Asking Geoscience Questions Through a Hybrid Machine-Human Learning Approach

Seongjin Park[1], Mihai N. Ducea[1,2*], Barbara Carrapa[1], Mihai Surdeanu[1], Robert Hayes[1], Dan Collins[1]

[1]University of Arizona, Tucson, AZ 85721, USA

[2]Faculty of Geology and Geophysics, University of Bucharest, 010041, Bucharest, Romania

*, corresponding author: ducea@arizona.edu

Submitted to Journal of Geology: July 9, 2020

ABSTRACT

A common challenge in science is human capability to evaluate the real impact of an observation and dataset. In order to overcome this important limitation, we need to be able to review all the available data and interpretations and evaluate the global distribution of a specific process. The increasing amount of scientific publications prevents scientists from being able to keep up with all the available literature. This challenge prevents them from objective evaluation of the global impact of a certain process. We present here an application of Artificial Intelligence to geosciences: we conduct a systematic analysis of geoscience literature through a hybrid machine-human approach. Such applications are more common in other fields but are in their infancy in the geosciences because of various difficulties the machines encounter in parsing geologic literature. We describe here some of these limitations and how we overcame them. We then use this approach as an example: we ask whether climate is influenced by volcanism in the geological past. Our results show, as expected, that most analyzed literature in this experiment conclude that volcanism influences climate change in deep time. Similarly, any question of potential global significance can be posed as an interrogating technique for our vast and fast growing literature in the field of geosciences.

Keywords: machine learning, geosciences, hybrid approach, climate change, volcanism.

1. INTRODUCTION

One of the cornerstone theories in natural sciences, Darwin's evolutionism, states that the evolution of flora and fauna in the geologic past goes through temporally determined and irreversible extinctions corroborated with the development of new species. That theory has been vetted by innumerable observations and stands today because of that. However, most potentially groundbreaking hypotheses in natural sciences have a difficult time being resolved at global scales because of the complexity of observations. In order to test complex hypotheses at global scale we need to have an objective and global review of the scientific literature. This task has turned into a near impossible challenge in recent years due to the vast amount of scientific data that have been published, which exceeds human capacity for processing and interpretation. This is particularly problematic in multi-disciplinary fields such as Earth Sciences that require the interpretation of data and hypotheses on a global scale and over large time intervals. Whereas data pertaining to regional geology of a particular area can still be tracked by the interested geologist (the number of papers is still within reach of human processing), the merit of so many global scale interpretations and hypotheses put forward in this field in recent years is difficult to evaluate. Did erosion of Earth's surface increase globally since the Pliocene as some have suggested (Herman et al., 2013)? Did the Earth's continental crust get significantly thicker overall in the latest Precambrian (Balica et al., 2020)? These are just a couple of examples of far-reaching but hard to evaluate hypotheses in a science that increasingly requires ingestion of too much information at global scale and commonly need placed into a complex deep time-space framework which is essential to Earth Sciences.

To address this issue, we build a hybrid machine-human approach for the systematic analysis of scientific discoveries in geosciences. The proposed approach employs machine reading to ingest publications at scale to construct causal models that aggregate scientific discoveries. These models allow scientists to attempt a truly global understanding of science, which facilitates the identification of (apparent) contradictions in scientific findings, as well as "white spaces" in the research landscape. For this purpose, we developed an application to geoscience to demonstrate the potential of our proposed approach, to experiment with the limitations of this type of literature and how they can be overcome. The application investigates the hypothesis that there is a causal relationship between volcanism and climate change in the geologic record as seen through the lens of published literature. Specifically, we ask whether volcanism influences climate change in the deep time geologic archive. It is obviously a pretty simplistic question used to initiate the experiment described below.

2. SYSTEMATIC MACHINE REVIEW OF GEOSCIENCE DATA

Since there was no pre-built corpus for the geosciences task, we selected 1,157 papers from the Web of Science website. These papers were selected because they contained keywords relevant to the hypothesis at hand such as volcanism or magmatism, and climate change. We then randomly chose 200 papers and extracted the abstract, introduction, and conclusion sections from each paper to be manually annotated with information if they support or do not support the hypothesis. Note that for this work we assume that the authors' data, interpretations and conclusions are correct. The annotation task was conducted on FindingFive[1], an online experiment platform. The papers were placed into one of four classes: SUPPORT, NEGATE, NEGATE&SUPPORT and UNRELATED. The annotations for these four classes were collected by two of the co-authors of this effort.

Next, we implemented a natural language processing (NLP) component for geoscience that extracts two types of information. First, we contextualize individual publications by extracting and normalizing the geospatial and temporal contexts addressed in these papers (e.g., *Pliocene*, *4 million years ago*, and *Bering Sea*). Second, we built a document classifier that is trained to determine whether any given paper supports the hypothesis that "volcanism affected climate change", so that we could make a prediction on *new* papers. The results of these two components were aggregated into a publication knowledge base, which contains the publication itself, the prediction of the hypothesis classifier (SUPPORT, NEGATE, NEGATE&SUPPORT, and UNRELATED), the occurrence of geological eras and epochs (e.g., the frequency of *Pliocene* in a given paper), and the occurrence of geological locations (e.g., the frequency of *Africa* in a given paper). We used this knowledge base to visualize the evidence for the hypothesis investigated on the world map to identify global temporal and geospatial patterns.

[1] https://www.findingfive.com

3. THE HYBRID MACHINE-HUMAN APPROACH

Below, we detail the three key components of our hybrid machine-human approach in this experiment.

3.1. Contextualizing findings: Time and site identification

To analyze the relationship between volcanism and climate change at different times in the geological past and locations, we built a custom *Named Entity Recognizer* to extract spatial and temporal information from the analyzed text. Named entity recognition (henceforth, NER), which is also known as entity chunking or extraction, is a common NLP task which aims to identify named entities within the given text and classify or categorize those entities under various predefined classes. Our focus in this work is on the identification of locations and geological eras and epochs, which are necessary to contextualize the findings discussed in the papers.

Existing NER tools such as Stanford's CoreNLP (Manning et al., 2014) or spaCy (Honnibal & Montani, 2017) focus on generic locations, times, and dates rather than geoscience-specific ones. For example, when we fed the example sentence "Clay mineral assemblages and crystallinities in sediments from IODP Site 1340 in the Bering Sea were analyzed in order to trace sediment sources and reconstruct the paleoclimatic history of the Bering Sea since Pliocene (the last 4.3 Ma)." to the Stanford CoreNLP NER, the result is:

Clay mineral assemblages and crystallinities in sediments from IODP Site [1340]DATE in the [Bering Sea]LOCATION were analyzed in order to trace sediment sources and reconstruct the [paleoclimatic]MISC history of the [Bering Sea]LOCATION since Pliocene (the last [4.3]NUMBER Ma).

