An Unsupervised System for Exploring Text Documents

Preprint · November 2021

DOI: 10.13140/RG.2.2.26104.57605

2 authors, including:

Biplav Srivastava
University of South Carolina
201 PUBLICATIONS **2,306** CITATIONS

SEE PROFILE

An Unsupervised System for Exploring Text Documents

Karan Aggarwal
cs1190699@cse.iitd.ac.in
Department of Computer Science, IIT Delhi

Biplav Srivastava
BIPLAV.S@sc.edu
AI Institute, University of South Carolina

ABSTRACT

We present an unsupervised system for exploring textual data which, given a document or its link (uniform resources locator), can generate multiple insights with holistic view, entity-centric view, events view and a detailed full-text view. The system also projects the insights into concepts of a set of configurable domains to help the user see its significance. The system can be used for unstructured documents like news articles, survey results, regulations and legal documents, profiles of people, and proposal calls/ request for proposals (RFPs), and has the potential to benefit common users in business, government, education or personal domains. Apart from demonstrating such general usage, we present a detailed case study with two types of documents - Proposals and Person profiles - where such a capability was found invaluable.

CCS CONCEPTS

• **Information systems** → *Retrieval tasks and goals*; • **Computing methodologies** → **Artificial intelligence**;

KEYWORDS

natural language processing, learning, visualization, unsupervised

ACM Reference Format:
Karan Aggarwal and Biplav Srivastava. 2021. An Unsupervised System for Exploring Text Documents. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

In many business scenarios, a user finds an unknown document or collection of documents that they want to make sense of. Such a document can be news, survey result, legal contract, financial report, environmental regulation or any other type. The user would initially benefit from unsupervised techniques which can help the user get started with exploring the documents and then the user can provide further guidance based on their needs.

In such situations, a few techniques have emerged. One is text summarization which works for textual content and generates a concise summary using the sentences from within the content or new sentences [14]. For structured content that is tabular, pandas profiling [2] is an unsupervised method to summarize metadata including data types, missing data and values distribution. Another emerging family of techniques is automatic visualization of data [5] which learns useful visualizations directly from data using declarative specifications of visualization [21] but only supports structured, non-textual input.

Many of the existing techniques work on visualization of structured data only. However, since unstructured data predominates, its visualization is an active area of research. The fact that unstructured textual data can span multiple domains, only makes visualization more challenging. Our work involves using several Natural Language Processing (NLP) techniques (for example, Named Entity Recognition, Dependency Parsing, Event Extraction, and Sentiment Analysis) to build a versatile system for visualizing unstructured data which spans multiple domains.

We next present our unsupervised system, *Kite*, for exploring textual data and demonstrate its capabilities. Given a document, Kite extracts keywords and generates an overview of the document from a holistic view, entity-centric view, events view and a full-text view. A snippet of *Kite's* output can be seen in Figure 1, where it visualizes a news article in the *Politics* domain. Our contributions are (a) developing a new approach for unsupervised exploration of documents at different scales, (b) projecting the document onto a set of configurable domains (Figure 2), and (c) demonstrating with a case study that the objectivity in analysis of the tool can provide meaningful and significant insights to other diverse fields, like team building in research.

The rest of the paper is organized as follows. Section 2 summarizes the related work with text visualization methods. Section 3 describes the working of our system, *Kite*. Section 4 presents a case study on research training. Section 5 presents experiments to evaluate the system output and usability. Section 6 discusses future work, and Section 7 concludes the paper.

