AIpiphany Final Report

Client: Dr. Michelle Archuleta
Advisor: Dr. Wendy Fisher
Team Members: Morgan Cox, James Simonson, Adam Stogsdill, Connor Stutsman-Plunkett
9th June 2020

Contents

Introduction

Client Background

AIpiphany is an artificial intelligence focused company with the vision to transform complex technical language into layman's terms, building trust and transparency between doctors and patients. AIpiphany Notes, the primary product of AIpiphany, is an artificial intelligence (AI) solution that takes the complex jargon of electronic health records (EHRs) and makes them into plain, actionable language. Since AIpiphany Notes uses AI to empower patients and increase understanding of medical terminology, AI was put at the forefront of the company's name.

Project Outline

AIpiphany's primary objective for summer field session 2020 is to create a program capable of correctly replacing medical acronyms and abbreviations with their appropriate long form, thus helping give patients unambiguous terminology and a better understanding of their required actions; this project was given a goal of 80% accuracy. To do this, the project team created a machine learning model that can perform the necessary tasks of searching through a vast assortment of expanded, long forms of an acronym. Then, the model can find the long form that most accurately represents the contents of what would otherwise be a confusing acronym or term. Once this is done, the corresponding long form can then be further simplified by existing AIpiphany algorithms before being given to the patient, empowering the patent to follow doctors' orders, thus improving communication between medical professionals and the public.

AIpiphany Notes utilizes natural language processing (NLP), machine learning (ML), and data science in conjunction together to gather appropriate data, create an ML model, and finally construct a program that brings these different aspects of the project together in one deliverable. Data science is an integral part of the project, primarily due to the data-hungry ML models that often need several gigabytes of data just to train effectively. NLP is also a key component of this project as understanding natural human language is nowhere near as simple for a computer as it might be for an average person, as meaning created by things like context could potentially be lost in the eyes of the computer. The beating heart of the project is undoubtedly machine learning, with ML providing the brains of the algorithm that replaces what would have been a confusing acronym with an appropriate expanded form.

Requirements

Functional Requirements

The overall functional requirement tasked to the team was to create a usable tool that AIpiphany can use as an integrated component of their entire system to help better patient-doctor relations. The tool must be able to accept textual data composed of an arbitrary number of English sentences that pertain to the medical field; a good example of this would be a doctor's note for a patient. The model will identify the ambiguous medical acronyms within the sentences and, if there are any medical abbreviations present, replace that piece of shorthand with the correct corresponding long form that retains the same meaning. The model will finally return the altered text, with non-ambiguous terms.

The most important functional requirement the team was tasked with was for the model to have an accuracy of at least 80%. Accuracy, especially when working within the medical field, is a critical component of a successful product; because the medical field is so complex, this can be a difficult task. When trying to accomplish this task, the team had to consider the possibility of overfitting or underfitting the final model. If the model is underfit, it is much more difficult to obtain the 80% accuracy requirement and may produce data that is not extremely useful to the client. On the other hand, if the model is overfit, then it cannot be effectively used on new unseen data because it relies too heavily on the input data it had received. When the team was working towards achieving the goal of an 80% accuracy it was important to do this carefully in order not to over or underfit the model as this could create false results for the client in the future causing preventable harm.

Another functional requirement for the team was to create testing and training datasets. Testing and training datasets are vital components for data science; these datasets help train and test the model, providing the team with the accuracy of the model and a good idea of its overall performance. The training and testing datasets are split into comma separated values (.csv) files that are formatted in two distinct ways. One configuration has columns that contain the following information: ‘Short Form,’ ‘Long Form,’ and ‘Sentence,’ while the other is formatted as ‘Long Form’, ‘Sentence’. The project team used an 80/20 split on the data, providing the training set with 80% of the data to help build a good model and the remaining 20% of the data to test the model’s abilities.

In order for the model to be used by anyone who wants to disambiguate a medical text or document, the team was asked to provide a way to take in input, run it through the model, and produce the final result with the disambiguated medical text or document. This would provide the user the capability to easily utilize the team’s model, making it simple for the client to implement it into their already existing features.

Non-Functional Requirements

Many of the non-functional requirements that guided the development of this project come from the context of the project itself. The project required the use of artificial intelligence, machine learning models, and the data science workflow. Since the product will be used within the medical field, this added some complexity that may not be present in other field session projects. This complexity was mainly a result of potential issues with personal information and the consideration of model accuracy. Additionally, the team had to consider which model to implement, dealing with web scraping legality, and creating a clean dataset.

Deciding which model to develop for the field session was the first step for the team and needed some careful considerations, such as training time and storage space. For a month-long project, these concerns were especially important, as the room for error was much smaller than a typical project that could be worked on for multiple months. The team looked at two different machine learning frameworks that could solve the task at hand. The first was BioBERT [1], which is a biomedical representation model that is used for natural language processing in medical fields. The other model was Word2Vec [2], a method to develop word embeddings, which the team decided on due to its ease of implementation, simplicity in understanding, and size as this allowed it to be trained much faster.

Data collection was also a critical non-functional requirement of the project. For the team to create a robust model, members had to gather so much data that the project could likely be classified as a “big data” project. Data collection involved web scraping, manual data retrieval, and parsing. The team collected data from PubMed [5] and Merck Manual [6], two publicly accessible, and reputable, online data sources. Data collected from PubMed and Merck Manual was used to develop the final dataset, which was used to train the machine learning model. The dataset created from web scraping is on the scale of gigabytes of text and took a large amount of time to gather and parse. When scraping these websites, privacy was not a concern the project team had to account for, as these sites are publicly available, and the data being scraped did not include any sensitive information. Recently, scraping such as this was ruled to be permissible in the court case, hiQ *Labs v. LinkedIn* [7].

