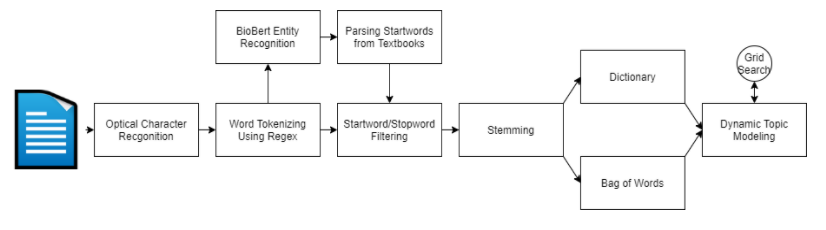
## Previous Work

This work is an expansion of the lab's previous paper[[1]](#footnote-1) using improvements in natural language processing (NLP) technologies such as graph-based clustering algorithms and transformer architectures in the form of BERT and GPT.

“Our previous work entailed compiling and and curating a corpus of over 600 published works in the neuroregeneration field spanning 1776 to 2020. We then presented an interactive time-dependent dynamic topic model specific for neuroregeneration neuroscience literature for researchers to grasp how topics change, how authors cluster based on topics, and how individual keywords rise and fall in popularity over time. Finally, we created a dynamic user interface for physicians and scientists to interact with the data. The interface displays visual results and ways to see how topic topology evolves. Moreover, the interface recommends literature for users to investigate.” The workflow for our previous work is shown in Figure 1.

[](https://private-user-images.githubusercontent.com/19865419/238205484-9ff2cc79-3589-4246-978b-6a493be1f96c.png?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJrZXkiOiJrZXkxIiwiZXhwIjoxNjg3NTU5MDQ3LCJuYmYiOjE2ODc1NTg3NDcsInBhdGgiOiIvMTk4NjU0MTkvMjM4MjA1NDg0LTlmZjJjYzc5LTM1ODktNDI0Ni05NzhiLTZhNDkzYmUxZjk2Yy5wbmc_WC1BbXotQWxnb3JpdGhtPUFXUzQtSE1BQy1TSEEyNTYmWC1BbXotQ3JlZGVudGlhbD1BS0lBSVdOSllBWDRDU1ZFSDUzQSUyRjIwMjMwNjIzJTJGdXMtZWFzdC0xJTJGczMlMkZhd3M0X3JlcXVlc3QmWC1BbXotRGF0ZT0yMDIzMDYyM1QyMjE5MDdaJlgtQW16LUV4cGlyZXM9MzAwJlgtQW16LVNpZ25hdHVyZT1iODM4NmEzODZkNDQ5MDhlOWRlOGE1MDRjYzZkZWNhYjUzNWI4MDk0MTdiOTViOGIxMWNiNmE1ZjUzNmQyNTg4JlgtQW16LVNpZ25lZEhlYWRlcnM9aG9zdCZhY3Rvcl9pZD0wJmtleV9pZD0wJnJlcG9faWQ9MCJ9.H2JLnaxQWzCGFwLk-rE6_g3rQO-suvUney5vdbuYmZ4)

“Figure 1: Overall process pipeline. The different steps of dynamic topic modeling for RegenX database is depicted in this flow chart.”

## Known Promoter and Inhibitors

To generate supervised learning sentences as well as to test accuracy of the various methods below we needed to have a set of known molecules that promoted optic nerve regeneration as well as molecules that inhibited optic nerve regeneration. These we will refer throughout this manuscript as **known promoters** and **known inhibitors** or simply **known molecules.** This list of molecules was manually created by domain experts in optic nerve regeneration and consists of 28 promoters and 19 inhibitors.

## Novel Molecules

One of the main goals of this work is to gain insight into new molecules that can be used for promoting optic nerve regeneration and inhibiting optic nerve regeneration. To identify these potential molecules we needed a way to extract these molecules from the literature. We attempted to use SciSpacy models for Named Entity Recognition trying all four of their named entity recognition models. [[2]](#footnote-2) However, the results were determined to be too noisy and not enough molecules were able to be extracted.

Therefore, we came up with another method of extracting molecules from our corpus using the scispacy abbreviation detector[[3]](#footnote-3). This was found to be much more robust and allowed for generating much higher-quality candidate molecules while also being able to extract the full name for the molecule alongside its abbreviation. This made it easier for a domain expert to curate the generated list to only select for extracted texts that referred to molecule names. The final curated list consists of 822 molecules. A portion of these molecules can be seen in the below figure.

|  |  |
| --- | --- |
| Molecule Abbreviation | Full Molecule Name |
| anti-fak | anti-focal adhesion kinase |
| vip | vasoactive intestinal peptide |
| lbp | lycium barbarum polysaccharide |
| trk | tyrosine receptor kinases, tropomyosin receptor kinase, tyro- sine kinase |
| ros | reactive oxygen species, reactive oxygen species |
| irs-1 | insulin receptor substrate 1 |
| nsc | neural stem cell |
| rarb | retinoic acid receptor beta |
| amp | adenosine monophosphate |

Table \_: Subset of the 822 molecules extracted from the corpus

## Determining Model for Word Embeddings from Literature

To evaluate the information of stored in our corpus, our first step was to generate high-quality word embeddings from the corpus. Choices had to be made on how to pre-process the data prior to generating the word embeddings including whether or not to use stop words, using stemming or lemmatization, and the choice of the word embedding model itself (Gensim vs GloVe). The Gensim model uses continuous bag of words to train word embeddings which is a neural-network approach whereas GloVe uses word co-occurrences matrices rather than a neural network to train word embeddings.

To determine the best model, the different models and model options were empirically trained on research papers from a given year. We then compared the top 20 words that were most similar to optic nerve regeneration related words and used this to determine the best choice of model and model option. The models were compared by only training on a single year due to the computational cost of training across multiple years. Table \_ illustrates this process by showing the various model’s predicted top 20 words most similar to ‘eye’ when the models were trained on the year 1907. We found the genism model trained on sentences that did not have stopwords removed and where the words were lemmatized yielded the best results.

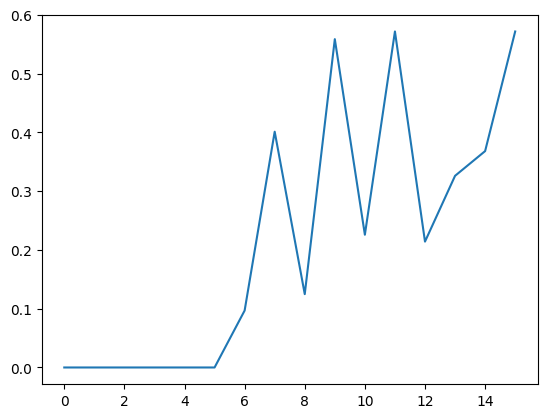
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Stemming Stopwords Included - Gensim | Stemming Stopwords Included - Glove | Stemming No Stopwords - Gensim | Stemming No Stopwords - Glove | Lemmatization Stopwords Included - Gensim | Lemmatization Stopwords Included - Glove | Lemmatization No Stopwords - Gensim | Lemmatization No Stopwords - Glove |
| part | should | treatment | angioma | part | operation | patient | lapse |
| other | neuralgia | patient | genev | lesion | sufficiently | report | direction |
| side | all | report | sound | side | semaphore | cornea | application |
| cornea | make | oper | enucl | motor | later | age | exciting |
| muscl | patient | age | sit | other | try | treatment | chorea |
| lesion | with | solut | correct | neuron | child | operation | subject |
| lower | ptosi | cornea | myopic | position | repeat | solution | host |
| involv | short | day | experi | diagnosis | his | 3 | quiet |
| on | brain | 1 | demonstr | cornea | use | paper | hausmann |
| upper | agitan | month | quiet | brain | instrument | lens | rosis |
| sensori | time | method | direct | posterior | necessary | vision | geon |
| motor | good | myopia | remov | coagulability | for | 1 | normal |
| posterior | degre | week | phototyp | muscle | should | myopia | notice |
| cell | also | tion | lid | on | trace | tion | painful |
| anterior | them | vision | stephenson | anterior | last | lid | beriberi |
| region | complic | 3 | requir | peripheral | before | normal | hypermetropia |
| opposit | wa | inject | opaqu | root | after | injection | squinting |
| peripher | repeat | 2 | lacau | papillo | become | form | remain |
| brain | trace | studi | specimen | upper | keratitis | day | require |
| posit | could | paper | stanc | paralysis | see | method | careful |

Table \_: A comparison between Gensim and Glove model with various parameters altered such as: stopwords included or not and stemming vs lemmatization used. All models shown were trained on research papers from year 1907. The rows represent the 20 most similar words each model gave for the word 'eye' and was found that the genism model trained on sentences that had stopwords removed and where the words were lemmatized yielded the best results.