Even though the Stanford CoreNLP NER correctly identified "Bering Sea" as a LOCATION, it did not recognize geosciences-specific expressions, and, further, it classified expressions into the incorrect entity types. For example, IODP Site 1340 (IODP stands for Integrated Ocean Drilling Program) refers to a certain location, but the recognizer identified only "1340", and classified it as a DATE. The recognizer missed the term Pliocene, which means "the geologic timescale that extends from 5.333 million to 2.58 million years BP." "Ma" in geosciences articles usually means "million years ago", but the CoreNLP NER could not identify it as TIME.

To recognize expressions which were not identified by CoreNLP or Spacy, we used the Odin event extraction framework and rule language (Valenzuela-Escárcega et al., 2016); henceforth, Odin), and added custom rules to capture geoscience-specific expressions. In particular, we developed rules to capture:

Temporal information. As mentioned, initially we utilized the named entity recognition tool in Stanford's CoreNLP (Manning et al., 2015); henceforth, CoreNLP) to identify time information. However, since CoreNLP was trained on general text data, it does not recognize geological temporal expressions, such as Paleocene or Jurassic. In addition, in geosciences papers, there were abbreviations such as "M.y.r." and "M.a.", which mean "millions of years" (duration), and "million years ago" (absolute time). Thus, we wrote custom rules to recognize geological temporal expressions and built a custom time normalizer to convert actual times (e.g., "170 M.y.r.", or "1.5 million years ago") to relevant temporal expressions (e.g., Jurassic, Quaternary) (Supplementary Document 1).

Site information. Similar to temporal information, there were domain-specific spatial expressions that could not be captured by existing NERs (e.g., Stanford CoreNLP). Further, some of these expressions did not have any information about the actual locations that they indicate. Thus, we wrote scripts to extract spatial expressions, disambiguate geoscience-specific spatial expressions (e.g., "*IODP Site U1360"*), and normalize these expressions to specific latitude-longitude bounding boxes (Supplementary Document 2).

3.2 Classifying the hypothesis of interest

Even though these spatial and temporal expressions are important to contextualize the findings of a publication, they provide no information on our key hypothesis, whether volcanism affected climate change. To make a prediction whether the given paper supports or negates the relationship between volcanism and climate change, it is necessary to build a machine learning classifier that infers if the hypothesis is supported (or not) from the text of these publications.

Among the wide variety of text classification methods, we experimented with Naïve-Bayes (Raschka, 2014), and Support Vector Machines (Cortes & Vapnik, 1995). Naïve-Bayes is a *probabilistic* classification algorithm that learns from the observation that there are certain words, or word sequences, which occur more in one type of text than another (e.g., "CO2" would appear more in texts that support the hypothesis that volcanism impacts climate change). Support Vector Machines (SVMs) are *geometric* learning algorithms that find separating hyperplanes between classes of documents such that most documents belonging to one class are located on one side of the hyperplane. More recently, Wang & Manning (2012) proposed Naïve-Bayes SVMs (NB-SVMs), which combine the two ideas into a unified classification algorithm.

Even though neural network models have shown good performance on text classification (Convolutional Neural Network; Kim, 2014, and Long Short-Term Memory Networks, Liu et al., 2016), the disadvantage of using deep neural network models is that it is hard to interpret why the model made a certain prediction, which is the reason why the neural network models are often called "blackbox". Since it was important to understand whether the volcanism-related words, temporal expressions, or climate-related words had any effect on making predictions, in the current project we decided to use SVM and NB-SVM classifiers instead of neural network models. Document classification routines are detailed in Supplementary Document 3.

3.2.1. Data annotation

Data annotation was performed via FindingFive. 200 papers were randomly chosen from the set of 1,157 downloaded papers, and then, title, abstract, introduction, conclusion/discussion sections of 200 papers were presented to annotators. After reading the provided text, annotators determined whether the given paper supported or negated the relationship between volcanism and climate change. As a result, we produced 400 annotation results (200 papers × 2 annotators). To measure the agreement between annotators, Cohen's kappa score was measured. The Kappa result was 0.523, which showed moderate agreement between annotators. This is to be expected for such a complex hypothesis.

3.2.2. Classification of results

We evaluated the quality of the proposed classifiers that were trained on the annotations by comparing micro-F1 score calculated using 10-fold cross validation. To be specific, we collected the algorithm's predictions on each test partition, and calculated micro-F1 score from all these predictions.

In these experiments, we observed that the NB-SVM classifier outperformed slightly the SVM classifier, but both performed reasonably well, at a micro-F1 score of over 83%. To take advantage of both classifiers, we built an ensemble model that lets both classifiers vote on what the final classification decision should be. In particular, we used the following criteria:

1. When the predictions from both models are the same (e.g., NEGATE and NEGATE), then that label (e.g., NEGATE) becomes the final output.

2. When the predictions from the two models are different and one of the predictions is UNRELATED (e.g., SUPPORT and UNRELATED), then the prediction which is not UNRELATED becomes the final output (e.g., SUPPORT).

3. When the predictions from the two models are different and neither of them is UNRELATED, then choose the prediction from NB-SVM.

The performance of the ensemble model was slightly higher than that of the individual models. For example, the micro-F1 score of the ensemble model was 83.99%. For this reason, we used this ensemble method to classify all remaining papers in the collected dataset on whether they support/negate or are unrelated to the hypothesis at hand.

3.3. Aggregation of results for visualization

With the two components described above that: (a) place a scientific finding in its proper geospatial and temporal context, and (b) identify if publications support or the hypothesis at hand, we can aggregate and visualize results at scale. To further simplify the visualizations, we used the *geopy*[2] Python library to convert IODP sites to latitudes and longitudes, and we converted the identified specific geological Periods and Epochs into broader (larger time intervals) geological eras. For each paper analyzed, we used the most frequent top k (where k = 1, or k = 3) spatial and temporal entities for context.

