OpenAlex: End-to-End Process for Topic Classification

This article describes the process that OpenAlex uses to assign topics to works in our database. This system was built on top of the recently released classifications from CWTS (An open approach for classifying research publications) which provided an excellent labeled dataset (Van Eck, 2024) that could be used for training a model. OpenAlex fine-tuned the multilingual BERT model and integrated that into a larger deep learning model that makes use of additional information such as the journal name and citation graph features. The end result is a model that can accurately predict topics using title, abstract, citations, and journal name.

OpenAlex and MAG Background

Microsoft Academic Graph (MAG) was a heterogeneous graph of research entities (including publications, journals, authors, institutions, and fields of study) and relationships between those entities (Sinha et al., 2015). Unfortunately, Microsoft discontinued this service at the end of 2021, to the dismay of users across diverse communities (including computer science, meta-research, bibliometrics, libraries, and startups).

OpenAlex (Priem et al., 2022) was launched to fill the gap. OpenAlex is a free and open catalog of the world's scholarly papers, researchers, journals, and institutions — along with all the ways they're connected to one another. The data is currently available via an API, a full database snapshot, and also a UI. For more information about OpenAlex, feel free to go to the website: OpenAlex.

OpenAlex Concepts (Legacy)

In order to provide a seamless transition from MAG to OpenAlex, an initial Concepts model was created based off of the MAG field of study data in order to provide users with the same quality of concept tagging. While this model is powerful and accurate, we believed that there was room for improvement in the way we categorize scholarly works. The MAG-based concepts have some problems—among them: (i) they use a somewhat convoluted and unintuitive concept hierarchy; (ii) there are frequent problems with polysemy and ambiguity of terms; (iii) Concepts are static and do not change with real-world trends.

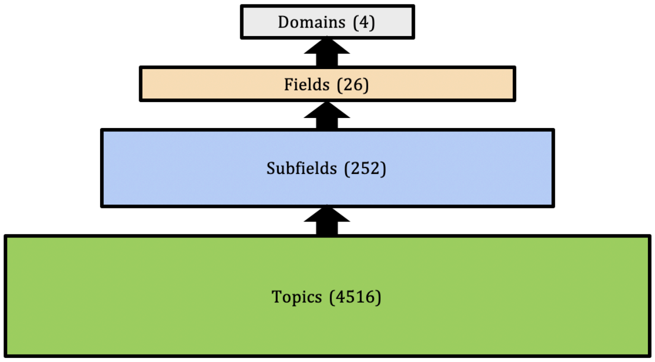
CWTS Classifications

In a recent effort by CWTS to create an open data version of fields of study/classifications for research publications (An open approach for classifying research publications), a labeled dataset was created which assigned a “micro-level” field to each paper that contained a citation in

OpenAlex. The methodology for this process can be found in a paper from 2012 by Waltman and Van Eck (A new methodology for constructing a publication‐level classification system of science). A paper from 2019 by Traag et al. (From Louvain to Leiden: guaranteeing well-connected communities) describes the clustering algorithm used. CWTS plans to update this dataset annually based on the latest publication data. OpenAlex decided to create a model based on the CWTS dataset which assigns topics to works as they come into the database.

OpenAlex Topics

OpenAlex Topics is a new system of topics and labels which can be used to accurately and succinctly describe what a paper is about. The new entity (Topics) ensures a one-to-one relationship between the topics and the higher levels of fields. The diagram below shows the new Topics structure:



The subfields, fields, and domains are from Scopus's ASJC structure and give users an established structure that they might be familiar with. As opposed to other systems that apply ASJC at the journal level, we will be applying the topics at the paper level which gives an increased level of granularity. Each topic will belong to one subfield, which will belong to one field, which will belong to one domain. See the following as an example:

Topic:	Natural Language Processing
Subfield:	Artificial Intelligence
Field:	Computer Science
Domain:	Physical Sciences

The full list of topics with their associated subfields, fields, and domains is available in the following spreadsheet: OpenAlex_topic_mapping_table .

The only missing link in creating this structure was how to connect the topics provided by CWTS to the ASJC structure. To solve this problem in an automated way, OpenAlex used a

Large Language Model (LLM) to assign topics to the most likely field and using that selection, the LLM then selected the most likely subfield. This allowed our team to efficiently connect the two separate pieces of data in a way that did not make someone go through over 4,000 topics and manually assign them to a subtopic. We expect to find situations where the LLM did not assign the correct field/subfield, but from testing the results, we are confident that the LLM does a good job of making these assignments. In addition to potential inaccuracies, there are pitfalls around ambiguity—some topics could belong to multiple subfields, and so an LLM might get confused. We plan to make adjustments based on user feedback.

Open to Suggestions/Improvements

Since this entire process (receiving the labeled data from CWTS, assigning the topic names, creating a topics model based off of that data, and assigning topics to subfields) is brand new, OpenAlex understands that not everything will be perfect. Since most of the ecosystem was created in an automated way, users will find things that don't make sense and need to be changed. There will also be biases found in the data since we are using openly available language models that were trained on data that contains biases. If there is any feedback regarding ways the system could be improved, we are always happy to hear them and make adjustments to our models to help OpenAlex improve. We are open to any suggestions regarding the names of the topics and also the assignments of the subfields. Feel free to submit a support ticket to report anything you see that should be brought to our attention (OpenAlex Support Request).