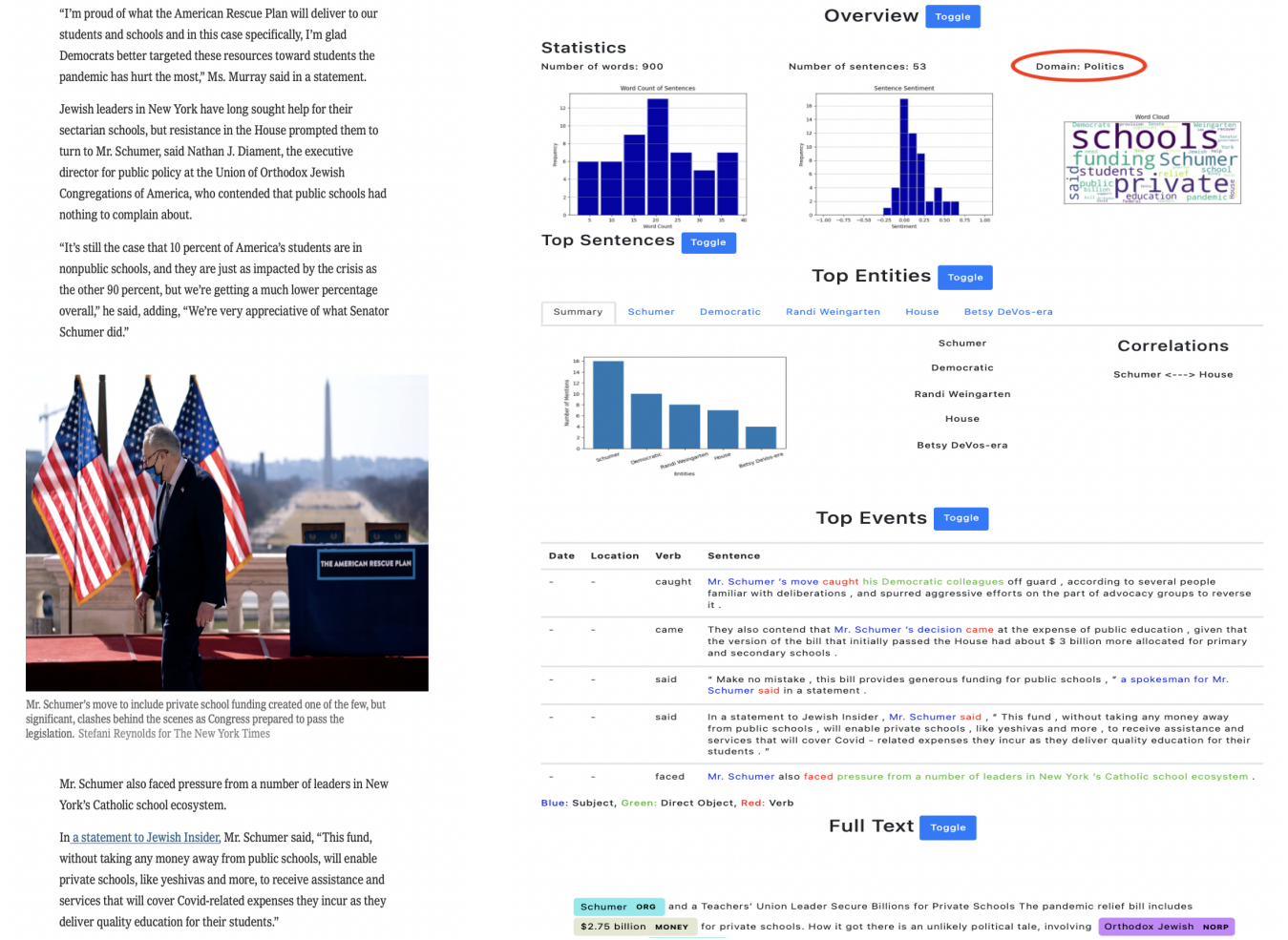
2 RELATED WORK

2.1 Text Visualization

Many works on text visualization [13] systems have used entity-centric visualizations to represent documents [7] [8]. [7] employs NLP techniques like *Named Entity Recognition* and *Sentiment Analysis* to assign roles (hero, villain, and victim) to the main entities in a news article. Another way to visualize documents is to summarize the text through a word cloud [12]. Event extraction [22] is also an active area of research in text visualization. [20] demonstrates a pipeline to extract and display events from text. Their system uses *Dependency Parsing* to extract events as (Subject, Verb, Object) triplets and represent them as a graph showing cause and effect relationships between entities. However, many of the above systems often focus on a single domain of documents. For example, [7] and [8] visualize news articles, [12] visualizes Twitter tweets, and [17] deals with scientific publications. Rarely we see something which can visualize text spanning multiple domains. Our contribution with *Kite* lies in visualizing text through a lens of multiple

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn



(a) Snippet of the original article.