[2] https://pypi.org/project/geopy/

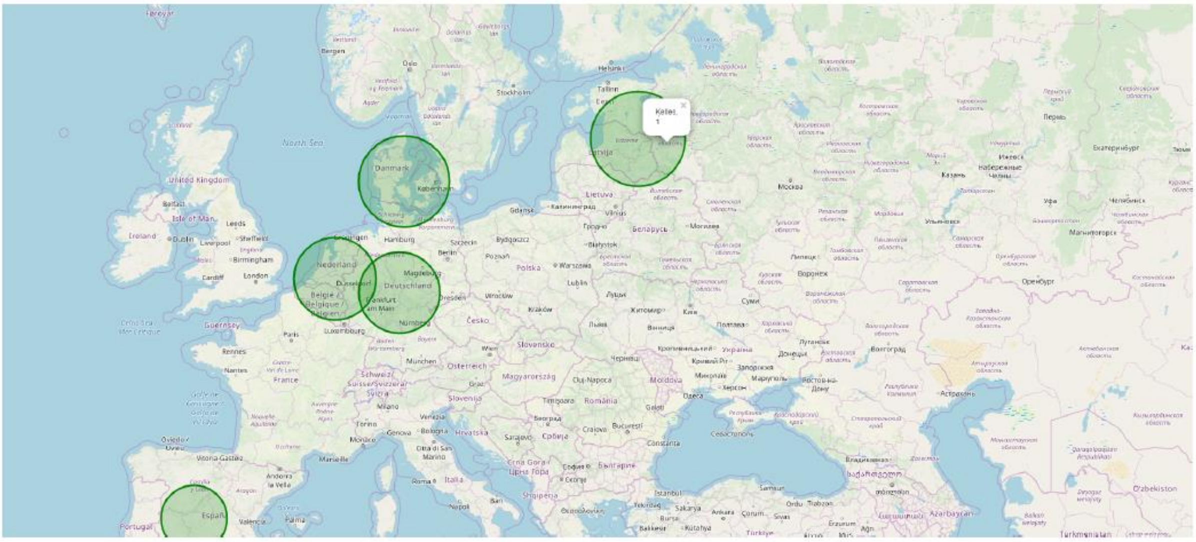

Figure 1. Top-1 map during Cenozoic (Europe): Circles represent the most frequent location found in each paper where the relationship between volcanism and climate change has been tested during Cenozoic. Green circles indicate the locations where the impact of volcanism on climate change was verified.

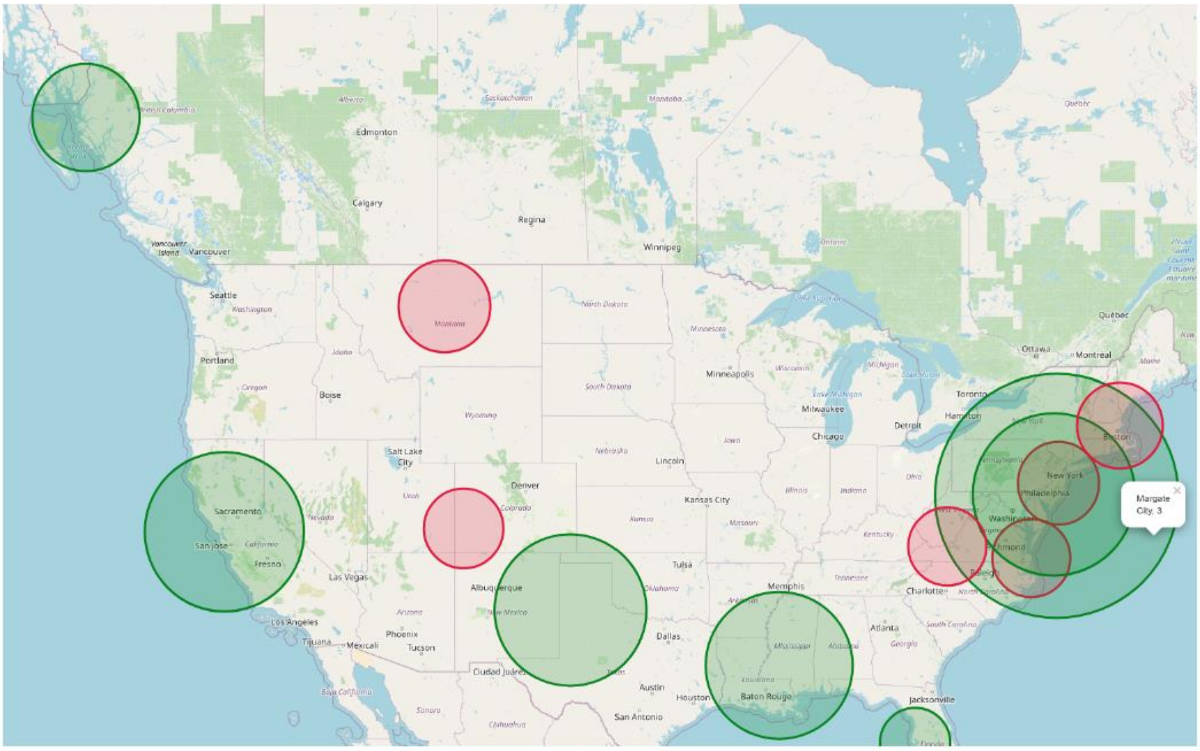

Figure 2. Top-3 map during Cenozoic (North America): Circles represent the top three most frequent locations found in each paper where the relationship between volcanism and climate change has been tested during Cenozoic. Green circles indicate the locations where the impact of volcanism on climate change was verified, and red circles indicate the locations where previous research negated the relationship between volcanism and climate change.