Data Exploration and Feature Creation

Initial exploration and testing was done on a classification of the OpenAlex August 2023 data snapshot. With this data, features were created and the initial version of the topics model was developed and tested. Later, CWTS finalized the classifications using the November 2023 snapshot and this data was used to create the final model. The following sections will detail how the data was explored, how the model features were created, and why we went the direction we did with certain elements of the data.

Data Exploration

There were two separate files (CWTS Classification Data) that were received from CWTS:

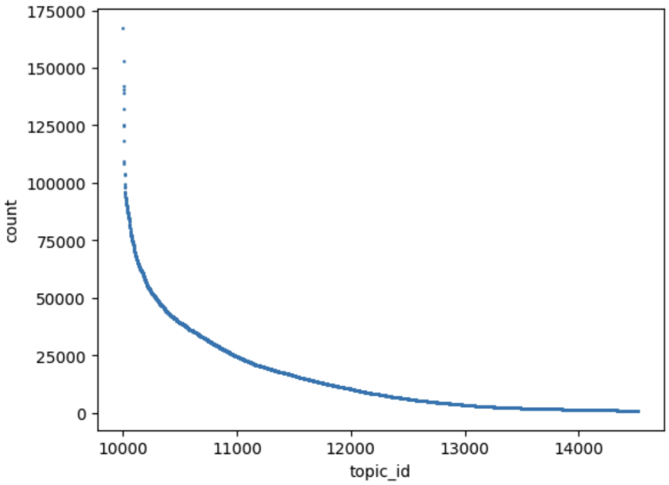
1. Classification labels
2. OpenAlex labeled data

For the classification labels, CWTS first clustered all data in OpenAlex that contained either outgoing or incoming citations (n = 71M works). It was from these clusters that the "micro-level" classifications were created, which is what OpenAlex is now calling "Topics". In previous years, CWTS would generate a set of important keywords for each cluster and those would be considered the labels for each cluster. However, with the increased performance of Large Language Models (LLMs) over the past year, CWTS used a different approach to label each cluster. The process is described in more detail in their blog post announcement (An open approach for classifying research publications) but at a high-level, CWTS fed paper titles for each cluster into an LLM and then asked the LLM to return a label, keywords, a summary, and a wikipedia URL. Our sampling of the labels and the labeled data showed us that the LLM generated mostly acceptable labels. These automated methods are never perfect, however; but finding errors such as poorly labeled Topics and making corrections will be a joint effort between us and the community over the next year.

The other file provided by CWTS was a labeled dataset of over 70 million rows which assigned a single topic to a paper (linked using an OpenAlex Work ID). The methodology for labeling this data can be found in Waltman & van Eck, 2012, and Traag et al., 2019 describes the clustering algorithm used. Since the data was labeled using an OpenAlex Work ID, we were able to connect other data in our ecosystem to explore trends with different features such as title, abstract, citations, journal, and publication date. Here are some general statistics about the dataset:

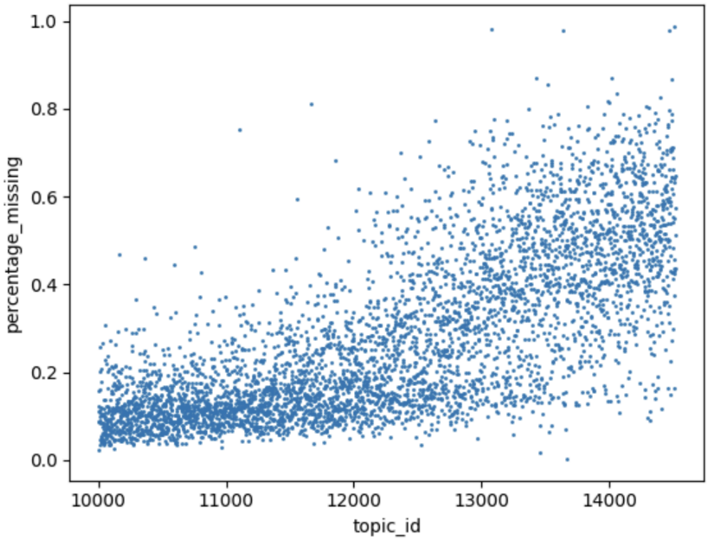
Smallest Cluster Size	1,000 papers
Median Cluster Size	7,954 papers
Largest Cluster Size	167,455 papers

First, we looked at the counts of the labeled data in order to see the overall distribution of how many papers belong to each topic.