Privacy was the team's final non-functional requirement, and while privacy was not a concern when developing the web scraping tools, there were some concerns with doctor-patient confidentiality when using doctor’s notes. The team confirmed that protected health information (PHI) and personally identifiable information (PII) are already nullified in the data that the project team used and, in the data, put into the final model. The client assured the team that this would be taken care of on their end, which allowed the team to largely disregard this concern during field session.

System Architecture

The project has many different components that must run in a specific order to create meaningful results. This started with code to scrape primary data sources such as HTML and XML files. It continues with various steps to strip the data of problematic text and identify the sentences that are contained within. Next, the cleaned data was split into sentences and paired with relevant data, including the unambiguous long form contained in the sentence and its corresponding short form, thus creating usable training data. From there, the data was split into training and testing data. The training data was used to train the model, and the testing data was subsequently used to determine the success of the model training. Once the model is trained properly, most of the preprocessing code is no longer needed to disambiguate acronyms, as the model is now independent of any training data.

Data Science Workflow

The project followed the data science workflow as a base level of system architecture, which highlights different and important stages of the entire data science process. As seen in Figure 1 below, the data science workflow begins with data acquisition. Data gathering was completed by scraping content from PubMed and Merck Manual, as mentioned in the *Non-Functional Requirements* section. Without this initial step in the data science workflow, it would be impossible to continue the rest of the project, because data science is built on data.

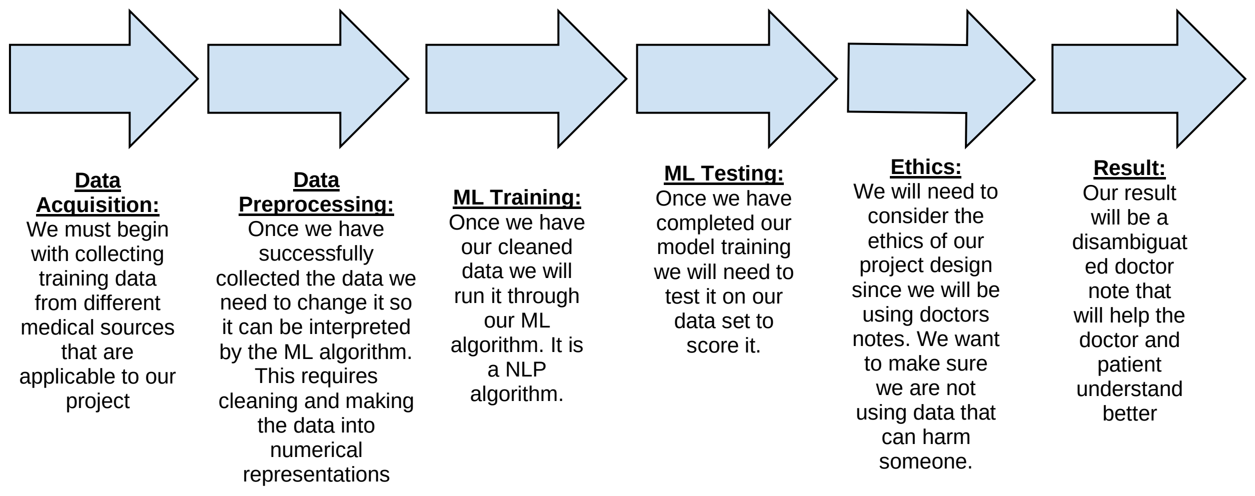

Data Acquisition: We must begin with collecting training data from different medical sources that are applicable to our project
Data Preprocessing: Once we have successfully collected the data we need to change it so it can be interpreted by the ML algorithm. This requires cleaning and making the data into numerical representations
ML Training: Once we have our cleaned data we will run it through our ML algorithm. It is a NLP algorithm.
ML Testing: Once we have completed our model training we will need to test it on our data set to score it.
Ethics: We will need to consider the ethics of our project design since we will be using doctors notes. We want to make sure we are not using data that can harm someone.
Result: Our result will be a disambiguated doctor note that will help the doctor and patient understand better

Figure 1: Data science workflow

After the team collected, preprocessed, cleaned, and filtered the data to make sure no unnecessary or dirty data is used in the model. Preprocessing the collected data is useful because it removes extra characters such as numbers, slashes, and other problematic characters in the data. This also helped to make sure the data was quality, as discussed in the *Data Quality Assurance* section. The use of bad data can throw off the entire model and produce poor results that might not be discovered until much later, which could negatively affect the models final accuracy and could make hitting the target of 80% accuracy almost impossible.

Once the preprocessing was completed, the model needed to be run on the training data, producing a working model that can be used by the client. This is step three of Figure 1 above, and this step can take quite a long time with datasets as large as the one the team produced. As noted previously, the team decided on doing an 80-20 percentage split of the total data used for training and testing, respectively.

Before the results from the machine learning algorithm can be used and put into place it is important to consider the potential ethical concerns that the project might deal with. The team had to consider the ethics of the data collected, sources, private information, use cases, etc. and the ethics of the final model that is to be used by the client. These are discussed in further detail under the *Ethical Considerations and Concerns* component of this document.

Finally, once the model has been trained and tested on the dataset, the model results are ready to be used, but ethical considerations must be considered. It is important to make sure the data used by the model does not have biases or, at least the model biases are noted and made clear to end-users.

Final Product

When the project is completed, the team will be providing the client with a python module. The final module will be used in the client's current architecture. Once the module is implemented, it will have the ability to be used by just running the teams imported module. The model and any lookup tables will also be shipped along-side the code to provide the client with a base usable product. With the tools provided the model can be iterated on more. If the final model needs to be retrained, the code provided can collect more data, clean, and retrain the model so it can be updated.