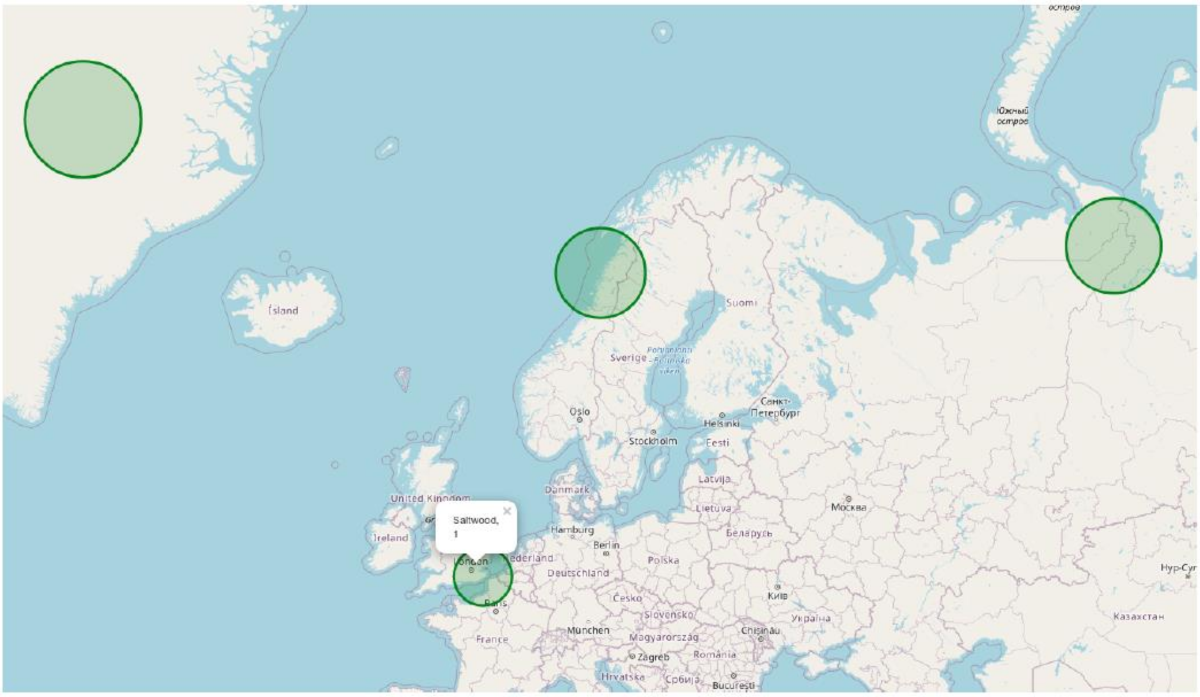

Figure 3. Top-1 map during Phanerozoic (Europe): Circles represent the most frequent location found in each paper where the relationship between volcanism and climate change has been tested during Phanerozoic. Green circles indicate the locations where the impact of volcanism on climate change was verified.

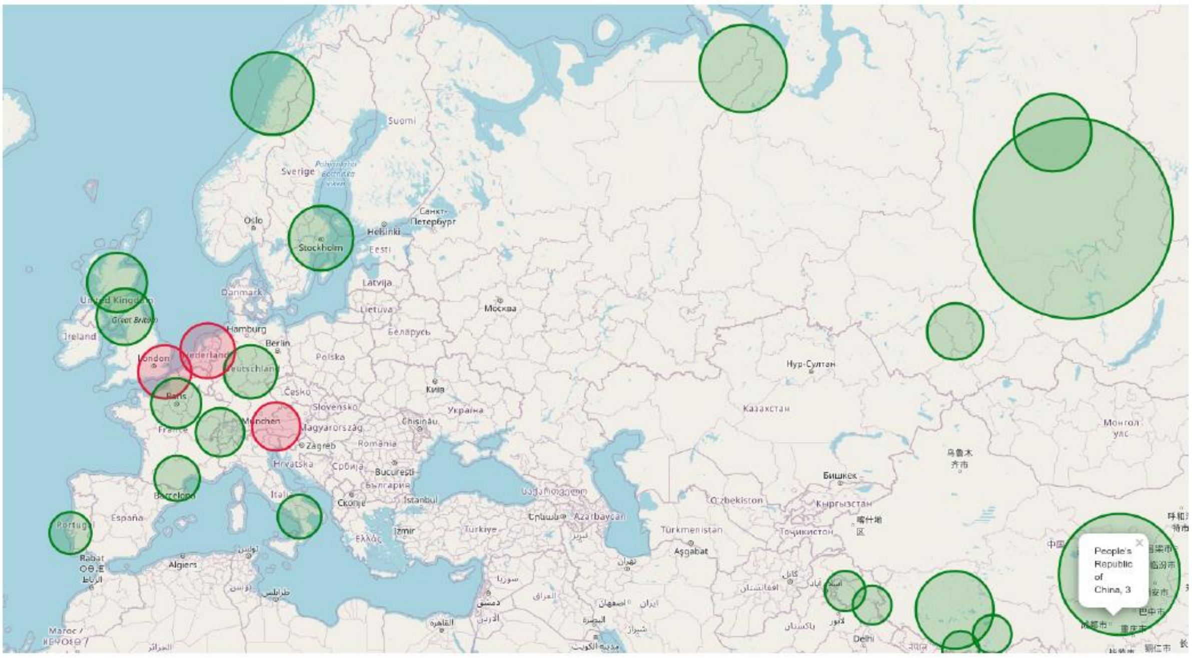

Figure 4. Top-3 map during Phanerozoic (Europe and Asia): Circles represent the top three most frequent locations found in each paper where the relationship between volcanism and climate change has been tested during Cenozoic. Green circles indicate the locations where the impact of volcanism on climate change was verified, and red circles indicate the locations where previous research negated the relationship between volcanism and climate change.

Figures 1 to 4 show several visualizations of the results, with green indicating support for the hypothesis, and red negating the hypothesis. The sizes of the circles were determined based on the number of papers that the classifier predicted the corresponding label (i.e., green for SUPPORT, and red for NEGATE). Figure 1 shows the most frequent locations during Cenozoic in Europe, and Figure 2 shows top three most frequent locations during Cenozoic in North America. When manually inspecting the results, we observed that 11 out of 17 data points within the North American continent were correctly identified and visualized on the world map. One red circle (i.e., the corresponding paper was classified as not supporting the hypothesis) was incorrectly predicted when the actual paper was unrelated with respect to the hypothesis. Further, 4 data points were from simulation papers, and 2 data points were based on incorrect predictions.

These figures immediately highlight several important observations:

- Following the adage that "a picture is worth a thousand words", we argue that a good visualization can summarize a thousand papers. Our visualizations allow the scientist to quickly draw important conclusions that would not be easily available otherwise. For example, our figures show that while the majority of publications support the hypothesis investigated that volcanism impacts climate change, not all do.

- Similarly, this bird's-eye-view of a scientific question allows one to quickly identify "white spaces" in research, i.e., topics that are insufficiently investigated. For example, our visualizations show that while empirical evidence for our hypothesis is well represented for the North American continent, it is scarce in other continents.

- Lastly, this work allows one to identify (potential) contradictions in scientific findings quickly, which provide opportunities for better science. For example, Figure 2 shows apparent contradictions in findings from the East coast of the North American continent in the Cenozoic.

4. Conclusion

The result of this preliminary work introduced a methodology to automatically provide an objective and global review of the geoscientific literature, and to evaluate the impact of specific hypotheses, in this case the causal relationship between volcanism and climate change. We show the promises and limitations of this approach to geoscience literature with this admittedly simplistic example. This approach helps us process and interpret a large amount of scientific papers that have been published, without the need for human annotators to invest time in reading and parsing all these papers. In addition, with the visualization, researchers are able to investigate chronological changes of the relationship between volcanism and climate change. This approach could be expanded to any number of queries in the geoscience literature for the systematic analysis of various hypotheses and ideas by examining a large body of previously published papers. Results can be further plotted on reconstructed various sample or study locations using paleogeographic maps.