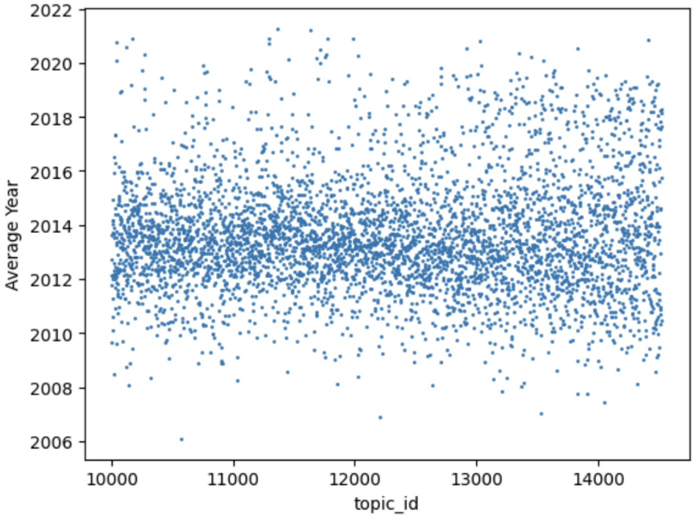
As you can see, the topic IDs were created in order of descending count, which means lower number IDs (for example, 10001) will have higher counts than higher number IDs (for example, 13600). This information will be used throughout our exploration and model testing.

Because the algorithm/process used to create this labeled dataset involved incoming and outgoing citations, we wanted to take a look to see if there are any trends observed when we only look at papers with outgoing citations. In a live production setting, outgoing citations are going to be the only citation feature we can use because only outgoing citations will be available when a newly published work is ingested into OpenAlex. Therefore, we calculated a “percentage_missing” metric to show what percentage of the labeled data contains incoming citations for each topic ID. This would be considered “missing data” for our model since we will be unable to use it.



It would appear that as the topic ID increases (i.e., as the labeled topic counts decrease), the algorithm created clusters that relied more heavily on incoming citations than outgoing citations. This will be good to keep in mind for the future, especially since we discuss later on that the model accuracy drops for higher levels of topic IDs.

One last thing to share was the average publication year distribution across the different topic IDs in order to see if as the labeled topic counts decreased, there were differences in the average publication year. We would expect that the size of the cluster has no relationship to the average publication year of the cluster.



There is a slight change as the topic IDs increase but we do not see the average publication year trending up or down, which is what we were hoping to observe. We looked into many other variables through the exploration but these were the main points we wanted to focus on for this analysis.

Feature Engineering

CWTS created the clusters/topics using only data that had incoming or outgoing citation data, but that only covers about a third of the data available in OpenAlex, and incoming citation data is not going to be available immediately when a newly published paper is ingested in OpenAlex. By using other features of works in addition to citations, we can classify all of these works that would otherwise need to be excluded, and even apply the classification to other research objects such as grants.

Based on our concepts model, we know that title, abstract, and journal name provide good signals for determining the topic of a paper. However, we wanted to improve on that by introducing citation features into our model so that signal could be used to predict as well. We will go through each of the features to talk about why we are including them and what was done to include them in our model.

Title, Abstract and Journal Name

The title and abstract are important features to bring into our model. After preprocessing, we found that about 90% of our works had a title that could be used as a feature to our model, and about 50% of works had a usable abstract.

Through testing, we discovered that our model still did not perform well on non-Latin character languages. This means that, unfortunately, any titles or abstracts in those languages could not be used as features to our model. However, we hope that for some of these works, the citation features (discussed below) will help to still predict an accurate topic for those papers.

After the preprocessing steps, the title and abstract are merged into a single text feature that are then sent through a tokenizer before being fed to a language model. This selection of this language model is discussed in a later section.

Journal names are also used as a feature for the final model. We wanted to use a feature that could generalize to new journals as opposed to using a fixed list of journals. Therefore, we apply some of the same preprocessing steps as the title/abstract before sending the journal name through an embedding model (sentence-transformers/all-MiniLM-L6-v2). The output embedding from the model was used as an input feature to the final topics model. This model was chosen for the low latency since it will be used each time there is a call to the model.

Citation Features

As mentioned above, citations are the major feature we wanted to add to this model, for the following reasons:

1. The citation graph has been shown to be useful in predicting topics of research papers (Waltman & van Eck, 2012).
2. Sometimes the signal from the title or abstract is weak (or non-existent).
3. Works in languages other than English have a better chance at having an accurate topic assigned if there is a feature that doesn't depend on language alone.

We designed a feature called “gold citations” to meet several conditions. We wanted to make sure that the feature would not be too sparse (for example, a feature where every citation is an input to the model). The feature would also need to be available in a production environment, which meant incoming citations could not be used. Lastly, the feature needed to be generalized to many works and be usable for works in the future.