(b) The domain for the document is identified as *Politics*, the top entities are e.g., *Chuck Schumer, Democratic*, and events have subjects and objects marked in blue and green respectively.

Figure 1: Visualizing the news article at: https://www.nytimes.com/2021/03/13/us/politics/schumer-weingarten-stimulus-private-schools.html.

domains and provide users with the opportunity to explore the document through different views (for example, entity-centric, and event-centric).

2.2 Keyword Extraction

Previous work in keyword extraction have employed both: supervised learning [11] and statistical techniques [3]. [11] discusses and compares 3 techniques for keyword extraction : N-grams, Noun Phrase chunks, and POS tag sequences.

However, both supervised and statistical approaches may identify completely different keywords for two different documents. It can be challenging to compare two documents with a completely different set of keywords. In our case study on research training, we hypothesize that identifying a set of standardized keywords for every document, especially a profile of a *Person* (Researcher), helps to better compare them and identify similarities. This also helps us in the task of team building in research.

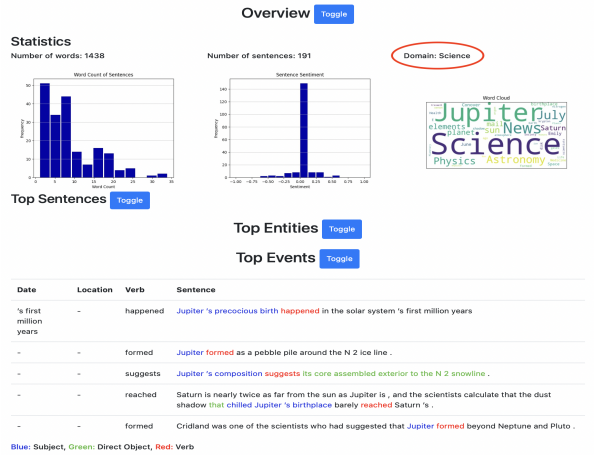
3 SYSTEM

We hypothesize that dividing our report into 4 main views helps the users get a quick overview, familiarize themselves with the main stakeholders, and answer important questions about their actions and mutual interactions. The 4 main views are:

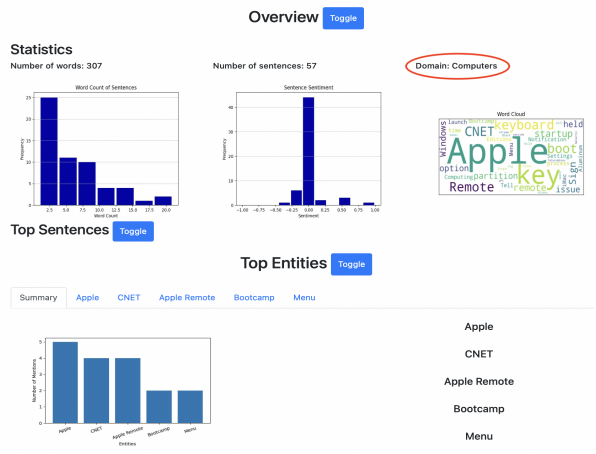
- Overview
- Top Entities
- Top Events
- Full Text

3.1 Overview

A general summary of the entire document helps the user gain insights into the sentiment of the author, and complexity and domain