It is vital to emphasize that the propose methodology is hybrid, requiring direct collaboration between humans and machines. For example, geoscientists were required to provide training data for our hypothesis classifier. Further, as discussed, our resulting classifier is only approximately 80% accurate, which means that, in order to improve it, it needs continuous feedback from the scientists using it. Longer term, we envision a community-wide effort in which such classifiers are created and deployed in the cloud to mine an arbitrary number of hypotheses, and are continuously improved over time by their human end users.

Acknowledgments. M.N.D. acknowledges support from the Romanian Executive Agency for Higher Education, Research, Development and Innovation Funding project PN-III-P4-ID-PCCF-2016-0014.

References

Balica, C., Ducea, M. N., Gehrels, G. E., Kirk, J., Roban, R. D., Luffi, P., Chapman, J. B., Triantafyllou, A., Guo, J., Stoica, A. M., Ruiz, J., Balintoni, I., Profeta, L., Hoffman, D., Petrescu, L. 2020. A zircon petrochronologic view on granitoids and continental evolution. Earth and Planetary Science Letters, 531, paper 11605

Cavnar, W. B., Trenkle, J. M., & Mi, A. A. (1994). N-Gram-Based Text Categorization.

Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, 161–175.

Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, *20*(3), 273–297.

Herman, F., Seward, D., Valla, P.G., Carter, A., Kohn, B., Willett, S.D., Ehlers, T.A., 2013, Worldwide acceleration of mountain erosion under a cooling climate, Nature, v. 504.

Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 1746–1751.

Liu, P., Qiu, X., & Huang, X. (2016). Recurrent Neural Network for Text Classification with Multi-Task Learning. *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, 2873–2379.

Honnibal, M., & Montani, I. (2017). spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.

Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., & McClosky, D. (2014). The Stanford CoreNLP Natural Language Processing Toolkit. https://doi.org/10.3115/v1/p14-5010

Raschka, S. (2014). *Naive Bayes and Text Classification I - Introduction and Theory*. Ithaca: Cornell university library.

Valenzuela-Escárcega, M. A., Hahn-Powell, G., & Surdeanu, M. (2016). Odin's Runes: A rule language for information extraction. *Proceedings of the 10th International Conference on Language Resources and Evaluation, LREC 2016*.

Wang, S., & Manning, C. D. (2012). Baselines and bigrams: Simple, good sentiment and topic classification. In *50th Annual Meeting of the Association for Computational Linguistics, ACL 2012 - Proceedings of the Conference*. https://doi.org/https://dl.acm.org/doi/10.5555/2390665.2390688

Supplemental Document 1 for Park et al., 2020

Temporal Expression Normalization

To convert mentions of temporal expressions (i.e., names of geological eras or epochs) to temporal intervals, we created a spreadsheet that contains the relations between date intervals and these temporal expressions. The file contains the name of the geological time era (e.g., *Jurassic*) and the time period (e.g., from 201.3 million years ago to 145 million years ago). The following table shows a subset of this spreadsheet:

Era/Epoch	From	To
Eoarchean	4,000,000,000	3,400,000,000
Paleoarchean	3,400,000,000	3,200,000,000
…	…	…
Jurassic	201,300,000	145,000,000
…	…	…

Table 1. A subset of the spreadsheet file that map names of geological eras/epochs to actual time intervals.

Algorithm 3: Sample rules for temporal expressions

```
name: time-period-1
    priority: 1
    label: TempExpr
    type: token
    pattern: |
        /Z?(Tertiary|Maastrichtian|Danian|
        Guadalupian|Triassic|Cenomanian|
        Cretaceous|Paleogene|Palaeocene|
        Pliocene|Pleistocene|Holocene|
        Zanclean|Cambrian|Paleozoic|
        Palaeozoic|Ordovician|Neogene|
        Phanerozoic|Silurian|Devonian|
        Carboniferous|Permian|Neoproterozoic|
        Mesozoic|Quaternary|Precambrian|
        Jurassic)/
name: time-period-2
    priority: 1
    label: TempExpr-Ago
    type: token
    pattern: |
    entity=/^NUM/+
      /(ma|myr|mya|ka|Ma|Myr|Mya|Ka|m.y.r)/
```

We extracted temporal expressions from text using two Odin rules, listed in Algorithm 3. The first rule (*time-period-1*) captures names of known geological epochs and eras. Note that, since the publications mined were automatically converted from PDF files to text files using Science-Parser[1], the result text files often had spelling mistakes. This rule captures the most common ones. The second rule (*time-period-2*) captures numeric temporal expressions such as *500 mya*, using common temporal abbreviations in geoscience papers.

[1] https://github.com/allenai/science-parse

Algorithm 4: Normalize the temporal expression and calculate the frequency

```
counter = count frequency of temporal expressions
for sentence in documents do
    TempExpr = [list of TempExpr in the sentence]
    Ago = [list of TempExpr-Ago in the sentence]
    for expr in TempExpr do
        counter[expr] += 1
    for expr in Ago do
        if expr starts with "M or m" then
            actualTime = NUM × 1000000
            timeExpr = find era using actualTime
            counter[timeExpr] += 1
        if expr starts with "K or k" then
            actualTime = NUM × 1000
            timeExpr = find era using actualTime
            counter[timeExpr] += 1
```

After capturing temporal expressions using the two rules summarized above, we used an additional script to convert and normalize the actual times to the corresponding geological times. The process is listed in Algorithm 4. For example, when one sentence contained a phrase *150 million years ago* or *150 m.y.r*, the script first converts the temporal expression to the time (in years) *150,000,000*, and then normalizes it to *Jurassic* using the spreadsheet listed in Table 1. After that, we counted the occurrence of geological eras/epochs in the document for later use, in the visualization. The following output shows an example of the statistics acquired from one paper, where lines 3 – 4 show the frequency of geological eras that occurred in the target paper.

```
1 "synthetic_data_on" : {
2 "time" : {
3       "Neogene" : 3,
4       "Pliocene" : 2    }
```

Supplemental Document 2 for Park et al., 2020

Spatial expression normalization

The second critical component necessary for the contextualization of geoscience results (in addition of the recognition of temporal expressions) handles the identification and normalization of location expressions. Similar to the recognition of temporal expressions, there are domain-specific spatial expressions that are not captured by existing Named Entity Recognition (NER) tools (e.g., Stanford CoreNLP). Further, some of these expressions (i.e., all IODP sites) do not contain direct information about the actual locations that they indicate. Thus, we wrote scripts to extract spatial expressions, disambiguate geoscience-specific spatial expressions (e.g., *IODP Site U1360*), and normalize those expressions. In this section, we will provide the algorithms used for site identification and normalization.