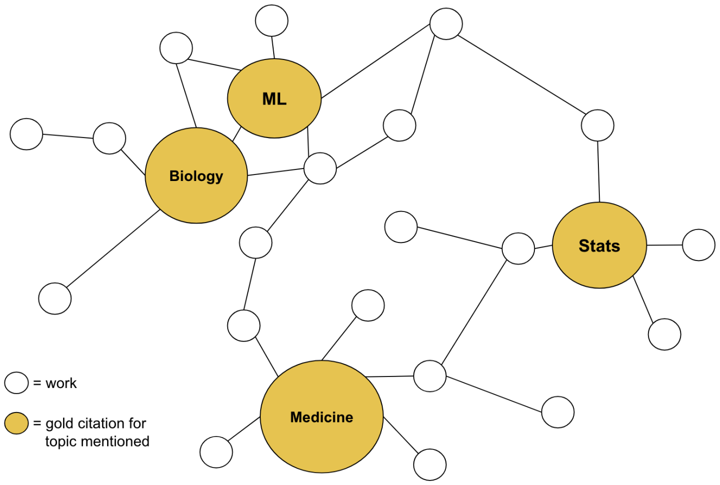
Gold Citations

We programmatically compiled a list of citations for each topic that were considered important and labeled them as gold citations. Basically, they are the highest cited works for each topic with some filtered out if they are highly cited in many topics. We set a maximum number of gold citations for each topic based on the labeled data count for that topic. Each topic ended up with somewhere between 10-75 gold citations which could then be used to generate citation features for each work. In the end, there were about 120k gold citations across all topics. We experimented with increasing the number of gold citations allowed for each topic, but we determined that higher numbers were leading to instability in the model and decreased accuracy. We came up with some theories as to why this was the case, the two most probable being:

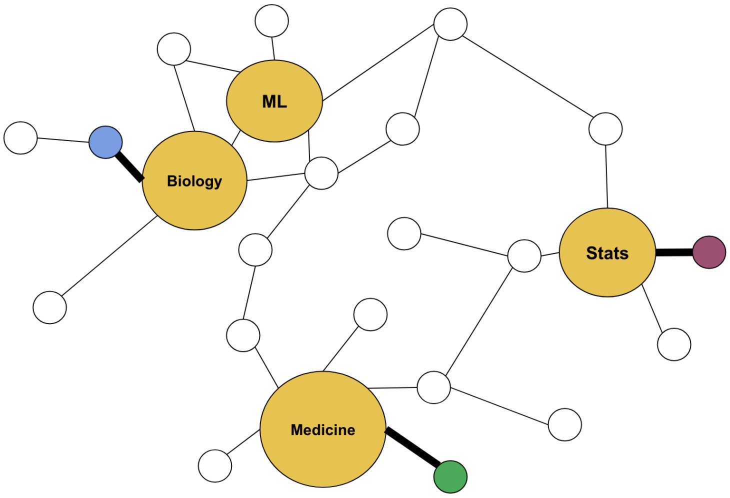
- Increasing the number of gold citations increases the “noise” in each topic, making it more difficult for the model to pick up the signal
- Not enough training data to support the increased number of gold citations

Creating the Features

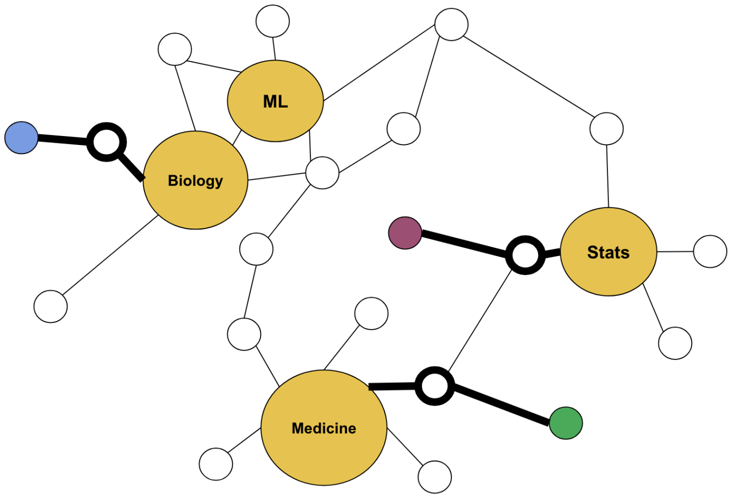
From the gold citations, 2 basic features were created for all works. The first is a list of gold citations that are directly cited in the work which is called Citation 1. The other feature, Citation 2, is when one of the citations in a work directly cites a gold citation (in graph network terms, the gold citation is 2 edges away). The following figure shows a fake citation graph that we will use to explain how the features are created:



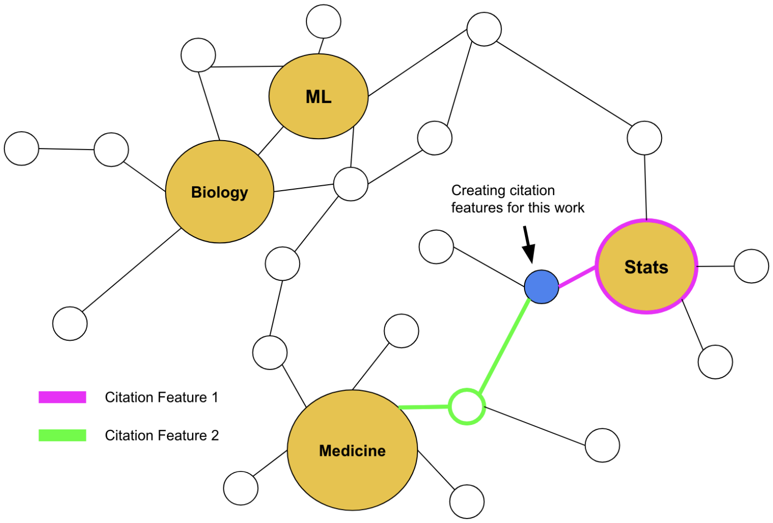
In the figure, each white dot represents a work that we will be looking to create citation features for while each gold circle represents a gold citation for the topic mentioned. The following figure shows that same graph with examples of direct links gold citations (Citation 1):