## Using Our Word Embedding Model

## Gensim Model Trained Over Binned Time Periods

Our original paper binned papers into 16 sequential time periods based on when the papers were written. We then trained individual genism word embedding models across each binned time period and characterize trends across the years. An example of such an analysis can be seen in the below figure.

[](https://private-user-images.githubusercontent.com/19865419/238206675-70007890-4f0b-495c-9289-cc129591238a.png?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJrZXkiOiJrZXkxIiwiZXhwIjoxNjg3NTU5MDQ3LCJuYmYiOjE2ODc1NTg3NDcsInBhdGgiOiIvMTk4NjU0MTkvMjM4MjA2Njc1LTcwMDA3ODkwLTRmMGItNDk1Yy05Mjg5LWNjMTI5NTkxMjM4YS5wbmc_WC1BbXotQWxnb3JpdGhtPUFXUzQtSE1BQy1TSEEyNTYmWC1BbXotQ3JlZGVudGlhbD1BS0lBSVdOSllBWDRDU1ZFSDUzQSUyRjIwMjMwNjIzJTJGdXMtZWFzdC0xJTJGczMlMkZhd3M0X3JlcXVlc3QmWC1BbXotRGF0ZT0yMDIzMDYyM1QyMjE5MDdaJlgtQW16LUV4cGlyZXM9MzAwJlgtQW16LVNpZ25hdHVyZT0yYjVjYmM0YjdjYzBmNTk5MDZlMzU3Nzc4OWI4NTc4MDBkN2RmN2E3ZGNiNGYwNTBkYWZmMWZhYjg1MDA1MmYyJlgtQW16LVNpZ25lZEhlYWRlcnM9aG9zdCZhY3Rvcl9pZD0wJmtleV9pZD0wJnJlcG9faWQ9MCJ9.I5cSAy0vOP376jSrmSfh-FZMYRkKMqAGi4TKF4ICKmU)

### Figure \_: Similarity between 'neuron' and 'axon' across binned time periods when using the Gensim Word Embedding model on lemmatized sentences without their stopwords removed.

## Gensim Model Trained Over Entire Corpus

Once we had chosen our word embedding model, we trained it across all research papers to get a better representation of words across all contexts.

To determine how much information and knowledge our word embedding model was able to capture we looked at the top 10 most similar words the model computes for various of our known promoters and inhibitors. An example of the top 10 most similar words our model predicted for the known inhibitor ‘socs3’ is shown in Figure \_. The molecule ‘socs3’ is known to be involved in similar pathways to ‘pten’ which is its closest match. Furthermore, additional words such as ‘deletion’, ‘downregulation’, and ‘overexpression’ are consistent with experiments run with ‘socs3’ in neuroregeneration literature, further validating the effectiveness of our word embedding model.

|  |  |
| --- | --- |
| Similar Word to ‘socs3’ | Strength of Cosine Similarity |
| pten | 0.9128 |
| s6k1 | 0.2103 |
| deletion | 0.8966 |
| mdm4 | 0.8921 |
| mtor | 0.8900 |
| rhoa | 0..8736 |
| conditional | 0.8732 |
| downregulation | 0.8722 |
| overexpression | 0.8688 |
| p50 | 0.8615 |

Table \_: Top 10 most similar words for the known inhibitor ‘socs3’ predicted by our word embedding model.

To visualize the relationships that our Gensim model were able to extract from the literature, we plotted our word embeddings for the known promoters and inhibitors on a PCA plot as shown in Figure \_.

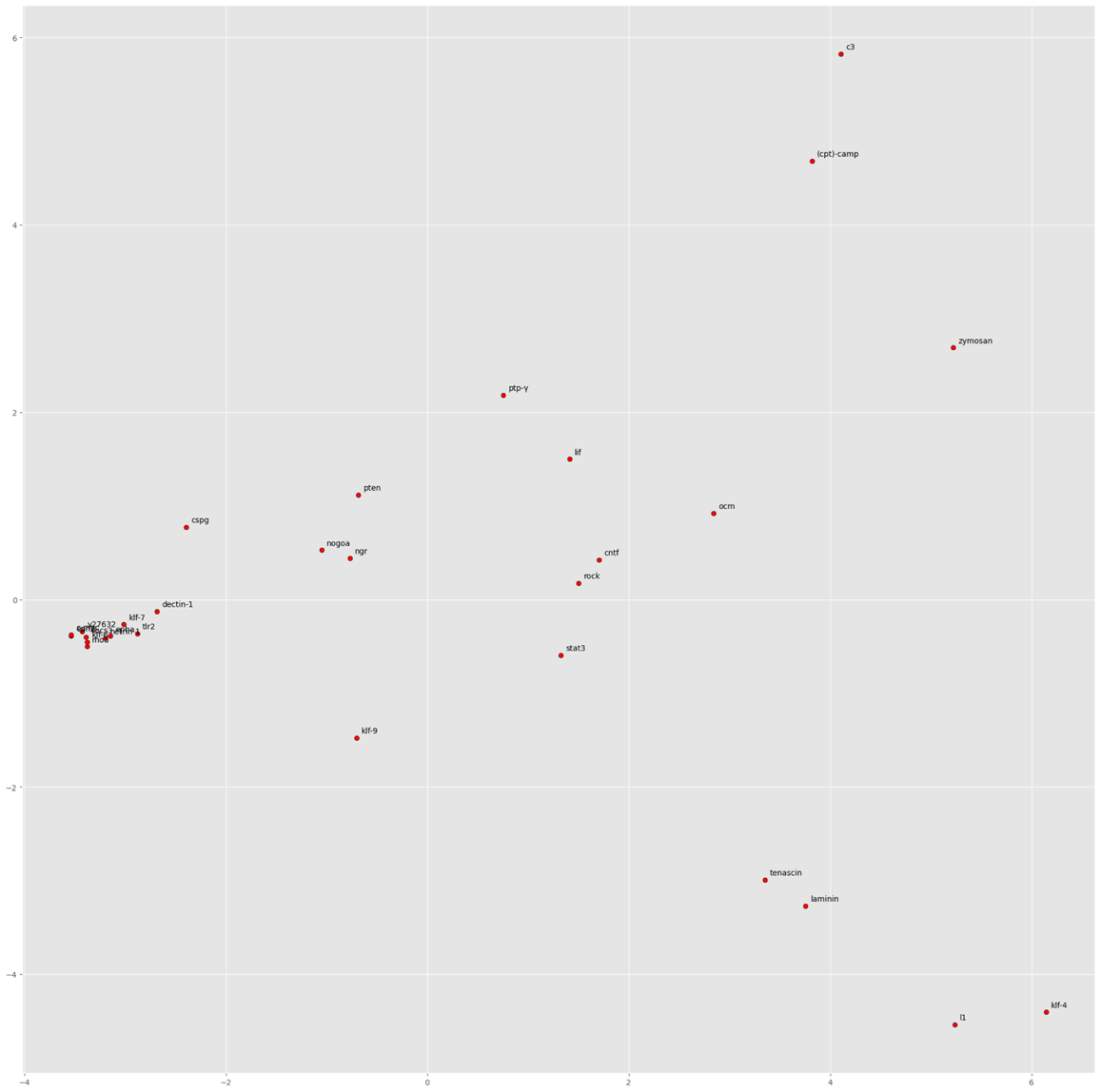


Figure \_: PCA plot of Gensim word embeddings for known promoters and inhibitors.