Technical Design

Model Description

The team decided on the Word2Vec model [2] after determining that BioBERT presented challenges that could not be reasonably dealt with before the end of field session. The Word2Vec machine learning algorithm starts by encoding all the words into vectors. For example, if one were to encode the sentence “The doctor said to use insulin,” the resulting encoded vector would be:

$$v = \{'the', 'doctor', 'said', 'to', 'use', 'insulin'\}.$$

Once the Word2Vec model encodes the words, it will have a corresponding vector that tells us its position in the sentence and zeros where it is not, for example, using the Word2Vec with the sentence above would return the word vectors as follows:

$$the = [1,0,0,0,0,0]\,, doctor = [0,1,0,0,0,0].$$

Once the project team embedded the words, the model uses Word Movers Distance (WMD) [3] to calculate the similarity of the context of the words and sentences. WMD uses the idea that the distance between words can help drive their semantic meanings and calculates the minimum distance between word vectors. The project team elected to use WMD because of its ease of implementation within the Word2Vec model. This choice gave the team the ability to implement it during the limited time provided by the field session and still produce quality results for the client.

Figure 2 below shows an overview of how the entire process works using the Word2Vec model. On the left side of Figure 2, it shows the words of the sentence and the representation of them as vectors. Word2Vec returns a vector representation of the word for any word given to the model. Using the Word2Vec model and cosine similarity, WMD calculates the similarity of the contextual information between two sentences.

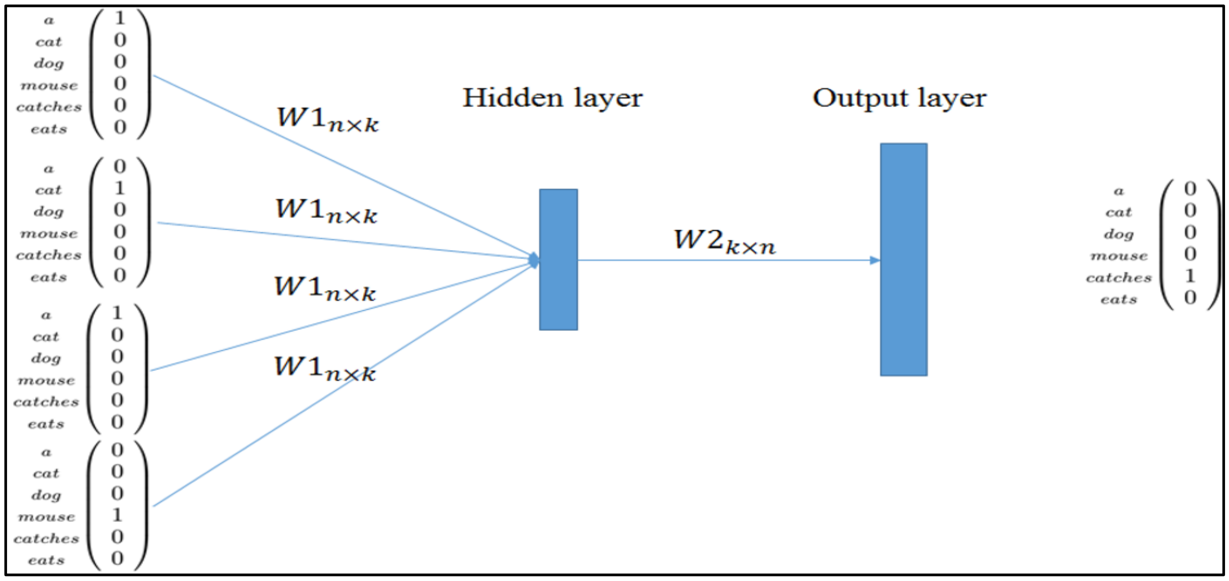


Figure 2: Word2Vec visualization [4]

Figure 3 depicts a visualization of the WMD algorithm. Words such as “chat” and “meet” are tied together because their overall contextual differences are quite small, allowing them to be grouped into the same idea. This is how the team’s model finds similar words in different sentences or contexts and can help identify and categorize them, helping the program determine the correct long forms within the different sentences.