Recognition of location expressions

First, we applied the named entity recognizer in Stanford CoreNLP to check how many spatial expressions it recognizes. CoreNLP captures most of the well-known locations, such as *Bering Sea* or *Aleutian Islands*, but it does not recognize geoscience-specific locations (e.g., *IODP Site U1360* or *Deccan Traps*). To quantify these errors, we analyzed the annotation results from 100 sample documents using CoreNLP.

Algorithm 5: Named Entity Recognition using Stanford CoreNLP

```
for sentence in document do
    words = [word in sentence]
    for word in words do
        entity = Recognizer(word)
        if entity == "Location" then
            return entity
```

For this analysis, we used Algorithm 5 to deploy Stanford's CoreNLP to recognize named entities in a given sequence of words. In particular, the document was tokenized into sentences, and then, each sentence was split into words using the word-tokenizer in the CoreNLP package. Next, the recognizer processes each sentence, and returns named entity categories (*Location, Person, Organization, Number, Date, Miscellaneous*) when the input word is (part of) a named entity, or *O* otherwise.

Our analysis indicated that CoreNLP does recognize: (1) specific geological locations (e.g., DSDP Site, IODP Site), (2) Traps[1], and (3) other specific locations that do not usually appear in general, open-domain texts. In addition, since the data were text files converted from PDF files, there were some

[1] Here, *Trap* means a structural trap, which is a type of geological trap that forms as a result of changes in the structure of the subsurface, due to tectonic, diapiric, gravitational and compactional processes.

misspelled words which made them unrecognizable.

To compensate for these limitations, we wrote a series of custom Odin rules to capture the above geological locations that are missed by this general-purpose tool. These rules are listed in Algorithm 6.

Algorithm 6: Rules for geological sites

```
name: geo-site-Site
    priority: 1
    label: SpatialExpr
    type: token
    pattern: |
        /DSDP/ /Site/ /U?[0-9]+[A-Z]?/
        |
        /IODP/ /Site/ /U?[0-9]+[A-Z]?/
        |
        /Site/ /U?[0-9]+[A-Z]?/

name: geo-site-Name
    priority: 1
    label: SpatialExpr
    type: token
    pattern: |
        /(Deccan|ParanaEtendeka|Karoo|Siberian)
          (Traps)?/
        |
        /(?i)flood/ /(?i)basalts?/
        |
        /Stevns/ /Klint/
        |
        /Tethyan/
```

Disambiguation of location names

As a result of the previous step, our location recognizer identifies both generic locations and locations specific to geoscience discourse. While the former can be disambiguated using existing resources, the latter cannot. For example, there is no resource to indicate the actual location for *IODP Site U1360*. To remedy this limitation, we implemented a data-driven algorithm that infers the actual location of those recognized terms. Our algorithm disambiguates these locations based on their collocation with other, known location names in the same document. In particular, we calculate the frequency of co-occurrence between a geological location (e.g., *IODP Site U1360*) and an actual location (e.g., *South Atlantic*). Then, we extract the distance between the two names based as the number of words between the names. Each geological location is disambiguated to the location with each it co-occurs the most in a collection of geoscience publications. In case of ties, we used distance information for disambiguation, i.e., we chose the actual location that tends to be closest in text. This algorithm is summarized in Algorithm 7. Table 2 shows some sample output for this disambiguation algorithm.

Algorithm 7: Get co-occurrence frequency and distance between geological location and actual location

```
site = HashMap for disambiguation
site_result = Hashmap for the result
for sentence in document do
    if (geo-site-Site or geo-site-Name) in
      sentence then
        entity = geo-site-Site or
          geo-site-Name
        site_idx = index of entity
        locations = [actual location in
          sentence]
        loc_idx = [indices of actual location
          - index of entity] for location in
          locations do
            site_result[entity][location]["freq"]
              += 1
            site_result[entity][location]["dist"]
              += loc_idx of location
    else
        pass;
for entity in site_result do
    site[entity] = location which has the
      highest frequency
```

Site	Location
Site 397	Africa
IODP Site U1341	Bering Sea
DSDP Site 216	Kerguelen
…	…

Table 2. Example results from the site inference component. The first column lists the unidentified sites; the second lists the most frequent co-occurring location.

The next step for the site identification is location normalization. Since there are multiple ways to describe the same location (e.g., *China* vs. *People's Republic of China*, or *Seoul* and *the capital city of South Korea*), the locations extracted from papers must be normalized. We used an external natural language processing tool, *geonorm*[2], for this purpose.

2 https://github.com/clulab/geonorm/

Algorithm 8: Extracting normalized entities and recalculating frequencies

```
geonorm = location normalizer
counter = frequency of locations
site = dictionary from site disambiguation
for sentence in document do
    for word in sentence do
        if word.entity == Location then
            norm_loc = geonorm(word)
            counter[norm_loc] += 1
        if word is in site then
            convert_site = site[word]
            norm_loc =
              geonorm(convert_site)
            counter[norm_loc] += 1
```

Lastly, Algorithm 8 summarizes our process to calculate the frequency of location expressions in a given document. If a given word was recognized as *Location* with CoreNLP, then we fed the recognized word into the location normalizer, and added one to the frequency of the normalized location. When the given word was in the result of site inference, then we converted the recognized word into the actual location using the result from site disambiguation, and fed the converted word into the location normalizer. We compute the frequencies of all normalized locations. Figure 1 shows an example output of this process for one paper.

```
"synthetic_data_on" : {
    "location" : {
      "Atlantic County" : 1,
      "Republic of France" : 5,
      "Aquitaine Basin" : 1,
      "Kingdom of Morocco" : 4,
      "Bretagne" : 2,
      "Portuguese Republic" : 1,
      "Mediterranean" : 1,
      "Kingdom of Spain" : 1,
      "Montenay" : 1,
      "Cahuzac" : 1
    }
```

Figure 1. The result of the site normalization for one sample publication.