Notably, using the Gensim model we were only able to generate word embeddings for \_\_ out of the 822 molecules we extracted from the corpus. This is most likely due to the sparsity of some molecules leading to difficulty in the neural network converging to generate embeddings for them. Difficulty converging could also be due to the wide range of sentences and contexts the molecules could be found in.

## Graphical Approach

Using a graph to map relationships between entities has been shown as a highly effective way to derive and visualize insights in NLP.

### Creating the Graph

To construct the graph we first started with our known promoters and inhibitors. Links between molecules were generated if the two molecules occurred in the same sentence. We originally wanted to use the learned Gensim Word Embeddings to represent the nodes of the molecules however, as described previously, our word embedding model was only able to learn an embedding for \_\_ out of the 822 extracted molecules. Therefore, we used a byte-pair encoding model trained across the entire corpus which allows for generation of embeddings for all 822 extracted molecules from the corpus. This is because byte-pair-encoding can handle extremely sparse words in a corpus by learning embeddings for bytes which can be combined to form any word.

The constructed graph is shown in Figure \_ with promoters colored in green and inhibitors colored in red. The color of the links do not have meaning. The relationships seen on the graph were found to match known molecular pathways of how the known promoters and inhibitors interact with each other. Two molecular pathways that are known in optic nerve regeneration literature can be found in Supplemental Figures 1 and 2. In the constructed graph, the neighbors of the node labeled as the molecule ‘nogo’ are ‘ngr’, ‘pten’, ‘rock’, ‘cspg’, ‘omgp’, ‘mag,’ ‘rhoa’, and ‘lar’ with many of these molecules existing in the same pathway as nogo in Supplemental Figure 1. This is also seen with the neighbors of the node labeled as the molecule ‘socs3’ which are ‘pten’, ‘stats3’, ‘bdnf’, ‘camp’, and ‘cntf’ again with many of these molecules existing in the ‘socs3’ pathway in optic nerve regeneration literature as shown in Supplemental Figure 2.

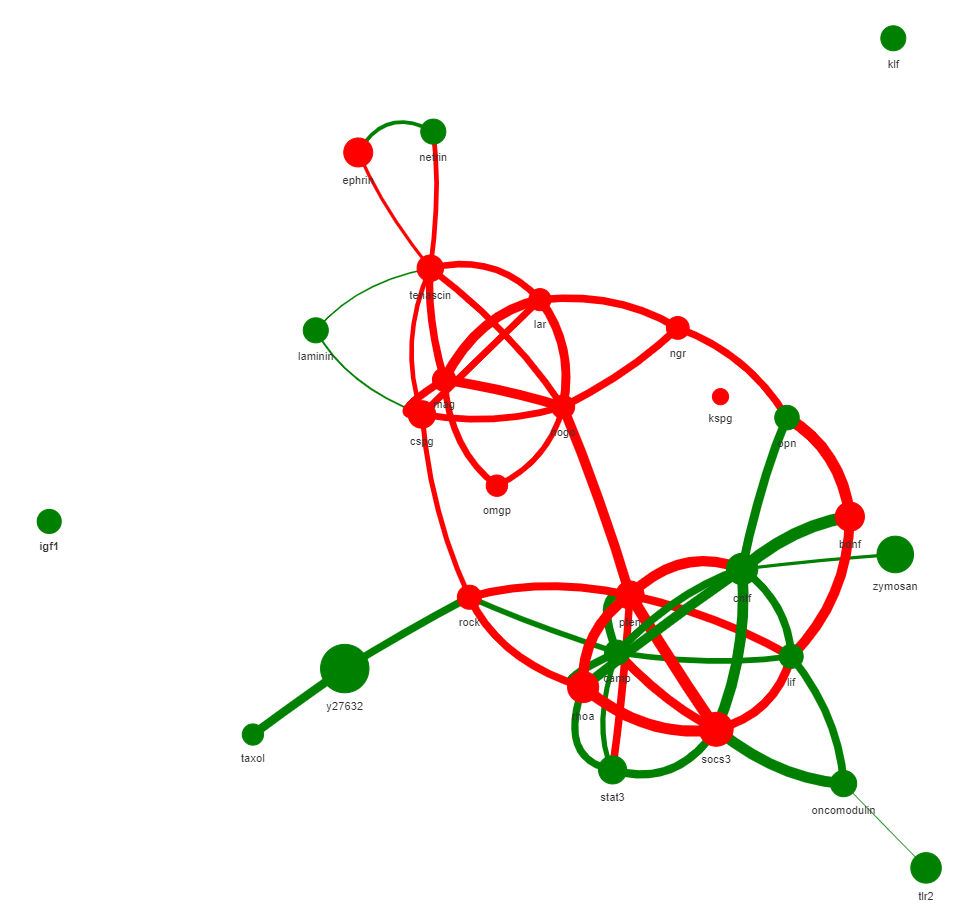
[](https://private-user-images.githubusercontent.com/19865419/238216796-12a6ff5a-6348-4be4-a44e-c074224d8423.png?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJrZXkiOiJrZXkxIiwiZXhwIjoxNjg3NTU5MDQ3LCJuYmYiOjE2ODc1NTg3NDcsInBhdGgiOiIvMTk4NjU0MTkvMjM4MjE2Nzk2LTEyYTZmZjVhLTYzNDgtNGJlNC1hNDRlLWMwNzQyMjRkODQyMy5wbmc_WC1BbXotQWxnb3JpdGhtPUFXUzQtSE1BQy1TSEEyNTYmWC1BbXotQ3JlZGVudGlhbD1BS0lBSVdOSllBWDRDU1ZFSDUzQSUyRjIwMjMwNjIzJTJGdXMtZWFzdC0xJTJGczMlMkZhd3M0X3JlcXVlc3QmWC1BbXotRGF0ZT0yMDIzMDYyM1QyMjE5MDdaJlgtQW16LUV4cGlyZXM9MzAwJlgtQW16LVNpZ25hdHVyZT04MDc3N2NmZjU4OWM2NTU4NjRmYjU4NDNiMTAxZDA5MjY0M2E3YzIwNTcwYmM5ZDlkMjkxZTJiYmQ3Y2I2ZDM4JlgtQW16LVNpZ25lZEhlYWRlcnM9aG9zdCZhY3Rvcl9pZD0wJmtleV9pZD0wJnJlcG9faWQ9MCJ9.mGyh_EtUmUAgBrAwu1qPvnOiQW-1BTvBVSGLtVC9Z7I)

Figure \_ : Graphical representation of the known promoters and inhibitors in the corpus. Node labels are the molecules and molecules that existed in the same sentence in the literature are connected to each other.

Clustering with GraphSAGE

A similar graph was constructed of all the 822 extracted molecules and known promoters and inhibitors together in one graph. After getting these molecules in a graphical format, we wanted to further explore their relationship to each other by clustering the molecules. To accomplish this we used the GraphSAGE algorithm.[[4]](#footnote-4) This algorithm allows for inductive learning of vector representations, known as node embeddings, of entities that are linked together in a graph. These learned embeddings an then be clustered in an unsupervised fashion using a 2D TSNE plot. This plot can be seen in Figure \_ where known promoters and inhibitors are labeled and the extracted molecules are represented by the unlabeled dots.

A picture containing map, text, screenshot

Description automatically generated

Figure \_: Node embeddings learned through GraphSAGE algorithm plotted using TSNE. Various clusters emerged with notable clusters labeled as A, B, and C.