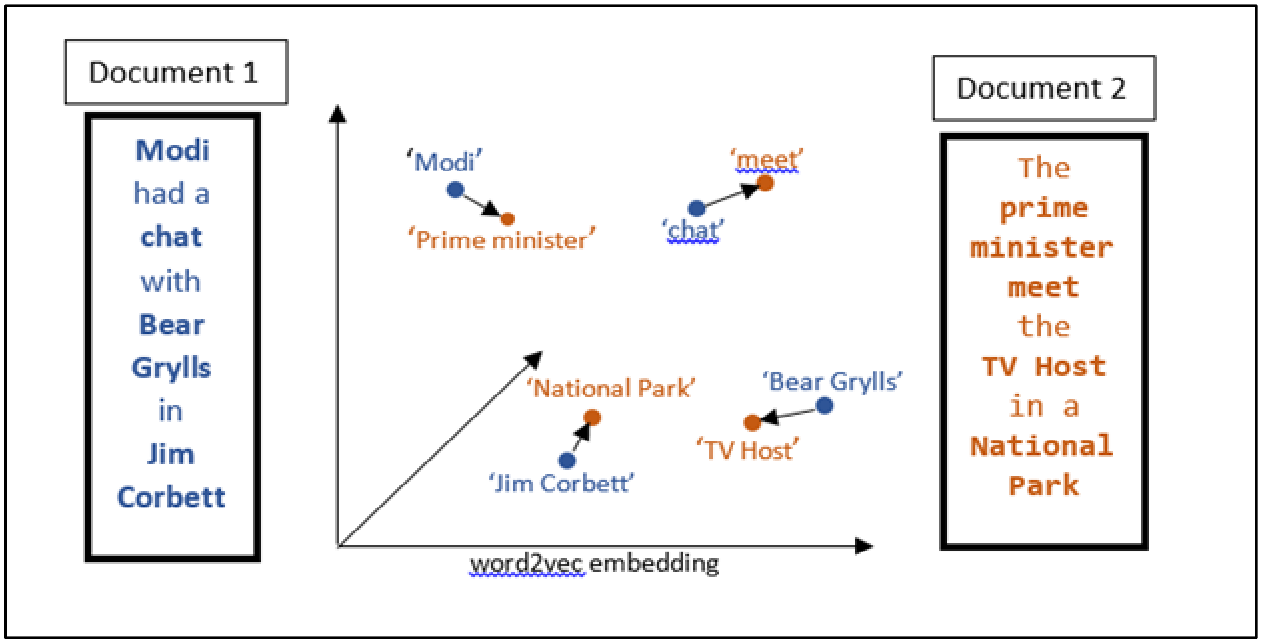


Figure 3: Word Movers Distance visualization

Design Optimizations

Initially when trying to predict the long form of the abbreviation, there would have to be a search through the whole dataset. This required the loading of 21 comma separated values (.csv) files, each approximately 3 GB in size, and caused immense stress on the system. To optimize long form predictions, the dataset files the model used to reference sentences were split into files where each file contains an individual long form and all sentences that contained that specific long form, which helped to decrease loading times and also provided simple checks for the availability of the data. Each resulting file is significantly smaller than the original size of 3 GB, allowing for faster querying of the data and other operations that might potentially be of interest in the future, thus increasing the usability of the algorithm. The project team referred to this in the code as “chunking” due to the separation of data into more manageable chunks for the program to handle.

Additionally, the prediction loop took a long time to run and the skew of data for specific words caused large considerations to be taken when making an abbreviation replacement. To remedy this, there needed to be an approach that lowered the amount of cases and optimized accuracy. At first, using the average number of samples seemed like a great way to perform this. However, there was a noticeable dip in accuracy. A method of using the median number of samples would be another way to solve this. A median account for skew and it is generally a better metric for this scenario. However, the time would sometimes be a bit longer than it could have been, and the accuracy was still low, approximately 63%. Finally, an approach limiting the number of samples for every other word to the same number of samples of the word with the least number. This yielded a large increase in time and accuracy in the model and remains the primary method of under sampling.

Quality Assurance

Strategies for Quality Assurance

As this project is centered primarily around the gathering and processing of data, quality assurance (QA) was a critical, yet slightly unorthodox component. In order to do high quality work and produce a final product that meets all requirements, the project team used some of the guiding principles laid out by the Agile framework and implemented a Scrum development process, both of which established positive habits and goals that everyone was able to agree on. Quality assurance for a ML model or data science project, such as this project, was difficult to achieve if the team strictly followed strategies laid out for typical application, so the team had to adopt different methods to ensure quality was a priority throughout the development process.

Data Quality Assurance

The team’s first steps in quality assurance were to make sure the data collected was properly formatted and passed through rigorous cleaning. By cleaning the collected data, the team made sure the Word2Vec machine learning algorithm was getting data in an interpretable format. An interpretable format means the team had to remove extra characters such as numbers, slashes, etc. to save time and improve the accuracy of the team’s model.

Another way the team made sure the data was of high quality was to run it through a validation script. The validation script was used to take in the cleaned data that only contains sentences with acronyms. The sentence is run through the validation script to make sure the correct long or short form is identified within the sentence. This validation just made the dataset stronger and provided a double check helping to ensure the quality of the data used in the final Word2Vec model.

Finally, before the data was sent into the team's machine learning model, the team computed data metrics. The data metrics provided an insight into what is being fed into the model. The team looked at the distribution of words across the data to see if there were any words that may have more occurrences than others. As Figure 4 depicts below, the data has a few words that have many occurrences while the average is much lower. This is due to the acronyms being something like ‘by’ or ‘a’ which are used in both normal language and as acronyms. To fix this, the team provided the client with notes and instructions for several methods to improve such as using Wikipedia to find more occurrences of the acronyms that are needed.