The blue-colored work would have "Biology" as the first citation feature, the purple-colored work would have "Stats" as the first citation feature, and the green-colored work would have "Medicine" as the first citation feature. The following figure demonstrates the Citation 2 feature:



The same colors are linked to the same topics as in the previous figure, the only difference being they would be considered in the second citation feature instead of the first since they are 2 edges away from the gold citation. To give a full example of how both citation features would be created for a single work, take a look at the following example:



In this example (blue-colored work) there would be both a Citation 1 and Citation 2 feature. The Citation 1 feature would be “Stats” while the Citation 2 feature would be “Medicine”. In practice, a work may have citations but not have any citation features because it does not link to a gold citation. Also, works could have a citation feature that contains links to many different citations. Through testing, we found that if we truncated the feature length to 16 and 128 for Citation 1 and 2, respectively, we keep over 95% of the gold citation data for works.

Feeding Citation Features to a Model

Once we have determined the list of gold citations for Citation 1 and Citation 2, we end up with a list like the following:

-> **Citation_1**: *[“Statistics”, “Medicine”]*
-> **Citation_2**: *[“Medicine”, “Statistics”, “Machine Learning”, “Biology”]*

The next question is: how will these features be fed to a model? There are several options:

- Take each topic name and run through an embedding model (similar to journal name) and then feed that embedding to the model
- Take each gold citation, assign it an integer, and feed that list of integers to an embedding layer in the model
- Take each gold citation, map back up to the topic it belongs to, assign each topic an integer, and then feed that list of integers to an embedding layer in the model
- Create a static embedding of each topic by getting the average embedding of the topic/abstract of many papers within that topic and feed that embedding to the model

For the options where we feed the data to an embedding layer in the model, we then have a choice to make whether you want the embeddings to be static or learned. After experimenting with different options, we decided to map each gold citation back to its topic, assign each topic an integer, and then feed that list of integers to an embedding layer. We also decided to use learned embeddings for each topic so that through training, the model can update the embedding to a representation that makes the most sense for that topic. These decisions were made for the following reasons:

1. It avoids the sparsity issue that would occur if we used the gold citations as the features
2. It allows the model to learn its own embedding representation of the input features so that it can encode useful information in that input
3. Because gold citations for a given topic would most likely have similar embeddings anyways, we most likely would not gain much additional information by allowing them to be created separately
4. A static embedding (from running the topic name through an embedding model) might not be representative of the topic
5. Creating static embeddings wouldn’t allow our model to encode as much information about the citation features as it trains

Model Methodology and Training

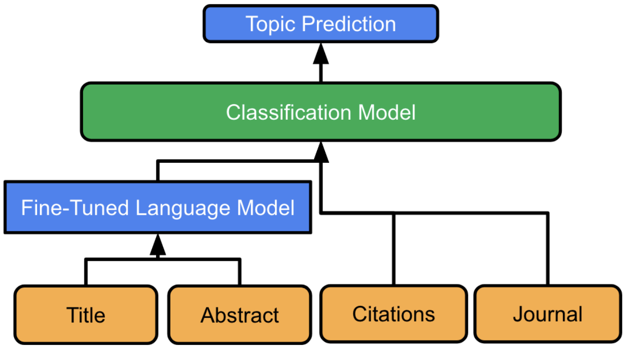
With the features processed, the final topics model was created by fine-tuning a title/abstract only language model on HuggingFace and then incorporating the outputs of that model into a smaller neural network that also incorporated the journal and citation features (see Figure).

Title/Abstract Only Model

We use a pre-trained model on HuggingFace, taking advantage of the work others have already done to build a multilingual language model. The multilingual model, as opposed to English-only models, gives up a little accuracy for English works in order to get better accuracy for additional languages. While the model could still improve when it comes to non-English works, we are happy with the results. In the future, if we are able to train the model on a larger percentage of non-English works, it could perform better in a live prediction setting. That being said, we decided to fine-tune the multilingual BERT (mBERT) model.

Full Topics Model

After we fine-tuned the language model, we incorporated the model and output into a separate neural network that was able to combine all features in order to make a prediction. A rough outline can be seen as follows:



Experiments were done to determine structure of the neural network as well as the hyperparameters which would give the best results during training. As always, we had to balance the desire to increase the performance of the model (by changing certain parameters) with our requirement that the final model be deployable for real-time inference.

Model Training

After finalizing the architecture and preprocessing all training data, we used a multi-GPU AWS EC2 machine (p3.16xlarge) to perform both parts of the model training. To start, the mBERT model was fine-tuned on the title/abstract. Then, that model and its outputs were integrated into a separate neural network which would be the final topics model. During the training for the final model, some features were held out in order to simulate how the real data would look in production. Most of the labeled data we had included citations but in reality, only about a third of our historical data contained citations. Therefore, we wanted to randomly hold out citations during training so that the model got used to seeing data without citations. We also randomly removed the other features as well so that in training, the model sometimes only had one feature to go on. This was done to improve model performance on data with missing features.