Visualizing the node embeddings using the TSNE plot we could see various clusters emerge. Cluster A which consisted of ‘ngr’, ‘lar’, ‘nogo’, ‘mag’, ‘omgp’, ‘rock’, and ‘y27632’, ‘rhoa’ matched that of the known molecule pathway in Supplemental Figure 1. Similarly, cluster B consisting of ‘pten’, ‘socs3’, ‘stat3’, ‘camp’, and ‘opn’ matched that of a known molecular pathway shown in Supplemental Figure 2. Another interesting cluster was cluster C which ahd ‘bdnf’, ‘cntf’, ‘lif’ - all molecules that bind to similar receptors as shown in Supplemental Figure 2.

**Q:** Get neighbors of those molecules and find their classifications using GPT and BERT.

**A:** Yes, worthwhile doing

## Classification

It was important to not only determine the relationships between the 822 extracted molecules in our project but also to be able to classify these molecules as potential promoters or inhibitors thereby enabling future exploration of these molecules as targets for optic nerve regeneration. To accomplish this we used three different models: BERT, BioBERT, and GPT. The BERT models were trained in a supervised fashion to differentiate molecules as promoters or inhibitors whereas the GPT model had no fine-tuning and was used in a zero-shot classification fashion.

### Creating a Supervised Learning Dataset

To create a supervised learning dataset to train our classification models we used the 47 known promoters and inhibitors. For each of our known promoters and inhibitors we extracted the sentences in our corpus that molecule was in using both its abbreviation and full name as ways to find sentences. If a sentence in the corpus had both a known promoter and known inhibitor in it, then that sentence was excluded from the dataset. For each sentence for a given known promoter or inhibitor, we masked instances of both the molecule’s abbreviation and full-name in the sentences. The masking was done so that the model 1) does not overtrain on that specific molecule’s name when determining if a sentence refers to a promoter or inhibitor and 2) does not rely on previous knowledge of the molecule whose class it is trying to predict. In our corpus we found 41 of the 47 pre-labeled known promoters or inhibitors and the split of our final dataset consisted of masked sentences for 28 promoters and 19 inhibitors. [Add balanced when sentences]

**Q:** For results do we share GPT with all 41 molecules or exclude l1,c3,mag, rock, and lif?

**Q:** Macro Avg vs Weighted Avg? Went with macro since penalizes for just over guessing on the class with more examples.

**Q:** Include GPT helped us determine that ‘lif’ was a bad fit?

### Choice of Models: BERT and GPT

BERT

BERT (Bidrectional Encoder Representations from Transformers) is an open-source and state-of-the-art NLP that’s architecture was first introduced in \_\_\_\_ by Google. We compared results of using both the base BERT model and the BioBERT model.

When training BERT, we found that in the supervised learning dataset the molecules ‘l1’, ‘c3’, ‘mag’, ‘rock’, and ‘lif’ had a significant amount of sentences with little or no meaning. This is most likely due to the small and ambiguous names of these 5 molecules which resulted in many sentences that did not actually refer to the molecule being included. We found these sentences would lead to both BERT models learning spurious associations and overfitting extremely fast (see Supplemental Figures \_). Even if early stopping was implemented the model would plateau at high levels of training and validation loss when not removing these 5 molecules (see Supplemental Figures \_). Therefore, we excluded these molecules when training BERT on our supervised learning dataset. We also split the masked sentence for each molecule in our supervised learning dataset into chunks of sentences that were each less than 512 tokens as this is the maximum token limit for a single sample in BERT models. We split our chunks by sentence boundaries so as not to cut sentences in half. After splitting sentences into chunks we had 255 sentence chunks from the known promoter molecules and 202 sentence chunks from the known inhibitor molecules. The BioBERT model uses cased versions of text for input, and therefore we used cased sentences of our masked sentences for BioBERT’s training. Due to differences in the BioBERT and base BERT tokenizer, the BioBERT sentence chunks slightly differed with 210 sentence chunks from promoters and 267 sentence chunks from inhibitors.

We used the BERT sequence classification module provided from Hugging Face and trained our model with the parameter seen in Table \_. The sequence classification BERT module adds a softmax classification layer on top of the BERT pooled output that allows for using BERT for classification. We originally used 10 epochs for training with the 5 beforementioned molecules removed, but the model still overfit pretty significantly as shown in Supplemental Figure \_. Since the BERT model from HuggingFace already implements dropout and l2 regularization, we prevented overfitting by training the model only for 8 epochs. Figure \_ shows the training and validation loss for our base BERT model and Figure \_ shows it for the BioBERT model.

|  |  |  |
| --- | --- | --- |
|  | Base BERT Model | BioBERT Model |
| Model Name in HuggingFace | bert-base-uncased | dmis-lab/biobert-base-cased-v1.2 |
| Training Epochs | 8 | 8 |
| Training Batch Size | 16 | 16 |
| Warmup Steps | 500 | 500 |
| Weight Decay | 0.01 | 0.01 |

Table \_: Training parameters for base BERT model and BioBERT model

Figure \_: Figure \_: Implementing early stopping by only training for 8 epochs for the base BERT model after removing ‘l1’, ‘c3’, ‘mag’, ‘rock’, and ‘lif’. There is a significantly improved ability to improve training and validation loss when sentences from these molecules are removed from the training dataset.

Figure \_: Figure \_: Implementing early stopping by only training for 8 epochs for the base BERT model after removing ‘l1’, ‘c3’, ‘mag’, ‘rock’, and ‘lif’. There is a significantly improved ability to improve training and validation loss when sentences from these molecules are removed from the training dataset.

### GPT

GPT is another state-of-the-art NLP model developed by OpenAI that was pre-trained on wide portions of the internet and fine-tuned with human reinforced feedback learning. To test the use of a GPT model we used the GPT web interface[[5]](#footnote-5). We chose to use the web interface rather than making API calls due to decreased cost. We used GPT version 3.5 and did not do any fine-tuning on the model but instead had a zero-shot classification approach where we classified molecules into two classes: promoter or inhibitor.

**Creating GPT Prompts**

We then need to provide prompts to GPT for classifying a molecule as promoter or inhibitor based on sentences for a molecule. Three types of prompts were tested: a prompt asking for the classification only, a prompt asking for the classification as well as justification, and a prompt asking for classification, justification, and confidence score. The prompts were followed by the masked sentences from the corpus relating to the given molecule it is to label. Occasionally, GPT would not give an answer saying there was not enough information or the sentences were too ambiguous. In these cases we would respond with "Please pick an option" until it provided a classification.

|  |  |
| --- | --- |
| Prompt Type | Prompt Text |
| Classification Only | Given the following sentences from scientific studies in the field of optic nerve regeneration, where a specific molecule is masked, can you determine whether the masked molecule acts as a promoter or inhibitor of optic nerve regeneration? Please respond only with if this molecule is a promoter or inhibitor of optic nerve regeneration. No explanation is necessary. You can only respond with the word promoter or inhibitor. Here are the sentences: |
| Classification and Justification | Given the following sentences from scientific studies in the field of optic nerve regeneration, where a specific molecule is masked, can you determine whether the masked molecule acts as a promoter or inhibitor of optic nerve regeneration? Please respond with if this molecule is a promoter or inhibitor of optic nerve regeneration. Please justify your answer. Here are the sentences: |
| Classification, Justification, and Confidence | Given the following sentences from a scientific study in the field of optic nerve regeneration, where a specific molecule is masked, can you determine whether the masked molecule acts as a promoter or inhibitor of optic nerve regeneration? Provide your best guess, a confidence score from 0 (no confidence) to 100 (absolute confidence), and justify your answer based on the context given in the sentences. Here are the sentences: |

Table \_: Three different prompts used for GPT zero-shot classification

### Evaluation of our Models on Known Promoters and Inhibitors

### **GPT** We used GPT version 3.5 which has a max token size of 4906 tokens. Therefore, we had to ensure the prompts plus all masked sentences fit under the token limit for a given molecule. To narrow down the prompt types we wanted to use moving forward, we began with only using the molecules in the supervised learning dataset which had their prompts plus their masked sentences at less than 4905 tokens. This consisted of 22 of the 41 molecules in the supervised learning dataset and the result of zero-shot classification performed by GPT on these molecules is shown in Table \_. The classification and justification prompt type as well **as** classification, justification, and confidence prompt type both had the same f1 score of 0.66 which was significantly greater than the performance of the classification only prompt type with an f1 of 0.57. Therefore, we did continue evaluating with the classification only prompt type.