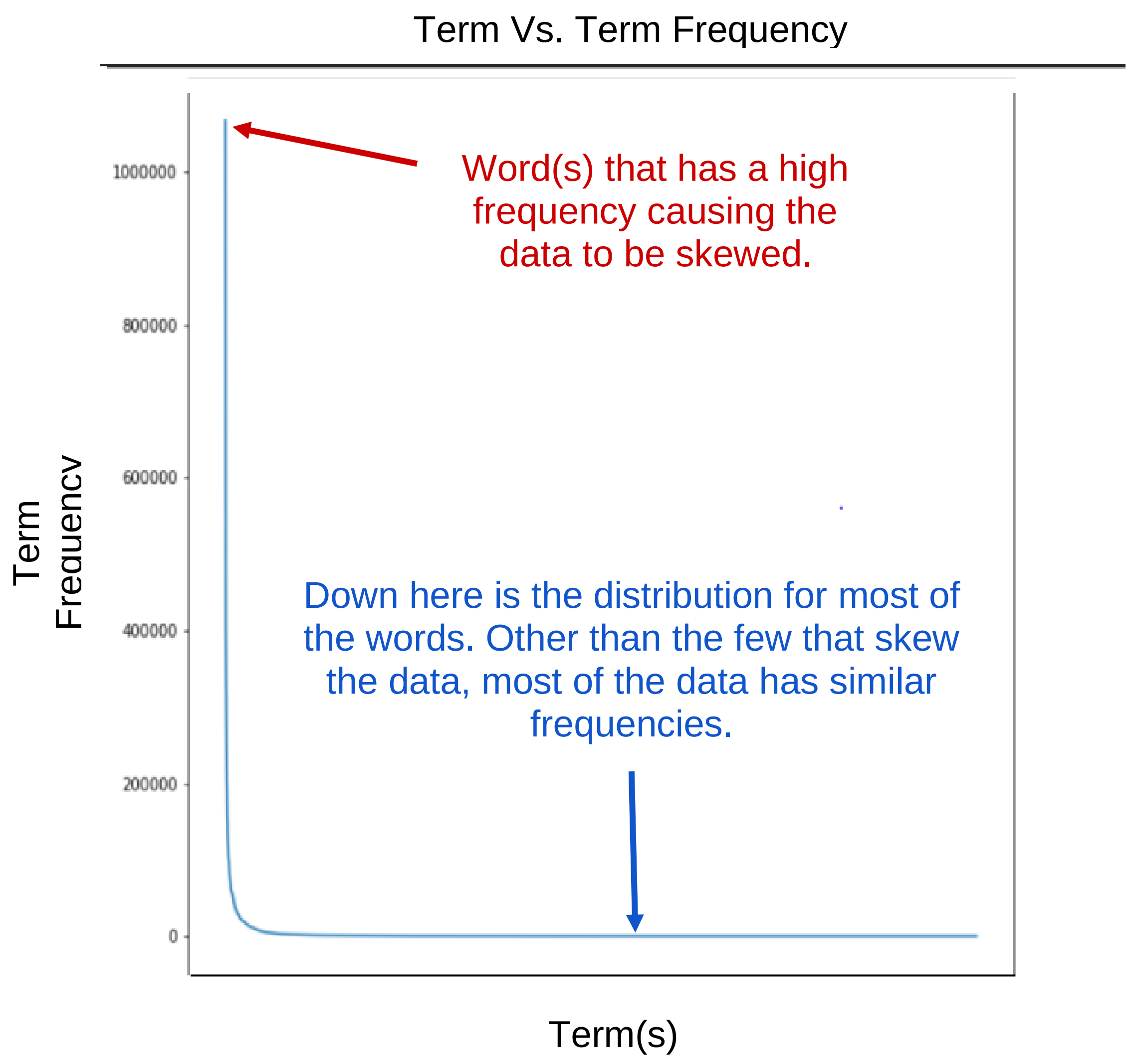


Figure 4: Distribution of terms and their term frequency, showing a skew in distribution

Machine Learning Model Quality Assurance

Once the model completed running on the training dataset, it had to be scored using the testing dataset. As mentioned earlier, the team split the data collected into 80% training and 20% testing data, as opposed to the typical 70% training and 30% used in the data science workflow. Once the model ran on the training data, the team then ran it on the remaining data using a validation script to properly score the model. By validating the model using training and testing data, it provided insight into the model's performance regarding its runtime and overall accuracy. With the overall model performance and scoring completed, the team compared this scoring result (found in the *Performance Testing of Results Section*) to the requirements of the project of having a model that scores at or above 80%. By completing this check, the team has validated the quality of the model's results.

Ethical Considerations & Concerns

Data science is a field that has many ethical concerns. From data collection to machine bias the ethical concerns need to be taken into consideration throughout the entire data science workflow. In the project, the team had to investigate the ethics behind data collection and machine learning ethics. Since the project dealt with both components of the data science workflow, it was very important to take them into consideration throughout the entire project and to continue to acknowledge them as future work continues.

Data Collection Ethics

The first piece of ethics considered was data collection. Data collection is a major ethical concern within data science but, it is also very necessary to create and train effective models. Deciding to use the Merck Manual and PubMed databases was also influenced by ethical concerns, since these sources needed to be reputable and ethical themselves. These websites are free and accessible to anyone, but this does not exactly mean that it is clearly legal. For this project, there are minimal, if any, concerns with the data collected from PubMed and Merck Manual because those websites contain anonymized data to protect individuals and avoid sub-group biases that could be amplified in the model.

Additionally, the team needed to consider potential bias that could be held within the dataset. If the data collected contained PII or PHI, this could create bias towards the patients if they are all from a similar demographic or all have similar medical issues. With the medical information collected from the websites PubMed and Merck Manual this was not a concern because the website information only has the medical terminology, definitions, and their abstracts. With this information, the risk of creating a bias in the algorithm deemed low enough for the project.

Machine Learning Ethics

General machine learning ethics is usually concerned with the model's fairness and effects on the current job market. Model fairness describes a machine learning model's fairness towards different sub-groups of people. Each group pertaining to a different race, religion, gender, disability, age, national origin, sexual orientation, gender identity, or gender expression. Specifically, machine learning in the medical domain has many considerations regarding ethics that must be considered.

A model needs to be able to show how it makes its decisions. While there are intuitive techniques that easily translate in the field of AI and computer vision, model explanation can be very difficult and lack intuition especially with word embeddings, where understanding associations created by the AI might be easy to understand. However, the overall conclusion of that might be harder to translate.

Bias analysis can be related to model fairness but refers to an equal amount of word representation in training samples. To promote model fairness, the team attempted to represent all words equally and avoid word skew-in-representation.

In the medical industry, a large majority of clinical trials clearly state the limitations of the study and the potential findings. One may say, specifically, that a study was conducted on a very small subset of people and does not translate to work on the population. When advertising an AI, it is critical that the same is done to avoid inflated expectations.