Model Performance

After training the model, extensive testing was done to determine if the model was sufficiently trained in order to accurately predict in a live setting. To give an idea of overall performance, the following table shows the TopK accuracy (% of samples where the label showed up in the top K values of the prediction) on the test set for different values of K:

k =1	0.53
k = 2	0.58
k = 5	0.67
k = 10	0.73

We also looked at the accuracy of the model for varying amounts of input data missing to see how the model might perform in the best case scenario (all input features are available to the model) vs other scenarios (some of the features are not available):

Title Available	Abstract Available	Citation 1 Available	Citation 2 Available	TopK=1 Accuracy
No	No	No	No	0.01
No	No	No	Yes	0.56
No	No	Yes	Yes	0.66
Yes	No	No	No	0.20
Yes	No	No	Yes	0.47
Yes	No	Yes	Yes	0.68
Yes	Yes	No	No	0.30
Yes	Yes	No	Yes	0.55
Yes	Yes	Yes	Yes	0.72

As expected, the more data our model has, the higher the accuracy. The top row in the table shows that when the main features are missing (i.e., only a journal title is available or no features are available), the model struggles to figure out the topic, although it still does better than a random guess. It is fairly obvious that the citations play a large role in the predictions, which is to be expected considering the labeled data was created by clustering citations.

Another observation from looking at the predicted data is that there is a trend for model accuracy based on the cluster ID number (since cluster ID was assigned in order of labeled data count, as explained in the data exploration section):

Topic ID Range	TopK=1 Accuracy

1 - 453	0.53
454 - 905	0.57
906 - 1357	0.59
1358 - 1809	0.59
1810 - 2261	0.58
2262 - 2713	0.56
2714 - 3165	0.48
3166 - 3617	0.42
3618 - 4069	0.36
4070 - 4521	0.33

From looking at the table above, we can see that there is an optimum cluster ID range for predicting the correct label. We think that the larger clusters at the lowest ID numbers are too big and may be hard to predict because of a lack of a coherent identity. However, for the highest ID numbers (i.e., where the labeled topic counts were lower) we also see a drop in accuracy which we believe is due to clusters that are less well-defined. In the future, it might be advantageous to cluster in 2 steps where the first step will have a higher paper count threshold and then for some of the largest clusters, splitting those up into smaller, more well-defined groups.

While it was important to look at accuracy metrics to confirm that the model was performing well, it was even more important to sample data from both the test set as well as data that wasn't used in training to make sure that predictions make sense. For this exercise, we sent the data through the model for predictions and then looked at the top 5 predictions compared to the label. This allowed us to see where our model was predicting different topics than the label and make a determination if the model was correct in doing so. We will list some examples here to show what we were looking for and how we determined the model was successful. For the predicted topics, the score next to the topic name is the score that came directly from the model output.

Example 1

OpenAlex Work ID: 183088995

Title: Multidisciplinary Team Approach to Cleft Lip and Palate Management

Abstract: None

Labeled Topic: 11374 (Genetic and Environmental Influences on Cleft Lip and Palate)

Predicted Topics:

- 1: 11374 (Genetic and Environmental Influences on Cleft Lip and Palate - 1.0)
- 2: 11679 (Genetics and Development of Craniofacial Abnormalities - 0.998)
- 3: 11310 (Genetic and Molecular Studies of Connective Tissue Disorders - 0.978)
- 4: 13107 (Genetic and Developmental Studies of Limb Malformations - 0.966)
- 5: 11260 (Clinical Management of Tracheal and Airway Disorders - 0.958)

This was a pretty easy example, even with the abstract missing. The model accurately predicted the topic and even the other predicted topics in the top 5 are closely related to the paper.

Example 2

OpenAlex Work ID: 2769228014

Title: Mitverbrennung von Sekundärbrennstoffen

Abstract: None

Labeled Topic: 13603 (Corporate Social Responsibility and Sustainability in Business)

Predicted Topics:

- 1: 12565 (Fluid Dynamics and Engineering Applications - 0.96)
- 2: 12959 (Advanced Lightweight Materials and Engineering Applications - 0.756)
- 3: 13990 (Advancements in Automotive Engineering and Fuel Technology - 0.575)
- 4: 10511 (Advances in Nanocomposite Dielectric Materials and Insulation - 0.547)
- 5: 10663 (Lithium-ion Battery Management in Electric Vehicles - 0.459)

This is an example of what our model is potentially capable of with regards to non-English works. The title is in German and translates to "Co-combustion of secondary fuels". The label is not good, but through training, the model seems to have predicted correctly. With a shorter title in a non-English language with no abstract, this was certainly a good sign for our final topics model.