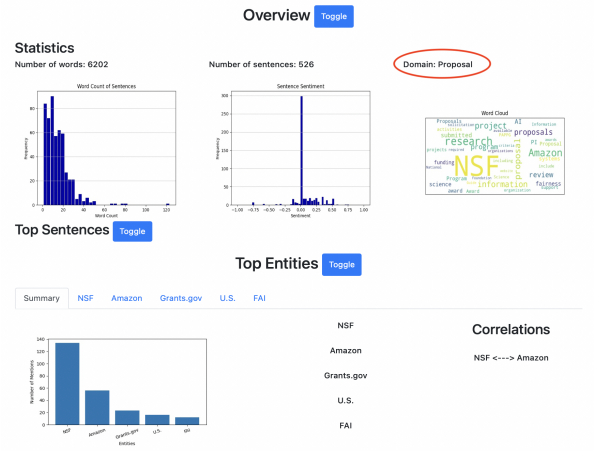


(a) Visualizing a science article at https://www.sciencenews.org/article/jupiter-chemistry-shadow-origin-planet-space-temperature.



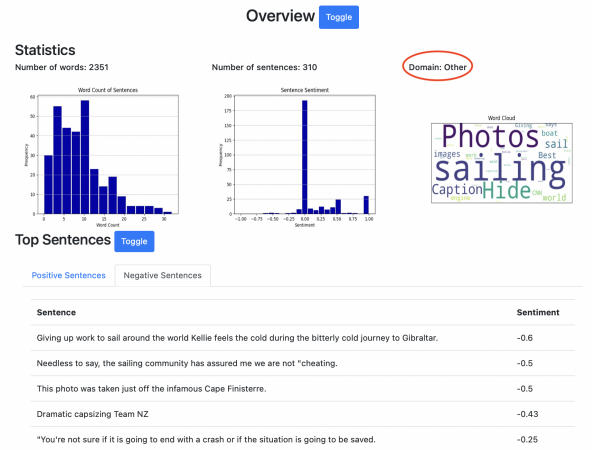
(b) Visualizing a tech news article at https://www.cnet.com/news/using-your-apple-remote-as-a-startup-key/.

(c) Visualizing a sports news article at https://www.sportingnews.com/us/tennis/news/tennis-scoring-explained-rules-system-points-terms/7uzp2evdhbd11obdd59p3p1cx. Notice that the domain is identified as *Recreational*.



(d) Visualizing the proposal at https://www.nsf.gov/pubs/2021/nsf21585/nsf21585.htm.

(e) Visualizing a legal document from the Federal Court of Australia, available as *06_126.xml* in the dataset at https://archive.ics.uci.edu/ml/datasets/Legal+Case+Reports. The domain for the document is identified as *Person*, and the top entities are e.g., *Commonwealth, Medicare, Dr. Selim.*



(f) Visualizing a news article at https://edition.cnn.com/2018/11/19/sport/sailing-wind-engine-spt-intl/index.html. Notice that the article is about 'sailing' and the domain is identified as *Other* since it did not fit into the other domains.

Figure 2: Partial output of Kite for the remaining 6 of 7 domains.

Table 1: *Domains*

Domain	Number of Documents
Politics	2373
Science	2373
Computers	2373
Recreational	2373
Proposal	754

of the document. Kite projects the document into a pre-configured set of seven domains: (a) Politics, (b) Science, (c) Computers, (d) Recreational (for example, sports), (e) Proposal, (f) Person (for example, a University profile web-page of a researcher), (g) Other (a document which does not fit into the previous six domains is classified as *Other*).

Dataset

We used the 20newsgroup [6] text dataset to collect 9492 labeled newsgroup posts, categorised into *Politics*, *Science*, *Computers*, and *Recreational*. We collected 2373 articles from each domain, with an average length of 270 words.

National Science Foundation (NSF) is a major funding agency in the United States. We used a scraping script to collect the *Introduction* and *Project Description* sections of 754 Proposals (both archived and active), from 2003 to 2022. These were on an average, 2031 words long.