Supplemental Document 3 for Park et al, 2020

Document classification

To determine whether a given geoscience paper supports (or not) the hypothesis investigated, i.e., that volcanism affects climate change, we built multiple document classifiers to automatically label a collection of papers with this information. To have the ability to investigate the details of the model such as the contribution of features to a prediction, we used two classifiers that provide this functionality: a linear support vector machines (SVM) classifier, and a Naïve-Bayes SVM (NB-SVM), using unigram and bigram features for both. In this section, we describe how the training documents were annotated, and how we trained and tested the two different SVM classifiers.

Paper Annotation

To have training and test data to build the proposed classifiers, 200 papers out of the 1,164 downloaded papers were presented to annotators, and they annotated whether the given paper supports or negates the hypothesis that volcanism impacts climate change, or are unrelated to the hypothesis. Two of the authors served as annotators. From each paper to be annotated we automatically extracted the title, abstract, introduction, and conclusion[1]. We used the crowd-sourcing platform FindingFive[2] to collect annotations. As a result, there were 400 responses (200 papers × 2 annotators), from which we constructed separate training and test partitions through cross-validation.

During the annotation, we allowed the annotators to choose more than one label per paper to encode more complex discourse. For example, the same paper could be annotated with SUPPORT and NEGATE labels, when a part of the given text supports the investigated hypothesis, but another negates it. However, this ambiguity tends to confuse machine learning methods, so we simplified multi-label annotations into a single label as follows:

1. We prioritized SUPPORT and NEGATE labels over UNRELATED. That is, when the annotator chose SUPPORT and UNRELATED, then the document would be labeled as SUPPORT. When the annotator chose NEGATE and UNRELATED, then the document would be labeled as NEGATE.
2. When SUPPORT and NEGATE were chosen at the same time (i.e., when the part of the given paper supports the idea and the other part does not), both labels would be kept as joint label NEGATE&SUPPORT.
3. When the annotator chose all possible labels (SUPPORT, NEGATE, and UNRELATED), UNRELATED is ignored, and the two remaining labels are merged into NEGATE&SUPPORT.

As a result, the responses from the annotators were normalized into four labels: SUPPORT, NEGATE,

[1] Since the papers were originally PDF files and converted to text files, some of the papers did not have correct section headings, or even any section heading in some situations. When the converted file did not have proper section headings, we extracted the first 300 words from the content to be presented to the annotators.

[2] https://www.findingfive.com

NEGATE&SUPPORT, and UNRELATED.

Linear SVM classifier

With the annotated data, we created a linear SVM classifier using the *scikit-learn*[3] package in the Python programming language. First, we extracted unigram and bigram features (e.g., from the sentence "The dog chased the cat", the unigram features are the individual words in the sentence, [the, dog, chased, cat], and the bigram features would be [start-the, the-dog, dog-chased, chased-the, the-cat, cat-end]). After extracting features, training and test data were converted to feature matrices, which contains the frequency of each feature (unigram and bigram) in the given document.

Table 1 shows an example of such a feature matrix. The first column shows the generated labels (e.g., UNREL. (UNRELATED) and SUP. (SUPPORT)), and the other columns show the frequency of each feature (e.g., geology (unigram) and volcanic-eruption (bigram)). For example, Table 1 shows that the first document is labeled as UNRELATED; the document does not contain the word "geology", nor the sequence of "volcanic" and "eruption". The second document is labeled as SUPPORT, and the word "geology" occurred once, and the sequence of "volcanic" and "eruption" occurred three times in the document.

label	geology	...	volcanic-eruption	...
UNREL.	0	...	0	...
SUP.	1	...	3	...
...	...	...	...	...

Table 1. Formatted response data for the classification task.

With the coded data, we evaluated the performance of the model using 10-fold cross-validation. In other words, we first split the data into 10 partitions, and trained the model with 9 partitions and evaluated it with the remaining partition. This process was repeated 10 times such that each partition serves as a testing partition once. Algorithm 1 summarizes this process.

The performance of this classifier is summarized in Table 2, using standard precision, recall, and F1 (i.e., the harmonic mean of precision and recall) measures, on all the 400 annotated papers. All in all, the F1 score was 82.4%, which we consider an encouraging result, especially considering the small size of the annotated dataset.

label	precision	recall	F1	N
NEG.	0.000	0.000	0.000	2
NEG.&SUP	0.000	0.000	0.000	6
SUP.	0.646	0.624	0.635	85
UNREL.	0.891	0.906	0.898	307
Overall	0.821	0.828	0.824	400

Table 2. Performance of the linear SVM classifier. N indicates the number of papers in each class.

With the linear SVM classifier, one can inspect the feature weights for each label to be predicted (i.e., the relative importance of each feature on each label). Table 3 shows the top 10 features for each label in the trained model. Even though not all top 10 features are strongly related with volcanism or

[3] https://scikit-learn.org/stable/

climate change, we find that some features were related with either volcanism (e.g., “volcanic CO”) or climate change (e.g., “cooling trend”, “fire regime”, “of flood”).

Algorithm 1: SVM classifier with 10-fold cross-validation

```
CV = 10 batches of the data
predictions = []
true_label = []
for test_data in CV do
    train_data = CV - test_data
    train_feature =
      get_features(train_data(text))
    train_label =
      get_labels(train_data(label))
    classifier = SVM()
    SVM.train(train_feature, train_label)
    prediction =
      SVM.predict(get_features(test_data(text)))
    true_label.append(get_labels(test_data(label)))
    predictions.append(prediction)
print(classification_report(prediction,
  true_label))
```

Ranking	NEGATE	NEGATE&SUPPORT	SUPPORT	UNRELATED
1	We found	may be	nannoplankton	montane
2	after tephras	both	that	lacustine
3	and increases	little	tree	Sweden
4	and that	The authors	Our	study
5	and vegetation	best correlation	10	oceanic
6	consistent statistically	efficiency	biological	history Received
7	conspicuous	extinctions the	the atmosphere	driven
8	cooling trend	of flood	from	2012 Accepted
9	deposition of	volcanic CO	anoxia	Ordovician
10	fire regime	1999	detection	12 December

Table 3. Top 10 feature weights for each label extracted by the linear SVM classifier.

NB-SVM Classifier

The above classifier uses the frequency of unigrams/bigrams as the feature values. However, Wang & Manning (2012) showed that using instead the log-count ratios produced by a Naïve Bayes (NB) model performs better for a binary classification task. Here we adapt this idea to multi-class classification, as detailed below.