Example 3

OpenAlex Work ID: 2155331734

Title: Novel, Potentially Zoonotic Paramyxoviruses from the African Straw-Colored Fruit Bat Eidolon helvum

Abstract: Bats carry a variety of paramyxoviruses that impact human and domestic animal health when spillover occurs. Recent studies have shown a great diversity of paramyxoviruses in an urban-roosting population of straw-colored fruit bats in Ghana. Here, we investigate this further through virus isolation and describe two novel rubulaviruses: Achimota virus 1 (AchPV1) and Achimota virus 2 (AchPV2). The viruses form a phylogenetic cluster with each other and other bat-derived rubulaviruses, such as Tuhoko viruses…

Labeled Topic: 11135 (Paramyxovirus Infections and Epidemiology)

Predicted Topics:

- 1: 11135 (Paramyxovirus Infections and Epidemiology (1.0)
- 2: 12047 (Viral Hemorrhagic Fevers and Zoonotic Infections - 0.999)
- 3: 11780 (Rabies Virus Transmission and Control- 0.997)
- 4: 10166 (Global Impact of Arboviral Diseases - 0.996)
- 5: 11581 (Ebola Virus Research and Outbreaks - 0.972)

In Example 3, the model predicted the correct topic for the work. Additionally, the second predicted topic could have also been a good choice.

Example 4

OpenAlex Work ID: 1590985406

Title: Bioinformatics basics: applications in biological science and medicine

Abstract: Contents BIOLOGY AND INFORMATION Bioinformatics-A Rapidly Maturing Science Computers in Biology and Medicine The Virtual Doctor Biological Macromolecules as Information Carriers Proteins: From Sequence to Structure to Function DNA and RNA Structure DNA Cloning and Sequencing Genes, Taxonomy, and Evolution BIOLOGICAL DATABASES Biological Database Organization Public Databases Database Mining Tools GENOME ANALYSIS The Genomic Organization…

Labeled Topic: 13280 (Role of Hackathons in Biomedical Engineering Education)

Predicted Topics:

- 1: 13937 (Challenges and Innovations in Bioinformatics Education - 0.997)
- 2: 10885 (Microarray Data Analysis and Gene Expression Profiling - 0.848)
- 3: 10887 (Analysis of Gene Interaction Networks - 0.823)
- 4: 13984 (Nutritional Genomics: Personalized Nutrition and Health - 0.771)
- 5: 11710 (Biomedical Ontologies and Text Mining - 0.761)

Example 4 showed another case where the labeled topic was not the best choice and our model did a good job of selecting a more appropriate topic. Examples like this are why we use prediction accuracy as an indicator of model performance but not a final determination. Extensive sampling from the test data is how we determined that this model was good enough to deploy.

That being said, our model does not always make a good prediction. Even though we use a multilingual model and make predictions using citation features, there are still times when the model does not seem to understand a language other than English. The number is smaller than if we had used a primarily-English model but there are still errors that we believe could be fixed with a better multilingual model. In addition, there are other times where the model does not do a good job at assigning For example:

Example 4

OpenAlex Work ID: 3187336104

Title: Machine Learning Approach to Predict SGPA and CGPA

Abstract: The prediction of SGPA and CGPA is beneficial to university students. Students will easily get an estimate of their final outcome from this project. As a result, the students will be able to brace themselves for a successful outcome. Students pass the day by participating in a variety of events. Students use social media sites such as Facebook, Instagram, and Twitter. They engage in various hobbies such as playing mobile games, listening to music, among others. As a result, they were able to move several times with these tasks. As a result, if a student spends so much time doing any of those things, she will not be able…

Labeled Topic: 11490 (Hydrological Modeling using Machine Learning Methods)

Predicted Topics:
- 1: 10273 (Internet of Things and Edge Computing - 0.987)
- 2: 12406 (Smart Vehicle Safety and Monitoring Systems - 0.958)
- 3: 13038 (Applications and Challenges of IoT - 0.922)
- 4: 11814 (Design and Control of Warehouse Operations - 0.899)
- 5: 10444 (Activity Recognition in Pervasive Computing Environments - 0.898)

The predicted topics are pretty much completely unrelated to the title and abstract. Upon further review of why these predictions seem so off, we found that most of the citations for the paper talk about edge computing and IoT. It seems like a strange set of citations for a paper about using machine learning to predict SGPA and CGPA but regardless, these predictions do not pass the "eye test".

The model also struggles with lower amounts of data, which was expected. We have set thresholds for abstract and title length to try to combat this phenomenon. For the title, we set the minimum character threshold to 4 characters. For the abstract, we set a minimum threshold of 30 characters. Also, we do not predict on a work unless it has at least a title, abstract, or citation feature available. While we found that the journal name feature gave some signal to the model, it was not enough to accurately predict a topic if no other feature is available. Lastly, we set a model probability score threshold of 0.04 in order to limit predictions where the model is uncertain which Topic should be assigned.