Domain Classification

Our task of projecting a document onto a set of configurable domains was approached as a combination of both rule-based and supervised-learning algorithms. We did not have enough data for the *Person* domain, so we used a rule-based algorithm to classify text as a *Person*. For the other 6 domains, we have used a supervised learning algorithm. For pre-processing text, Kite uses standard NLP techniques like word-tokenization, stop-word removal, and TF-IDF. The pre-processed data is then used to train a Multinomial Naive Bayes classifier to classify text into one of 6 domains: *Politics*, *Science*, *Computers*, *Recreational*, *Proposal* and *Other*. The documents which did not fit into any domain were classified as *Other*. This achieved an F1 score of 0.79 (Precision: 0.77, Recall: 0.81).

For classification as *Person*, we developed a rule-based classifier which calculates a classification score on the basis of the number of mentions of all significant persons in the document. Let S be the set of persons in the Top 5 entities. If S is not empty, we call the person with the most mentions as the *Top Person*. Let M be the number of mentions of the *Top Person*, and T be the total number of mentions of all persons in S. Let R be the rank of the *Top Person* in the top 5 entities. This rank varies from 1 to 5. The score is calculated as:

$$Score = \frac{M \times (6 - R)}{5 \times T}$$

This score is then compared with the score of other classifiers to find the domain.

Statistics

As shown in Figure 1b, we include sentiment-based and word count-based plots of the distribution of sentences in the document, along with a Word Cloud of the most frequent words to provide insights into the readability and complexity of the document, along with the author's overall sentiment.

Top Sentences

As shown in figure 2f, Kite section identifies the most polar sentences in the document. We use Textblob [15] to calculate the sentiment polarity of each sentence in the document (from -1 (negative) to 1 (positive)) and rank them in 2 separate categories: Top 5 positive sentences and Top 5 negative sentences. This helps identify the most polar sentences and the author's opinion on various topics.

3.2 Top Entities

Our main idea is that visualizing the top entities can reveal central information about many documents, especially news articles. We show the 5 entities, their actions, associated words and context, sentiment and mutual correlation.

We use Spacy [10] for our Named Entity Recognition task. In addition, we unify references to an entity by related names, by using fuzzy string matching with *Levenshtein* distance. Furthermore, a person might be introduced by the full name and subsequently referenced by their surname. These references are also merged into a single entity. (For example, we unify mentions of "John R. Allen" and "John Allen" into a single entity as they have a string similarity of 87%. "Barack Obama" and "Obama" are also merged into a single entity as "Obama" is contained inside "Barack Obama")

The extracted entities are then ranked in order of importance according to their frequency of occurrence. Now, Kite selects the Top 5 entities and analyzes them on the following parameters:

- Actions
- Associated Words
- Sentiment
- Correlation

Actions

We assume that actions performed related to the most important entities help visualize how different entities interact with each other, and the overall effect they have on the document. As shown in figure 3, we visualize this as 2 views: (a) Actions On, and (b) Actions By.

Kite goes through the following process:

- For every entity E, Kite uses the *Spacy Dependency Parser* [9] to identify verbs associated with it.
- Verbs are usually the roots of their dependency trees.
- After traversing the tree, two cases are possible: (a) The entity E is the subject of the verb, (b) E is the object.
- If E is the subject, we add the verb to "Actions By".
- If E is the object, the verb is added to "Actions On".
- Up to 3 verbs from each case are ranked according to their sentiment and frequency, and displayed to the user.

To accurately visualize the actions, we need to know the context in which verbs are used. For example, the verb *short* can be used as *the electric circuit shorted-out* or *I shorted the stock*, depending on the domain of use. Kite displays different interpretations of the verb in various domains. As shown in Figure 3, verb interpretations

are shown for 6 domains: (a) *Politics*, (b) *Science*, (c) *Computers*, (d) *Recreational*, (e) *Proposal*, and (f) General (a domain-independent category). To display different interpretations, Kite goes through the following steps:

- *Word2Vec* [16] is used to train 100-dimensional domain-specific embeddings on the same dataset used for domain classification.
- For domain-independent embeddings, pre-trained Glove [19] embeddings are used.
- Now, given a set of 100-dimensional points for each embedding space, Kite uses Scikit Learn [18] to calculate 5 nearest neighbors (by Euclidean distance) for every verb.
- These neighbors (verbs) are then displayed to the user.