To classify the molecules that had a prompt size plus masked sentence size greater than 4905 tokens we split the sentences for each given molecule into separate chunks that were each under the token limit. This is similar to having to separate BERT sentences into sentence chunks of 512 tokens. We then took the mode of all the class labels predicted for each sentence chunk for a given molecule and that mode was used as the predicted class label for the molecule. These results are also in Table \_. The f1 scores for the GPT predicted labels were significantly greater than for the molecules with sentences less than 4905 tokens. We hypothesize this is due to molecules with more sentences being more studied in literature and therefore, having more clear categorizations as promoter or inhibitor in sentences. These molecules also perhaps have greater representation from recent years which tend to have higher-quality data for GPT to interpret. The justification and classification prompt performed slightly better than the justification, classification, and confidence prompt but because there were only 13 molecules this was only due to a difference of labeling one additional molecule accurately.

The results for all 41 known promoters are inhibitors, regardless of the token size of there sentences, are also shown in Table \_ with the GPT classification and justification prompt yielding a 0.80 overall f1 score versus the GPT classification, justification, and confidence score prompt which yielded an overall f1 score of 0.78. The equivalent performance between these two prompts showed the robustness and reliability of GPT classifications. It also highlighted the fact that adding confidence scores did not result in more or less accurate results for the known promoters and inhibitors.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Classification Only Prompt | Classification and Justification Prompt | Classification, Justification, and Confidence Score Prompt |
| Known Molecules <4905 tokens (N=22) | | | |
| Accuracy | 0.59 | **0.68** | **0.68** |
| Precision | 0.57 | **0.67** | **0.67** |
| Recall | 0.59 | **0.69** | **0.69** |
| F1 | 0.57 | **0.66** | **0.66** |
| Known Molecules >4905 tokens (N=19) | | | |
| Accuracy | - | **0.95** | 0.90 |
| Precision | - | **0.95** | 0.89 |
| Recall | - | **0.95** | 0.89 |
| F1 | - | **0.95** | 0.89 |
| All Known Molecules (N=41) | | | |
| Accuracy | - | **0.80** | 0.78 |
| Precision | - | **0.80** | 0.78 |
| Recall | - | **0.81** | 0.79 |
| F1 | - | **0.80** | 0.78 |

Table \_: Evaluation of GPT models on 41 known promoters and inhibitors

**GPT Justification**

One of the main advantages that GPT has is being able to provide justification for why it labeled a given molecule as a promoter or inhibitor when using a prompt that asked for justification. GPT was found to be able to determine surprisingly nuanced and complex relationships between molecules and optic nerve regenerations by using the context of provided sentences. An example of this justification is provided below for the molecule bmp4:

Based on the given context, it is likely that the masked molecule acts as an inhibitor of optic nerve regeneration. Here's the justification for this conclusion:

Sentence 1: "xcrmp-2 expression is induced by neural inducers noggin and chordin which antagonize the neural inhibitor [MASK1]": This sentence suggests that the molecule masked by [MASK1] is a neural inhibitor.

Sentence 2: "receptor also induces xcrmp-2 expression suggesting that transcription of xcrmp-2 is negatively regulated by the [MASK1] signaling": The phrase "negatively regulated" implies that the [MASK1] signaling inhibits the transcription of xcrmp-2.

Sentence 3: "xcrmp-2 is thus an early response marking neural commitment and that transcriptional control of the xcrmp-2 gene is one of the targets of [MASK1] signaling": This sentence reinforces the idea that the [MASK1] signaling influences the control of the xcrmp-2 gene, which is involved in neural commitment. Since [MASK1] signaling negatively regulates xcrmp-2 expression, it implies that [MASK1] acts as an inhibitor of optic nerve regeneration.

Sentence 4: "one study demonstrated that bmp2 and [MASK1] injection activated smad1 thereby promoting axonal regeneration": The fact that [MASK1] injection is mentioned along with bmp2, which is known to promote axonal regeneration, suggests that [MASK1] has an inhibitory effect on axonal regeneration.

Considering the consistent indication of [MASK1] as a neural inhibitor and its negative regulation of key genes involved in neural commitment and axonal regeneration, the best guess is that the masked molecule acts as an inhibitor of optic nerve regeneration.

Confidence score: 90

You can see how GPT is able to point to specific sentences in the text, explain their relevance to their molecule and optic nerve regeneration, and synthesize multiple sentences to come to a final classification. However, the example of bmp4 also demonstrates the fallacies of GPT’s justifications. For example, it correctly evaluated sentences 1,2, and 3 but its justification using sentence 4 for bmp4 being an inhibitory molecule due to it being mentioned alongside a promoter molecule does not make sense. Other examples of justifications can be seen in Supplemental Table \_.

**BERT and BioBERT**

After the BERT model and the BioBERT model were fine-tuned on the known promoters and inhibitors sentences, they were evaluated on a holdout dataset of these training sentences. Recall that the base BERT trains on uncased sentences whereas BioBERT trains on the cased sentences. The results for predicted labels for this holdout set are shown in Table \_. For the base BERT model, the f1 of 0.76 is comparable to the performance of GPT on the known promoters and inhibitors. The BioBERT model received an f1 of 0.82 actually outperforming the base BERT model and also outperforming all GPT prompts for classifying known promoters and inhibitors. Its important to note that the BERT results are based on a holdout dataset of the original supervised learning known molecules dataset whereas GPT predicted labels based on the entire supervised learning dataset. Furthermore, the BERT results shown are on a per-sentence-chunk basis rather than per-molecule basis.

|  |  |  |
| --- | --- | --- |
|  | Base BERT model | BioBERT model |
| Accuracy | 0.76 | **0.83** |
| Precision | 0.77 | **0.82** |
| Recall | 0.76 | **0.83** |
| F1 | 0.76 | **0.82** |

Table \_: Evaluation of BERT models on holdout testing set after training

Q: How is it after training on a **per molecule** basis?

A: Do it on a per molecule **basis**

**Wet-Lab Labelled Molecules Evaluation**

**GPT**

**To further evaluate the GPT model, researchers from our experimental lab manually labeled 73 molecules from the 822 extracted molecules from our corpus. These molecules’ classifications will be more challenging for the model to predict** because they are lesser known and likely have less literature associated with it.These 73 moleculesconsisted of 44 promoters and 29 inhibitors. However, only 70 molecules were able to be labelled by GPT with 3 of them GPT refusing to provide an answer despite repeatedly asking it to select an option. The 3 molecules that GPT couldn’t answer were found to have few sentences and the few sentences had very little meaning to allow for predicting a classification as promoter or inhibitor. Therefore we were left with 70 molecules consisting of 41 promoters and 29 inhibitors.

