity

### **3.0.3 Recommender System**

We then 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. We use the GloVe Model to measure the semantic similarity of words. We have implemented our recommender system and it is currently being productionized in the matchmaking platform.

### **3.0.4 Impact**

Our project will improve scalability, automation, and user engagement of PilotCity's efforts. 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.

### **3.0.5 Code**

Our code is viewable on our github ([click here](#)) and the most current version of the website can be viewed at [pilotcity.com](http://pilotcity.com).



## Chapter 4

# Conclusion

As described above, this project involves a combination of web development, construction of a flexible recommender system, and the use of topic modeling to extract information about a user's experience and course descriptions.

We conclude the semester with the following accomplishments:

- A launched website, created with the support of PilotCity's local team, which is already being widely used by PilotCity's academic and industry partners.
- A maintainable, scalable Google Cloud database with secure log-in and optimized data structures.
- An online-updatable and parallelized recommendation engine to match classrooms and employers for projects. Features of the recommendation engine include:
  - The engine outputs a ranked list of "top candidates" for the user to browse. From this candidate list, employers are free to request and accept collaboration opportunities with classrooms (and vice-versa).
  - Natural Language Processing tools, including the GloVe model, are harnessed to score uncurated user-inputs such as course names, industry preferences, and project descriptions.
  - We parallelized the algorithm to reduce its runtime.

The power, user-friendliness, and flexibility our product provides will completely shift PilotCity's paradigm, and the best part is that we

will get the chance to test-run it during PilotCity programming next semester.

The website design and implementation has been an iterative process involving multiple rounds of user interviews. Together, all five of us have learned HTML, CSS, and JavaScript for the first time which has helped design brand new PilotCity web pages with our liaison, Derick Lee and translate those designs into workable, open-source code. At each step, we have interviewed and presented our product to tech companies, school district administrators, high school teachers, PilotCity student fellows, and other stakeholders.

From these interviews, we have gained insight about how the educational system works and how automation and natural language processing can greatly impact it. Often, time required to enter boxes in a form bars participation in work-based learning programs for teachers, employers, and students alike. For this reason, leveraging topic modeling on uncured data sources like resumes, syllabi, and employer/teacher websites has increasingly become an area of interest. The next challenge will be to ensure good recommendations, despite being given even less consistently informative data through uncured sources.

To that end, we look forward to investigating topic modeling for automatic recommendation feature extraction on web-scraped or PDF documents to match classrooms and employers. To score results, we'll look into modifying the GloVe model, introducing different models (like word2vec), or using a combination of models. We will also try applying topic modeling and other Natural Language Processing tools to extract information from student project documents uploaded to PilotCity's website, possibly building another recommendation system whose goal is to match students with employers for internship opportunities.



## Chapter 5

# Future Work

Next semester, the team will dive deeper into optimizing our natural language processing deliverables. Specifically, our next steps consist of the following:

1. Train a specific language model that will be used in PilotCity's match-making system.
2. Incorporate topic modeling in PilotCity's recommendation system as well as to generate features for language models.
3. Implement our algorithms and models on PilotCity's platform.

### 5.0.1 Improving the GloVe Model

As mentioned in the previous section, we are currently using the GloVe model to measure similarity between responses in teacher and employer surveys. However, there are several issues with our current implementation:

1. The model is too general in that certain words that we want to be semantically close correspond to dissimilar vectors.
2. The model does not deal with compound words well.
3. The model is outdated as it has been pre-trained on the Wikipedia corpus in 2014 and therefore does not reflect current trends.
4. The model is not adaptable in the sense that it does not update upon user input.

The team's solution to these problems is to train a language model from scratch so that it resolves all the problems listed above. Specifically, the model will be trained with data more relevant to our purpose: matching employers with teachers. This means the training data will likely include scraped employer websites, teacher course syllabi, resumes, and some baseline data (relevant Wikipedia pages). In addition, we will implement the model in such a way that it is online updatable and does not require training the entire corpus again for updating. Finally, to take care of the compound words, we can modify the algorithm to keep track of consecutive compound nouns as opposed to individual words.

Based on these constraints on the model, there are a couple language models to consider: GloVe, word2vec, doc2vec.

### GloVe

To obtain online updating, we first need to store the co-occurrence matrix  $X$  –  $X_{ij}$  is the number of times word  $i$  occurred in the same context as word  $j$ . Note that the GloVe model is based on minimizing the following cost function

$$J = \sum_{i,j} f(X_{ij})(w_i^T w_j - \log X_{ij})^2$$

where  $f(x)$  is a function with desirable properties that is not important here and  $w_i$  is the word vector for word  $i$ . Notice that this function is convex, so  $w_i$  can be updated using gradient descent. The main computational bottleneck is the storage of the matrix  $X$ , which is a  $N \times N$  matrix, where  $N$  is the size of the vocabulary list.

To keep track of compound words, we plan on incorporating a compound word detector, possibly using existing language models, and including the compound as a word in the matrix  $X$ . This way, we obtain co-occurrences of compounds words and other words.