Lastly, physician understanding is one of the most important intangible aspects creating a successful product that the product team can provide the client with. Professionals in the medical field often use tools that have been proven to be reliable and trustworthy, and integrating a machine learning model into a healthcare system should have similar concerns in mind. If a physician does not understand how a tool works, they may believe that the associated risk outweighs the potential positives. To appropriately consider these concerns, the project team added code documentation and attempted to create visuals to help the public better understand and trust the finished product.

Results

Unimplemented Features

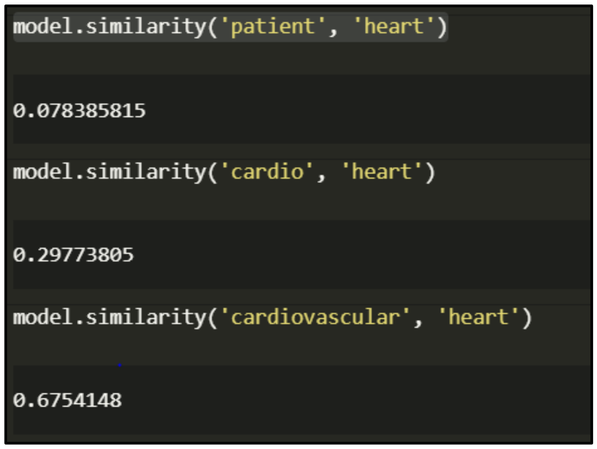
Due to time constraints, the team was unable to implement a method to highlight the newly disambiguated long form of the acronym in question, and then use color to tell the user if the algorithm got the disambiguation correct. This feature would have added some visual aspects that would have further helped patient and user understanding when interfacing with the algorithm but was deemed less-than-critical by the project team due to time constraints. Originally, this feature was discussed with the client as a potential expansion of the project, this would provide the AIpiphany team useful information about the algorithms performance and the coloring would have provided an easy to visualize solution.

Additionally, the project team considered writing code to allow an entire document to be analyzed for confusing acronyms, rather than just one sentence at a time. This plan was eventually scrapped not necessarily because of code complexity but due to the time constraints of the field session.

With the unfortunate time constraints of field sessions compounded by the data preprocessing and algorithm training taking a longer time than expected, the team was unable to implement these features into the final product provided to AIpiphany.

Performance Testing Results

Due to lengthy runtimes involved in training the model, the first results that the team was able to provide were similarity testing results. These tests help to show the models ability and understanding of relations between words based on the corpus the team used to train the model. In each of the figures below the model is being tested on different words and checking their similarity. In Figure 5, the words “patient” and “heart” do not score very high similarities while “cardiovascular” and “heart” score quite high. This shows that the model is doing its job quite well as patient is a very broad word and the dataset had many occurrences of it, making the similarity lower because the word patient is related to many components of the healthcare system.



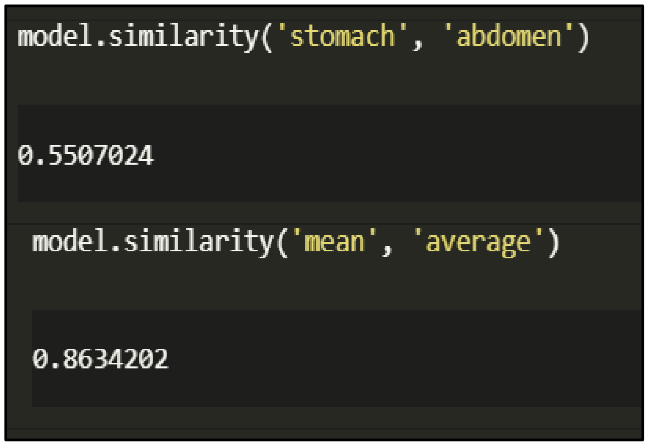
```
model.similarity('patient', 'heart')

0.078385815
model.similarity('cardio', 'heart')

0.29773805
model.similarity('cardiovascular', 'heart')

0.6754148
```

Figure 5: Word similarity values for example data 1



```
model.similarity('stomach', 'abdomen')

0.5507024
 model.similarity('mean', 'average')

 0.8634202
```

Figure 6: Word similarity values for example data 2

When the team tested the model, the resulting testing statistics indicated the model has a peak accuracy of approximately 70% correct pairings of ambiguous acronyms and their appropriate long forms. (For additional clarification, this accuracy statistic was derived from the number of correct pairings of acronyms with the appropriate long forms over the number of comparisons attempted by the model within the testing data.) While this is less than the target accuracy of 80%, an accuracy of 70% is close and can be adjusted to get to that accuracy. There is also reason to believe that the accuracy of

70% would be higher in real world scenarios because the team’s model currently is only testing on abbreviations in the ambiguous case. In the data given by the client, the ambiguous case only accounts for 43.67% of all abbreviations, which also means that the majority of abbreviations are not ambiguous and could be replaced without the team’s ML model but, with the algorithm as a whole. Assuming it is 100% accurate when transforming non-ambiguous cases, as mappings between these are 1-to-1 and rely only on the code’s ability to find them in text, then it is reasonable to assume that the model has a projected accuracy that is much higher than what the project team tested on. With the assumption that the uses of these words follow this distribution, the project team can assume that the accuracy is approximately 87%. This accuracy stands higher than the project's goal of an 80% accuracy.

Summary of Testing

Testing was done in two parts for this project and was divided based on whether the testing was in the preprocessing stage or model testing stage. The team had to test the code throughout the data preprocessing phase of development to ensure that the data was not being inadvertently altered in a manner that could adversely impact the team’s model during training and/or testing. Additionally, testing the model itself is the primary concern, since this will provide metrics describing how the model will perform in the real world.