For the GPT model, the classification and justification prompt was used to perform zero-shot classification on these molecules. The results can be seen in Table \_. The molecules were also stratified by those had sentences with total length less than 4905 tokens and those that had sentences with total length greater than 4905 tokens to see how token length influenced labeling accuracy. GPT performed quite poorly with the molecules less than the token cutoff with the f1 score of 0.50 indicating it was no better than random guessing. ~~Notably, the f-1 score for predicting promoters was 0.62 versus 0.37 for inhibitors signifying its greater ability to predict promoter’s classifications rather than inhibitors.~~ The overall f1 improved significantly for molecules above the token cutoff with an f1 score of 0.76. This improvement for molecules with longer sentences was also noticed with our pre-labeled known promoters or inhibitors. ~~GPT improved on both predicting promoters (f1=0.80) and predicting inhibitors (f1=0.73). Please see Supplemental Table \_ for the full classification report across both labels.~~ Overall, the f1 score across all 70 molecules was 0.57.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Classification and Justification Prompt | Classification, Justification, and Confidence Score Prompt | Base BERT model | BioBERT model |
| All Wet Lab Labeled Molecules (N=70) | | | | |
| Accuracy | 0.57 | Not Run Yet | **0.70** | 0.61 |
| Precision | 0.55 | Not Run Yet | **0.75** | 0.62 |
| Recall | 0.55 | Not Run Yet | **0.65** | 0.62 |
| F1 | 0.55 | Not Run Yet | **0.64** | 0.61 |
| Wet Lab Labeled Molecules <4905 tokens (N=57) | | | | |
| Accuracy | 0.53 | Not Run Yet | **0.68** | 0.63 |
| Precision | 0.50 | Not Run Yet | **0.73** | 0.63 |
| Recall | 0.50 | Not Run Yet | **0.64** | 0.63 |
| F1 | 0.50 | Not Run Yet | 0.62 | **0.63** |
| Wet Lab Labeled Molecules >4905 tokens (N=13) | | | | |
| Accuracy | **0.77** | Not Run Yet | 0.62 | 0.54 |
| Precision | **0.76** | Not Run Yet | 0.65 | 0.60 |
| Recall | **0.78** | Not Run Yet | 0.65 | 0.59 |
| F1 | **0.76** | Not Run Yet | 0.62 | 0.54 |

Table \_: Evaluation of GPT and BERT models on the 70 wet-lab labeled molecules

**BERT**

The base BERT and BioBERT models were evaluated only on the same 70 molecules that GPT was able to label. Due to the token limits of BERT, as described previously, the sentences for each promoter or inhibitor were separated into sentence chunks of length 512 each. For the base BERT model, this resulted in 312 sentences from promoter molecules and 142 sentences from inhibitor molecules. For the bioBERT model, this resulted in 282 sentences from promoter molecules and 131 sentences from inhibitor molecules due to differences in tokenization when using cased sentences.

To compare the BERT models with GPT’s classification we had to predict a label for each of the 70 molecules. To do this, we used the mode of all the predicted labels for each of the sentence chunks for a given molecule. This mode was used as the predicted label for that molecule. The results are shown in Table \_ (same table as GPT results). You can see across all molecules, the base BERT model had an f1 of 0.64 which outperformed both the GPT classification and justification prompt (f1=0.55) as well as the BioBERT model (f1=0.61).

For molecules with sentences of GPT token size less than 4905 tokens, the BioBERT and BERT model are very similar with an f1 of 0.63 and 0.62 respectively. Both models outperform the GPT Classification and Justification Prompt which only had an f1 score 0.50. For molecules with sentences of GPT token size greater than 4905 tokens, the GPT model is significantly superior with an f1 score of 0.76 versus 0.62 for the base BERT model and 0.54 for BioBERT.

**Q:** For BERT this is all per-molecule basis since its only using the 70 molecules that GPT could label so didn't do per-sentence.

**Q:** Should we report 73 molecules with 3 molecules just labeled as wrong for GPT?

### Confidence Scores

Adding confidence scores for a model’s prediction allow for a quantitative way of narrowing which molecules extracted from the text are the most worth pursuing.

**BERT and BioBERT**

The BERT models have confidence scores inherent in their classification due to their classification being the maximum of the output of their softmax layer. To obtain the confidence score for a given prediction, you simply use the probability output that the model assigned to the predicted label. To study if confidence scores have meaning we looked at if confidence scores and accuracy were significantly correlated. Significant correlation was done using the spearman rank correlation test due to our confidence scores being in a nonparametric distribution (see Figure \_). We looked at the confidence scores for both the holdout evaluation dataset for the known promoters and inhibitors as well as the manually-labeled 70 molecules to determine if there was a correlation. The results are shown in Table \_. The known promoter and inhibitors confidence scores’ correlation to accuracy of labeling their 512 token sentence chunks was significant for both the base BERT (p=4.73e-6) and BioBERT (p=1.4e-4) model. The confidence scores of the wet lab molecules were similarly tested for their correlation to their 512 sentence chunks labeling accuracy, i.e per-sentence-chunk basis, but also on the accuracy per molecule label. Recall, the molecule label was determined by the mode of all sentence chunk predictions for that molecule. The base BERT model’s confidence scores were not correlated to either the sentence chunks or molecule labels for the 70 wet-lab labeled molecules (p=.110 and p=.896). However, the BioBERT model was significantly correlated on the molecules’ sentence chunks (p=0.0025) but not on the molecule labels (p=0.84).

**Q:** Should we stratify confidence score by >4905 tokens and <4905 tokens too for BERT? We already did for GPT

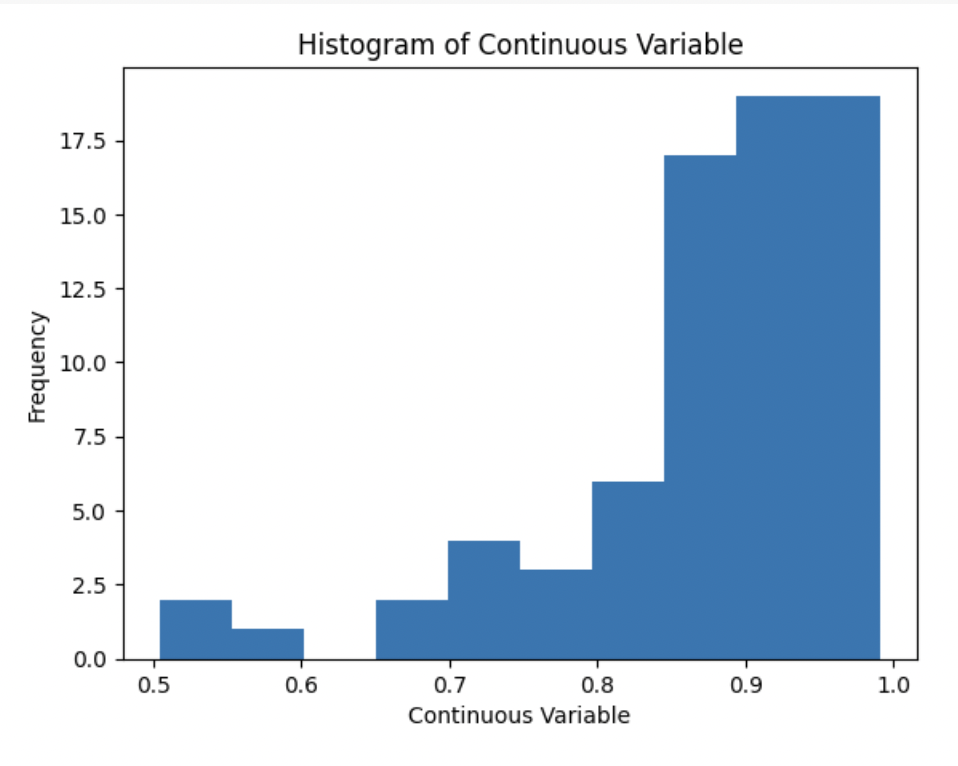


Figure \_: BioBERT distribution of confidence scores for the 70 wet-lab labeled molecules. Similar to confidence score predictions for the base BERT model. Note the non-parametric distribution.