Log-count ratio

Let $f^{(i)} \in R^{||V||}$ be the feature count vector for training example i with label $y^{(i)} \in \{\boldsymbol{negate}, \boldsymbol{negate\&support}, \boldsymbol{support}, \boldsymbol{unrelated}\}$. V is the set of features, and $f_j^{(i)}$ represents the number of occurrences of feature V_j in training case *i*. For example, define the count vectors as $\boldsymbol{p} = \alpha + \sum_{i:y^{(i)}=negate} f^{(i)}$ and $\boldsymbol{q} = \alpha + \sum_{i:y^{(i)}=negate\&support,support,unrelated} f^{(i)}$ for smoothing parameter $\propto$. For example, the log-count ratio for the label **negate** is:

$$\mathbf{r}_{\mathbf{negate}} = \log\left(\frac{\mathbf{p}/||\mathbf{p}||_1}{\mathbf{q}/||\mathbf{q}||_1}\right)$$

As a result, we have four different **r** ratios for NEGATE, NEGATE&SUPPORT, SUPPORT, and UNRELATED.

SVM with NB features

This classifier, henceforth referred to as NB-SVM, is similar to the previous linear SVM, with the exception that we use $\boldsymbol{x}^{(k)} = \tilde{\boldsymbol{f}}^{(k)}$ where $\tilde{\boldsymbol{f}}^{(k)} = \hat{r}_i \circ \hat{f}^{(k)}$ is the element-wise product and $i \in \{negate, negate\&support, support, unrelated\}$ (e.g., the element-wise product of the ratio $\boldsymbol{r}_{negate}$ and $\boldsymbol{f}^{(k)}$).

With the given parameters, four different SVMs (**NEGATE** vs. rest, **NEGATE&SUPPORT** vs. rest, **SUPPORT** vs. rest, and **UNRELATED** vs. rest) were trained using different ratios. As a result, for $SVM_i\ where\ i \in \{negate, negate\&support, support, unrelated\}$, $x^{(k)} = \tilde{f}^{(k)} = \hat{r}_i \circ \hat{f}^{(k)}$ and w_i, b_i could be obtained using the *linearSVC* module in *scikit-learn* package.

The original paper suggested the model $\boldsymbol{w}' = (1-\beta)\underline{w} + \beta\boldsymbol{w}$ where $\underline{w} = \left|\left|\boldsymbol{w}\right|\right|_1/|V|$ is the mean magnitude of $\boldsymbol{w}\ and\ \beta \in [0,1]$ is the interpolation parameter. In the current model, $\boldsymbol{w}_i'$ could be obtained by using $\boldsymbol{w}_i'$ of $\boldsymbol{SVM}_i$ where $i \in \{negate, negate\&support, support, unrelated\}$.

For the prediction, each $\boldsymbol{SVM}_i$ classifier makes a prediction $y_i^{(k)} \in \{-1, 1\}$. For example, $\boldsymbol{SVM}_{negate}$ returns 1 if the prediction is true (in this case, the classifier would return 1 if prediction for the test *k* is NEGATE) and -1 elsewhere. For $\boldsymbol{SVM}_i$, the prediction for the test case *k* is

$$y_i^{(k)} = sign(\boldsymbol{w}_i^T \boldsymbol{x}^{(k)} + b)$$

where $i \in \{negate, negate\&support, support, unrelated\}, \boldsymbol{w}_i\ is\ \boldsymbol{w}_i', and\ \boldsymbol{x}^{(k)}\ is\ \boldsymbol{r}_i \circ \tilde{\boldsymbol{f}}^{(k)}$. After that, **argmax** is applied to the result of the SVMs to obtain a prediction with the highest score. Thus, $i = argmax\ y_i^{(k)}$ will be the prediction for the test case *k*.

As in the evaluation of the previous linear SVM classifier, we also evaluated the performance of the NB-SVM classifier using 10-fold cross-validation. The difference here is that we tried four different NB-SVMs (i.e., four one-vs-rest NB-SVM classifiers) for each label, and we applied **argmax** over the 4 predictions at the end to select the best one, i.e., the one with the highest score (see Algorithm 2).

```
Algorithm 2: NB-SVM classifier with 10-
fold cross-validation

CV = 10 batches of the data
predictions = []
true_label = []
for test_data in CV do
    train_data = CV - test_data
    train_feature =
      get_NBfeatures(train_data(text))
    train_label =
      get_labels(train_data(label))
    test_feature =
      get_NBfeatures(test_data(text))
    test_label = get_labels(test_data(label))
    temp_prediction = []
    SVMs = [4 NB-SVM classifiers]
    for svm in SVMs do
        SVM.train(train_feature,
          train_label)
        pred = SVM.predict(test_feature)
        temp_prediction.append(pred)
    prediction = argmax(temp_prediction)
    true_label.append(test_label)
    predictions.append(prediction)
print(classification_report(prediction,
  true_label))
```

3.3.3. Results

Table 4 lists the results of the NB-SVM classifier. Similar to the observations of Wang & Manning (2012), we observed that this classifier performs better than the "vanilla" SVM, but, in our case, the improvement was not large. For example, the F1 score of the NB-SVM classifier was 83.75%, while the linear SVM's F1 score was 82.4%.

label	precision	recall	F1	N
NEG.	0.000	0.000	0.000	2
NEG.&SUP	0.000	0.000	0.000	6
SUP.	0.684	0.635	0.659	85
UNREL.	0.901	0.915	0.908	307
Overall	0.836	0.838	0.8375	400

Table 4. Performance of NB-SVM classifier.

Ensemble Model

Lastly, we build an ensemble model that combines the predictions of these two individual classifiers. Our ensemble method uses a simple voting scheme:

1. When the predictions of both models are the same (e.g., NEGATE and NEGATE), then that label (e.g., NEGATE) becomes the final output.

2. When the predictions from the two models are different, and one of the predictions is UNRELATED (e.g., SUPPORT and UNRELATED), then the prediction which is not UNRELATED becomes the final output (e.g., SUPPORT).

3. When the predictions from the two models are different and neither of them is UNRELATED, then choose the prediction from NB-SVM.

Table 6 lists the performance of this ensemble model. The ensemble performs better than the best individual model (NB-SVM), but the improvement is not large, e.g., 83.99% F1 vs. 83.7%. Nevertheless, because the ensemble method was the best overall, we used its output to classify the remaining papers in our dataset, and generate the visualizations discussed in the main body of the paper.

label	precision	recall	F1	N
NEG.	0.000	0.000	0.000	2
NEG.&SUP	0.000	0.000	0.000	6
SUP.	0.675	0.659	0.667	85
UNREL.	0.900	0.912	0.906	307
Overall	0.834	0.840	0.8399	400

Table 6. Performance of the ensemble model that combines the SVM and NB-SVM classifiers.