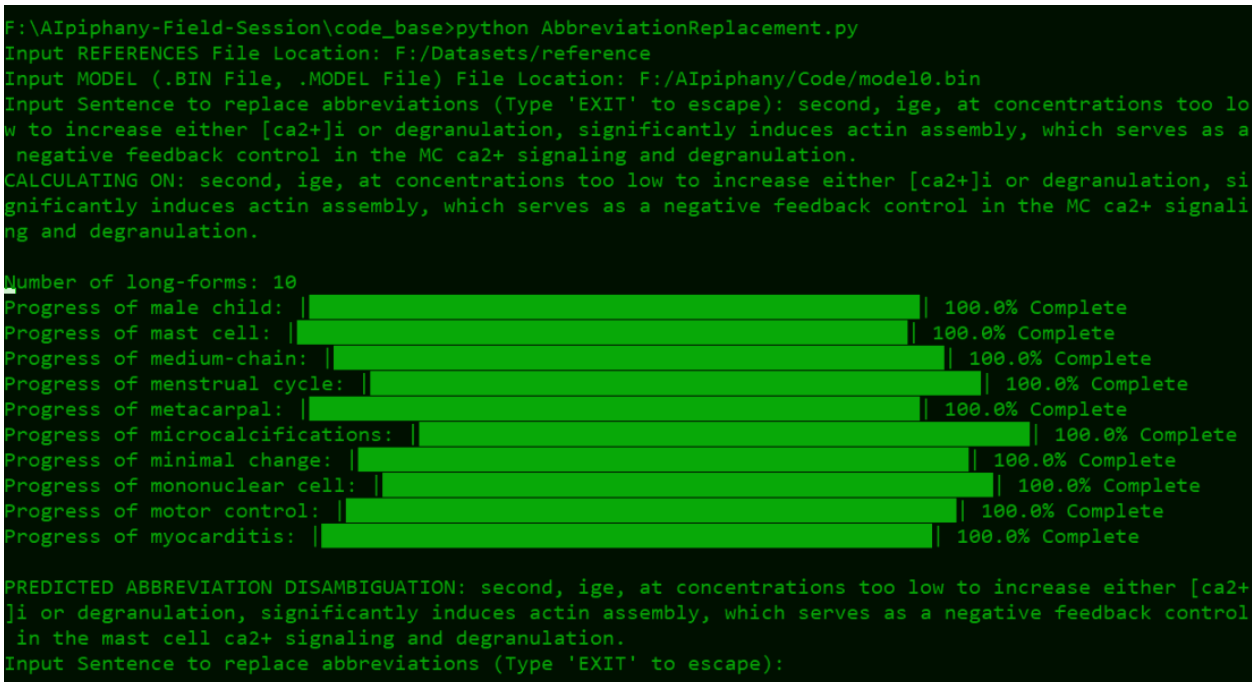
Results of Usability

The project team provided the client with a machine learning model that has been trained on a large dataset. The model can be used on various textual data, such as doctor notes, allowing the client to provide their input to the already trained model and the model will produce the disambiguated version of the provided input text.

By providing a model that can take in textual data and return the disambiguated version of the text, the field session team has completed the task that the client wished. The model is functioning and can be used, as stated above, on various textual data which allows the client to begin testing on doctor notes and adjusting as needed.

If the model proved inadequate, either now or in the future, additional training can be conducted on other datasets to improve accuracy and widen applicability. The code that the team provided to the client, other than the final model, included web scraping, file parsing, and data cleaning code. With this code the client can go and collect more data that can be used to update and retrain the model as needed. However, this process can take a large amount of time which is why the team spent such a significant portion of the field session just gathering and cleaning the data, so this process must be done with care and should only be done when large amounts of new data is found or the model is proving to be inadequate.

Included below in Figure 7 and Figure 8 are screenshots of the final program performing disambiguation on selected sentences. Figure 7 depicts an ambiguous acronym being disambiguated, which requires more processing and comparisons. Figure 8 affirms that the program works as expected for unambiguous acronyms and requires significantly fewer computations when searching for a 1-to-1 match between long form and short form.

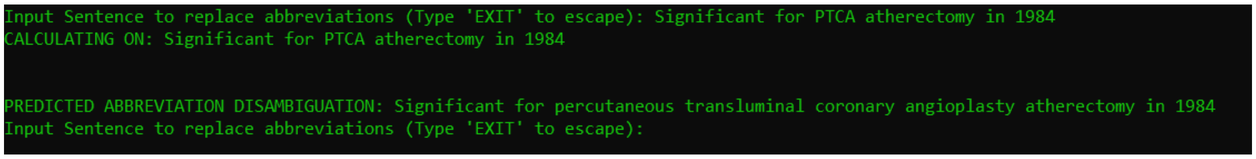


```
F:\AIpiphany-Field-Session\code_base>python AbbreviationReplacement.py
Input REFERENCES File Location: F:/Datasets/reference
Input MODEL (.BIN File, .MODEL File) File Location: F:/AIpiphany/Code/model0.bin
Input Sentence to replace abbreviations (Type 'EXIT' to escape): second, ige, at concentrations too lo
w to increase either [ca2+]i or degranulation, significantly induces actin assembly, which serves as a
 negative feedback control in the MC ca2+ signaling and degranulation.
CALCULATING ON: second, ige, at concentrations too low to increase either [ca2+]i or degranulation, si
gnificantly induces actin assembly, which serves as a negative feedback control in the MC ca2+ signali
ng and degranulation.

Number of long-forms: 10
Progress of male child: |██████████████████████████████████████████████████| 100.0% Complete
Progress of mast cell: |██████████████████████████████████████████████████| 100.0% Complete
Progress of medium-chain: |██████████████████████████████████████████████████| 100.0% Complete
Progress of menstrual cycle: |██████████████████████████████████████████████████| 100.0% Complete
Progress of metacarpal: |██████████████████████████████████████████████████| 100.0% Complete
Progress of microcalcifications: |██████████████████████████████████████████████████| 100.0% Complete
Progress of minimal change: |██████████████████████████████████████████████████| 100.0% Complete
Progress of mononuclear cell: |██████████████████████████████████████████████████| 100.0% Complete
Progress of motor control: |██████████████████████████████████████████████████| 100.0% Complete
Progress of myocarditis: |██████████████████████████████████████████████████| 100.0% Complete

PREDICTED ABBREVIATION DISAMBIGUATION: second, ige, at concentrations too low to increase either [ca2+
]i or degranulation, significantly induces actin assembly, which serves as a negative feedback control
 in the mast cell ca2+ signaling and degranulation.
Input Sentence to replace abbreviations (Type 'EXIT' to escape):
```