All in all, the main benefit of GloVe is that it is one of the most accurate and reliable word embedding models. The main draw back is that it takes a long time to update vectors as well as a lot of space to store the infrastructure needed for online-updating.

### word2vec

Word2vec is based on a single hidden layer neural network where the weights from the input layer to the hidden layer represent the word vectors.

For example, say the input layer has 10000 neurons – this corresponds to 10000 words in the vocabulary – and the hidden layer has 300 neurons – this corresponds to our word vector having 300 components, then the 300 weights between input neuron  $i$  and the hidden layer represents the word vector for word  $i$ . The output layer, which has the same number of neurons as the input layer, is computed through a softmax function so that it outputs a probability depending on the input. One immediate advantage that word2vec offers over GloVe is storage, since it only requires storing  $2N \times d$  values where  $N$  is same as before and  $d$  is the desired dimension of the word vector.

In addition to space advantages, a byproduct of the neural network is that it keeps track of words that frequently appear next to each other, namely compound words. The model is intrinsically online updatable because the network is trained by feeding in pairs of words from a new document, which is easy to implement.

### 5.0.2 Topic Modeling

We envision that topic modeling and various other natural language tools will help in the matchmaking process as well as other functions on the website. For example, when the site enables students uploading text document for their projects, we could utilize topic modeling to tag each document with associated keywords.

Below we will compare and modify as necessary a series of topic modeling approaches, a handful of which are described below.

#### Latent Semantic Analysis

LSA is the simplest topic modeling scheme, so we plan to start with LSA as a baseline. This method simply performs Single Value Decomposition (SVD) on the TF-IDF matrix of all the documents. If we suspect there are  $k$  topics in the document corpus, then taking the factorization for the  $k$  largest singular values:

$$A \approx U_k S_k V_k^T$$

the rows of  $U_k$  represent the document vectors in terms of  $k$  topics and the rows of  $V_k$  represent the word vectors expressed in terms of topics.

Although LSA is efficient to use, its main drawback is that it lacks interpretable embeddings, i.e. we do not know what the topics are.

**Probabilistic Latent Semantic Analysis (pLSA)**

This approach is a more generalizable, probabilistic version of LSA. The core idea is to produce a probabilistic model with latent topics that is capable of generating the data observed in the TF-IDF matrix.

**Latent Dirichlet Allocation (LDA)**

The classic approach for topic modeling is to use LDA, the Bayesian version of pLSA. Its main benefit is its performance over LSA. Unlike LSA, the results from LDA are interpretable. For example, after specifying a certain number of topics, LDA outputs a list of topics where a topic is represented by a distribution of words, as shown in Figure 5.1. We can then infer from the output that the topics are respectively about games, space, and hardware.

Topic 1		Topic 2		Topic 3	
term	weight	term	weight	term	weight
game	0.014	space	0.021	drive	0.021
team	0.011	nasa	0.006	card	0.015
hockey	0.009	earth	0.006	system	0.013
play	0.008	henry	0.005	scsi	0.012
games	0.007	launch	0.004	hard	0.011

**Figure 5.1** Example of a possible output for LDA with 3 specified topics.

However, LDA is not perfect as it does not always yield the most usable nor informative topic clusters (Nijessen, 2017).

**LDA2vec**

This approach is an extension of word2vec and LDA that jointly learns word, document and topic vectors. In some ways, LDA2vec is more powerful than ordinary LDA because it uses word vectors and can detect semantic similarity of words. On the other hand, this method is still quite new and has limited capabilities in terms of implementation.

**doc2vec**

Rather than using LDA at all, an alternative is to simply employ doc2vec. Doc2vec (Mikilov and Le, 2014) is very similar to word2vec, except that rather than convert words into vectors, it converts entire documents into vectors. Doc2vec has been shown to demonstrate higher accuracy than word2vec on certain tasks, is faster, and requires even less storage than word2vec (Schberber, 2017). The only concern is that, given too large or varied a document, doc2vec could overgeneralize the content. However, paragraphs can be treated as documents, so this could allow for more fine-grained keyword extraction.

In fact, by using word2vec and doc2vec together, we can extract vectors for documents (or paragraphs of documents) which contain information about the semantic meanings of words. We can try clustering the document vectors (using k-means, k-medoids, or another approach), taking the cluster center (a vector), and using word2vec to find keywords similar to those centers.



# Bibliography

Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. *International Conference on Machine Learning*, pages 1188–1196, 2014.

Chris McCormick. Word2vec tutorial - the skip-gram model. Available online at [mccormickml.com](http://mccormickml.com), 2018.

Richard Socher Pennington, Jeffrey and Christopher Manning. Glove: Global vectors for word representation. *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

Gidi Shperber. A gentle introduction to doc2vec - scaleabout - medium. Available online at [medium.com](http://medium.com), 2017.

Joyce Xu. Topic modeling with LSA, PSLA, LDA Lda2Vec - NanoNets - medium. Available online at [medium.com](http://medium.com), 2018.