|  |  |  |  |
| --- | --- | --- | --- |
|  | GPT Classification, Justification, and Confidence Score Prompt | Base BERT model | BioBERT model |
| Known Promoter/Inhibitors | | | |
| Spearman Correlation Coefficient | 0.1154 | 0.46 | 0.38 |
| P-Value | 0.4723 | **0.0000047** | **0.00014** |
| Wet Lab Molecules (per sentence basis) | | | |
| Spearman Correlation Coefficient | Not Run Yet | 0.075 | 0.15 |
| P-Value | Not Run Yet | 0.110 | **0.0025** |
| Wet Lab Molecules (per molecule basis) | | | |
| Spearman Correlation Coefficient | Not Run Yet | -0.015 | -0.02 |
| P-Value | Not Run Yet | 0.896 | 0.84 |

Table \_: Confidence scores correlation to label accuracy for GPT and BERT models

1= molecule was correctly labeled (by GPT or BERT) when compared to ground truth

0 = molecule was incorrectly labeled (by GPT or BERT) when compared to ground truth

[90, 1]

[60, 0]

[80,1]

[100,1]

[80, 1]

MAP = mean average precision

**GPT**

To obtain the confidence score for GPT we had to use the classification, justification, and confidence score prompt. For all 43 known promoters and inhibitors, the confidence score outputted by GPT and the labeling accuracy of the molecule were not significantly correlated as shown in Table \_. However, we did find that the scores were more correlated to accuracy for the molecules with sentences greater than 4905 token molecules compared to the molecules with sentences less than 4905 tokens (r=0.09 vs r=0.19, see Supplemental Figure \_).

## Data Characterization of Sentences Per Year

“We have compiled and curated a corpus of 700 published works in the neuroregeneration field spanning from the years 1776 to 2020. To assemble this corpus, we searched in different databases, including Google Scholar and PubMed. During our search, we also included papers in other languages, including French and German. In the 18th century, the scientific community was heterogeneous, and scientists often communicated their findings in their native languages. Despite this variation in language, the majority of the corpus is in English.” Figure \_ illustrates the document distribution across the years.

We first wanted to visualize the overall distribution of sentences per year in our corpus.

Due to the wide span of years or corpus covers, we also wanted to see which years the sentences were coming from for both our 43 known promoters and inhibitors used for our supervised learning of BERT as well as the 70 extracted molecules that was used in evaluating all the models.

A picture containing text, screenshot, font, diagram

Description automatically generated

Figure \_: Literature distribution over time: 1771 to 2020. The curated papers from 1700–2020, x- and y-axis are years of citation and number of documents obtained respectively.

Figure \_ shows the distribution of sentences per year across our entire corpus (1776 – 2015). You can note a trimodal distribution with the majority of sentences in our corpus falling into 3 buckets of time periods: 1907–1923, 1942-1944, and 2000-2015. Figure \_ shows the distribution of sentences per year for our supervised learning dataset of the 43 pre-labeled promoters and inhibitors. Here, the molecules are much more concentrated in the years 1995 – 2015. This makes sense as to classify these as promoters and inhibitors our team needed to pull from high-quality research which is more likely to occur during this time period. Figure \_ shows the distribution of sentences per year for the sentences for the 70 wet-lab labeled molecules that were classified by BERT and GPT. As you can see, this is similarly skewed toward the recent years just like the training dataset.

[ ] Find 2015-2020 papers

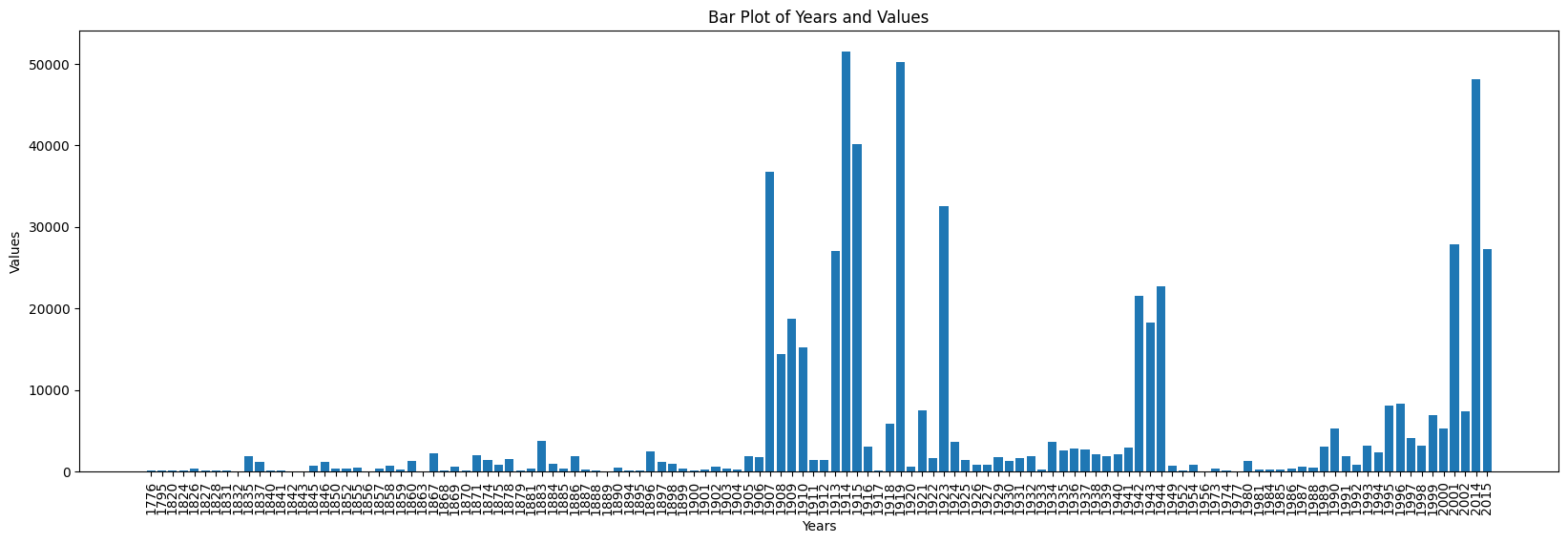
[](https://private-user-images.githubusercontent.com/19865419/248416409-a4620c0f-8924-4474-8232-e178eef508b8.png?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJrZXkiOiJrZXkxIiwiZXhwIjoxNjg3NTU5MDQ3LCJuYmYiOjE2ODc1NTg3NDcsInBhdGgiOiIvMTk4NjU0MTkvMjQ4NDE2NDA5LWE0NjIwYzBmLTg5MjQtNDQ3NC04MjMyLWUxNzhlZWY1MDhiOC5wbmc_WC1BbXotQWxnb3JpdGhtPUFXUzQtSE1BQy1TSEEyNTYmWC1BbXotQ3JlZGVudGlhbD1BS0lBSVdOSllBWDRDU1ZFSDUzQSUyRjIwMjMwNjIzJTJGdXMtZWFzdC0xJTJGczMlMkZhd3M0X3JlcXVlc3QmWC1BbXotRGF0ZT0yMDIzMDYyM1QyMjE5MDdaJlgtQW16LUV4cGlyZXM9MzAwJlgtQW16LVNpZ25hdHVyZT03ODc3OGMyM2QwYjMzN2I4YTNiYWI0M2M4OTJjYzI4YzgzMDU3ZDdiZGQ4ODZkZmUzZWMwZjg3ZDQyZDQ0NWUwJlgtQW16LVNpZ25lZEhlYWRlcnM9aG9zdCZhY3Rvcl9pZD0wJmtleV9pZD0wJnJlcG9faWQ9MCJ9.cOQn3kj-EqThwcmcBIHzl1mYVrfK18g4KWhADG_vtqc)