Figure 7: Model disambiguating on ambiguous case



```
Input Sentence to replace abbreviations (Type 'EXIT' to escape): Significant for PTCA atherectomy in 1984
CALCULATING ON: Significant for PTCA atherectomy in 1984


PREDICTED ABBREVIATION DISAMBIGUATION: Significant for percutaneous transluminal coronary angioplasty atherectomy in 1984
Input Sentence to replace abbreviations (Type 'EXIT' to escape):
```

Figure 8: Model disambiguating on unambiguous case

Future Work

Through data collection, the team was able to find examples of 67.619% of the ambiguous terms provided by the client. Examples of the remaining terms could potentially be obtained by expanding data collection sources to sources such as Wikipedia. Additional accurate data could be used to further train the model, thus increasing its accuracy and capabilities. Additionally, the team could implement a method to visualize the results of the model prediction. This would give the user some visual idea of how the model is performing by providing coloring that allows the user to see if the model predicted the disambiguation correctly or incorrectly. The team's final idea for future work would be to Improve current methodology to update data. Currently, the model can be retrained but that would take a large amount of time.

Lessons Learned

Through the development process for this project, the team learned quite a few lessons about the data science workflow and how to effectively work together. Communication within the team and with outside stakeholders proved to be critical throughout, and the team's organization and planning also allowed lines of communication to remain open with regular correspondence. Additionally, using software development principles and Scrum tools, like Trello, allowed the team to spend more time working on the project itself while minimizing unneeded downtime and distractions.

With the abundance of data sources available today, choosing data collection methods that do not violate any ethical standards is a key aspect of data science. Additionally, this data may not always come in the cleanest of forms so you must pay careful attention to the data collected and potentially apply proper cleaning techniques.

Dealing with big data can take most of the time required for machine learning projects simply due to its large size. The team collected a large amount of data that then had to be preprocessed. This preprocessing took some time due to data volume and the team did not properly estimate the potential time it may take. Additionally, identifying the acronyms in the dataset took a large amount of time but the team decided to focus its efforts on other areas during this time.

The team's machine learning model has been developed to take in the big data that the team have created and hopefully produce a result that can score at 80%. When creating and running this model the team has learned there are multiple ways to accomplish the team's goal and decided to go with a Word2Vec model because of the speed and size capabilities that outpaced BioBERT. The team also learned that this could take quite a long time to train (roughly 60 hours) and is memory intensive to run. The team had the opportunity to use a separate, dedicated high-performance computer to run model training code due

Without teamwork, a project like this would take a very long time to complete. The team had learned to communicate with each other in an effective and open way, this prevents anyone from feeling left out and left behind. If there are any issues, the team makes sure to discuss them and share with other team members what could be a better route/path that will best benefit the team. Also, the team learned how to work together remotely while maintaining positive productivity and effectiveness. While there have been some limitations and working from home is not as engaging as working in person can be, the project team has been able to successfully do this with few issues.

Bibliography

[1] J. Lee, W. Yoon, S. Kim, D. Kim, C. H. So, and J. Kang, “BioBERT: a pre-trained biomedical language representation model for biomedical text mining,” *OUP Academic*, 15-Feb-2020. [Online]. Available: https://academic.oup.com/bioinformatics/article/36/4/1234/5566506. [Accessed: 10-Jun-2020].

[2] “models.word2vec – Word2vec embeddings,” *Genism*. [Online]. Available: https://radimrehurek.com/gensim/models/word2vec.html. [Accessed: 10-Jun-2020].

[3] N. Saxena, “Word Mover's Distance for Text Similarity,” *Medium*, 28-Aug-2019. [Online]. Available: https://towardsdatascience.com/word-movers-distance-for-text-similarity-7492aeca71b0#:~:text=Word Mover's Distance targets both, embedded words of another document. [Accessed: 10-Jun-2020].

[4] “Symbolic, Distributed, and Distributional Representations for Natural Language Processing in the Era of Deep Learning: A Survey,” *ResearchGate*. [Online]. Available: https://www.researchgate.net/figure/word2vec-CBOW-model_fig1_313247648 [Accessed 10 Jun, 2020]

[5] “PubMed,” *National Center for Biotechnology Information*. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/. [Accessed: 10-Jun-2020].

[6] “The Merck Manual,” *The Merck Manual*. [Online]. Available: https://www.merckmanuals.com/. [Accessed: 10-Jun-2020].

[7] “hiQ Labs, Inc. v. LinkedIn Corp., No. 17-16783 (9th Cir. 2019),” *Justia Law*. [Online]. Available: https://law.justia.com/cases/federal/appellate-courts/ca9/17-16783/17-16783-2019-09-09.html. [Accessed: 10-Jun-2020].