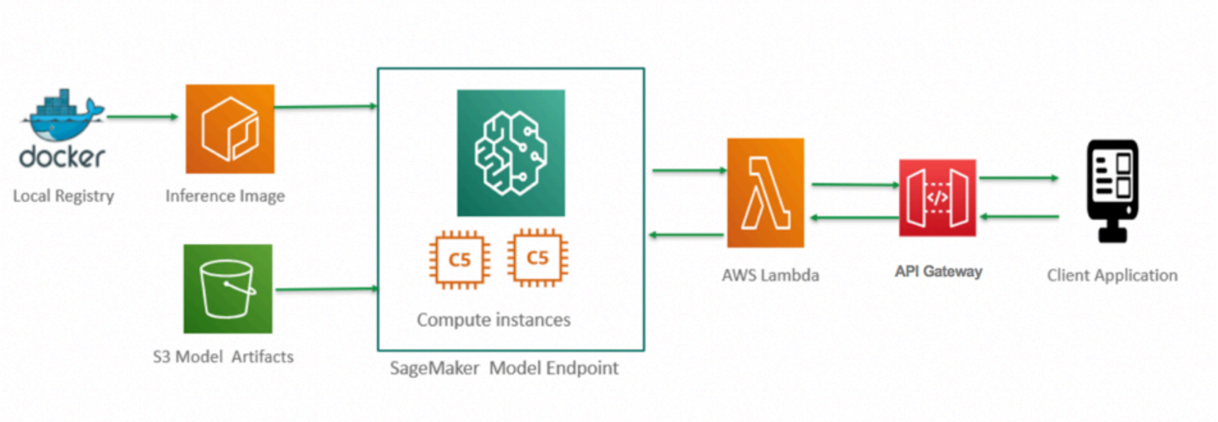
At the time of running all works through the new Topics model, there were around 247 million works in OpenAlex. The following table shows how many works were classified with at least one Topic and how many works were excluded from Topic classification for various reasons:

Subset of OpenAlex Data	Count	Topics available?
Title, abstract, or citations available (score > 0.04)	207,604,236	Yes
Only journal available	3,585,506	No
Title, abstract, journal, and citations not available	13,772,352	No
Title, abstract, or citations available (score <= 0.04)	22,125,755	No

Since this is a new model and a new system being applied to a large dataset, there will surely be other errors found in the predictions that we did not see in testing the model. In addition, we expect that over time there will be changes to the topic names in order to better reflect the papers contained within each topic. We will learn from any mistakes we find and improve the model/data for the future so these problems do not persist. But overall, this model seems to do an effective job of predicting topics and we believe that these topics will become an essential part of OpenAlex.

Deploying on AWS

In order to effectively use the model in production, AWS was chosen as a service provider to host the model and API endpoint. The following is an architecture diagram to show which services were used and how everything is connected.



The steps taken to deploy the model are as follows:

1. Use Docker to create a container which contains the inference code as well as the serving code used by AWS SageMaker
2. Upload the newly created image to AWS ECR (Elastic Container Registry)
3. Upload the model artifacts to AWS S3
4. Create an AWS SageMaker endpoint using the image and model artifacts

5. Use AWS Chalice to quickly create a REST API using AWS API Gateway and AWS Lambda

Once these steps were completed, requests could be made to the API to assign topics to papers using the model. Different configurations can be set up in AWS SageMaker to scale appropriately with the number of requests being made to the API.

Testing Model Throughput and Latency

Thorough API testing was done in order to figure out how many works could be predicted in a live setting. We performed load testing with Locust (docs) in order to see how many requests we could make each second for different configurations in Sagemaker. Previous models (institution tagging and concept tagging) make use of CPU instances on Sagemaker so this setup was replicated in order to reduce the time it will take to deploy the model. For now, we use ml.c6i.2xlarge instances, which give us 3-4 predictions per second for each instance. This will allow us to scale up to 8-10 instances to give us 30-40 predictions per second which is sufficient for the amount of data we will be processing each day. If needed, we could scale up higher to accommodate higher loads.

Summary

We have described the process of creating the topic classification model that is currently being used in OpenAlex to assign topics to papers. Our model makes use of the title, abstract, journal, and citation graph in order to determine the best possible topics to assign to a work. For any questions, feel free to reach out to support@openalex.org. All of the code used to create this model can be found on GitHub at: https://github.com/ourresearch/openalex-topic-classification.

References

Priem, J., Piwowar, H., & Orr, R. (2022). *OpenAlex: A fully-open index of scholarly works, authors, venues, institutions, and concepts* (arXiv:2205.01833). arXiv. https://doi.org/10.48550/arXiv.2205.01833

Sinha, A., Shen, Z., Song, Y., Ma, H., Eide, D., Hsu, B.-J. (Paul), & Wang, K. (2015). *An Overview of Microsoft Academic Service (MAS) and Applications*. 243–246. https://doi.org/10.1145/2740908.2742839

Traag, V. A., Waltman, L., & van Eck, N. J. (2019). From Louvain to Leiden: Guaranteeing well-connected communities. *Scientific Reports*, *9*(1), Article 1. https://doi.org/10.1038/s41598-019-41695-z

Van Eck, N. J. (2024). *Classification of research publications based on data from OpenAlex* (Version 2023nov) [dataset]. Zenodo. https://doi.org/10.5281/zenodo.10560276

Waltman, L., & van Eck, N. J. (2012). A new methodology for constructing a publication-level classification system of science. *Journal of the American Society for Information Science and Technology*, *63*(12), 2378–2392. https://doi.org/10.1002/asi.22748