Figure \_: Number of sentences per year for entire corpus

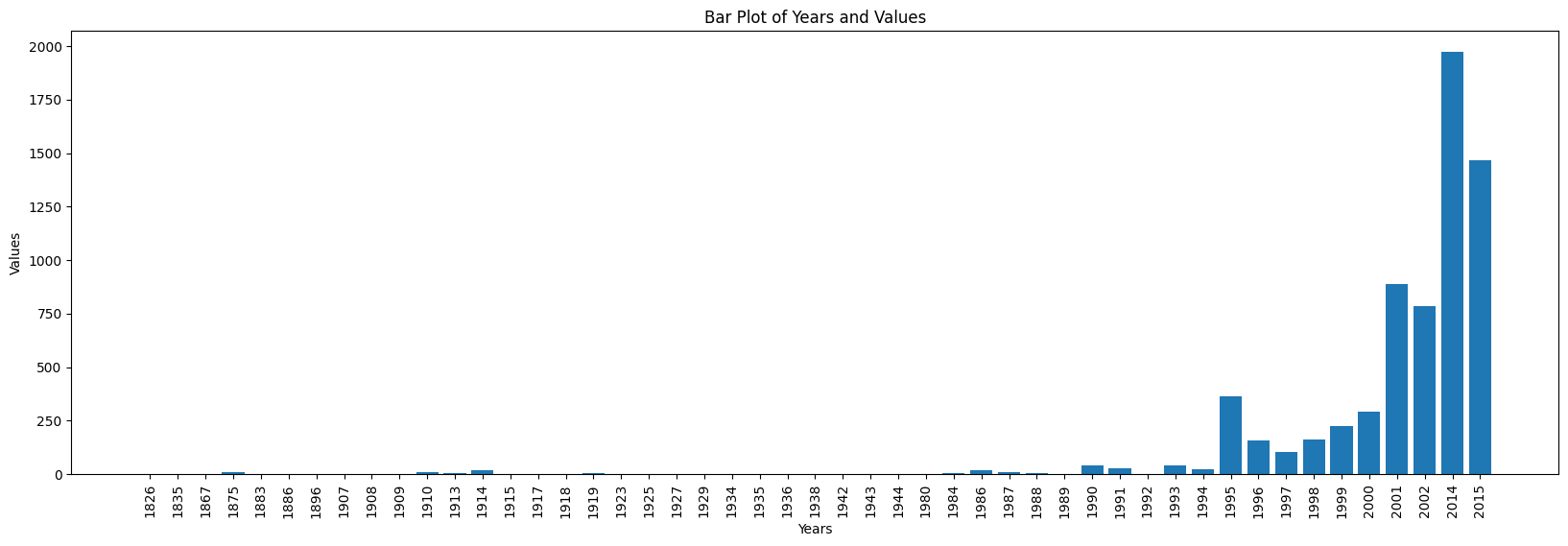
[](https://private-user-images.githubusercontent.com/19865419/248416686-af9c2809-9044-4f91-9acc-a089d34af836.png?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJrZXkiOiJrZXkxIiwiZXhwIjoxNjg3NTU5MDQ3LCJuYmYiOjE2ODc1NTg3NDcsInBhdGgiOiIvMTk4NjU0MTkvMjQ4NDE2Njg2LWFmOWMyODA5LTkwNDQtNGY5MS05YWNjLWEwODlkMzRhZjgzNi5wbmc_WC1BbXotQWxnb3JpdGhtPUFXUzQtSE1BQy1TSEEyNTYmWC1BbXotQ3JlZGVudGlhbD1BS0lBSVdOSllBWDRDU1ZFSDUzQSUyRjIwMjMwNjIzJTJGdXMtZWFzdC0xJTJGczMlMkZhd3M0X3JlcXVlc3QmWC1BbXotRGF0ZT0yMDIzMDYyM1QyMjE5MDdaJlgtQW16LUV4cGlyZXM9MzAwJlgtQW16LVNpZ25hdHVyZT1mZmM4MzA1NGQwMzEwMTNiZTMwY2JjYTVkNTE5MDFmZDAyYmY5ZTUxYTNkYjMzOTNhZTc1Njg5NTdkMDE2MzA5JlgtQW16LVNpZ25lZEhlYWRlcnM9aG9zdCZhY3Rvcl9pZD0wJmtleV9pZD0wJnJlcG9faWQ9MCJ9.sqBlEUXsRRMZzP2Zs8KX7YE-Lukq-5PhKX1T9Ltw8AY)

Figure \_: Number of sentences per year for 43 Known Molecules  
A picture containing screenshot, line, plot, rectangle

Description automatically generated

Figure \_: Number of sentences per year for 70 wet-lab labeled molecules

## All Questions

**Q:** Get neighbors of those molecules and find their classifications using GPT and BERT.

**Q:** For results do we share GPT with all 41 molecules or exclude l1,c3,mag, rock, and lif?

**Q:** Macro Avg vs Weighted Avg? Went with macro since penalizes for just over guessing on the class with more examples.

**Q:** Should we include that GPT helped us determine that ‘lif’ was a bad fit?

**Q:** How is BERT and BioBERT on holdout evaluation dataset on a **per molecule** basis?

**Q:** For BERT wet label evaluation, it is all per-molecule basis since its only using the 70 molecules that GPT could label so didn't do per-sentence-chunk. Should we do per-sentence-chunk?

**Q:** Should we report 73 molecules with 3 molecules just labeled as wrong for GPT?

**Q:** Should we stratify confidence score by >4905 tokens and <4905 tokens too for BERT? We already did for GPT

## Future Steps

* [✔️] Regraph Word2Vec embeddings with known molecules in graphsage and see if you get similar clusters that group on pathways
  + [✔️] Get F1 when comparing known molecules to “promoter” and “inhibitor”
  + [✔️] Get F1 when comparing wet lab labelled molecules to “promoter” and “inhibitor”
* [✔️] Graph BPE and see how clusters are
  + [✔️] Create a BPE model trained just on recent year and see how that does
  + [✔️] Get F1 when comparing known molecules to “promoter” and “inhibitor”
  + [✔️] Get F1 when comparing wet-label molecules to “promoter” and “inhibitor”
* [✔️] Use confidence scores with known molecules using GPT and see if they correlate to accuracy of predicted label
* [✔️] Check BERT model confidence scores on a per molecule basis
* [✔️] Get distribution of years for masked known molecules and masked wet lab molecules
* [✔️] Try removing 'lif' from BERT training as well...
* [✔️] Add biobert results
* [✔️] Add data characterization by year results
* [✔️] Create Slides for results and convert Readme.md to word document
* [ ] Go through and check support on classification report matches the splits I’ve been reporting
* Meet with Dr. J to see next steps for final GPT model for known molecules and 73 wet lab labeled
* For final decided GPT model remove l1, c3, mag, and rock from results since BERT did not use those for known molecules (or don’t)
* Do per-molecule basis of BERT known molecules if Dr. J wants it
* Add >4905 and <4905 confidence scores for BERT only if Dr. J wants it
* Test BERT model on explicit sentences to see how it does
* If you really want to evaluate performance you need to have manual graders for the masked sentences that GPT and BERT are given and see how well they do to a manual labeler. Because right now we are seeing its accuracy when compared to someone that has access to all literature to make a classification.
* [❌] Using Logistic Regression with Wet Lab Molecules how is the F1? -> However would have to convert wet lab molecule sentences to freq vectors so not straightforward will take time probably not worth it since won't be included in final paper
* [❌] Using Naive-Bayes with Wet Lab Molecules how is the F1? -> However would have to convert wet lab molecule sentences to smoothed freq vectors so not straightforward will take time probably not worth it since won't be included in final paper
* [❌] Using GPT4 which would allow for longer token sizes for input. Unfortunately the cost is very high and the web API has a cap of 25 messages every 3 hours.

1. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9531894/> [↑](#footnote-ref-1)
2. <https://scispacy.apps.allenai.org/> [↑](#footnote-ref-2)
3. <https://github.com/allenai/scispacy> [↑](#footnote-ref-3)
4. <https://cs.stanford.edu/people/jure/pubs/graphsage-nips17.pdf> [↑](#footnote-ref-4)
5. chat.openai.com [↑](#footnote-ref-5)