Associated Words
Our assumption is that words associated with an entity reveal information which can help the user understand the context in which they are mentioned.

For every entity E, Kite defines a set S of words associated with E. For every mention of E in the document, a total of 10 candidate words are selected (5 before and 5 after the mention). After filtering out stop-words and verbs (which are already analyzed in the above section), the remaining words are added to S. Finally, the set S is used to display a Word Cloud in the report.

Sentiment
Visualizing the most polar sentences for an entity helps reveal the author's sentiment for the particular entity, and potential biases, if any.

For every entity E, we use our earlier approach to identify and rank the most polar sentences which mention it.

Correlation
We display highly related entities. This is done by maintaining a *score* for every pair of entities. Suppose we have 2 entities, E_1 and E_2. Let S_i be the set of sentences which mention E_i.

$$score(E_1, E_2) = |S_1 \cap S_2|$$

A pair (E_1, E_2) is only displayed if

$$score(E_1, E_2) \geq max(5, n \times 0.05)$$

where n is the total number of sentences in the document.

3.3 Top Events

Events can help users contextualize information in the document and visualize how entities interact with each other and change over time. As shown in Figure 1b, Kite displays the Top 5 Events.

Each event is described as a tuple: $\langle Date, Location, Verb, Sentence \rangle$. Since a verb defines each event, Kite uses *Spacy* to identify verbs in the document. Usually, verbs are at the root of their dependency trees. The system traverses the corresponding tree for each verb and extracts relevant information about other tuple fields like *Date* and *Location*. Here, some fields may remain empty due to a lack of information present in the document. So, we rank the events on a rule-based algorithm and display the Top 5 in our report.

Table 2: *Terms* and their *Codes*

Term	Codes
Sequential circuits	B.6.1
Artificial Intelligence	I.2
Graph Algorithms	G.2.2
Theory of Computation	F
Iterative methods	G.1.4, G.1.5

3.4 Full Text

We provide an option to the user to read the full text of the document, presented in a clear and readable way, with all entities highlighted. Users can choose to read this after gaining valuable insights from other sections mentioned above.

4 CASE STUDY: RESEARCH TEAMING

Consider the domain of research teaming. Funding agencies issue *Request For Proposals* (RFPs) on themes where they are looking for ideas which they can fund. Researchers respond to these RFP calls with proposals where they explain their ideas, list the activities that will be conducted and budget to complete the work.

There are two aspects to the problem - (a) *matching*, to determine which researchers may be of interest to the calls, and (b) *teaming*, to determine which subset of researchers may want to pool together to respond to the call. We show that *Kite* can be useful in exploring both proposals and persons' background.

4.1 Keyword Extraction

Given a document, we want to extract a list of keywords. This will help us with our problems of *matching* and *teaming*. For example, if we find similar keywords in 2 documents: a Proposal and a Person (researcher), we can associate (*match*) these together. If similar keywords are found in 2 persons (researchers), they may want to pool (*team*) together to respond to the call. Now, in an unsupervised setting, we determine the keywords using a set of pre-configured *Terms* and their respective *Codes*. In our case study, we have focused on the area of Computing and have used The ACM Computing Classification System (1998) [4] to generate a set of 1532 *Terms* and their *Codes*. Kite extracts keywords (*Terms*) from the document, associates them with their *Codes* and displays the Top 10 *Codes* with their frequencies. This can be seen in Figure 4. To display *Codes*, Kite goes through the following process:

- We maintain a list of *Codes* found in the document, L
- All N-grams (from bigrams to 7-grams) are identified in the text and fuzzy string matching is used to match them to set of 1532 *Terms*. If a match is found (the string similarity is more than 85%), the *Code* corresponding to the *Term* is added to L. For example, if the document contains *Artificial Intelligence*, *I.2* is added to L.
- To ensure efficient string matching, a Trie data structure is used to store the set of pre-configured *Terms*.
- A *Term* might be associated with one or more codes. In such cases, there is a conflict. We resolve this conflict by choosing

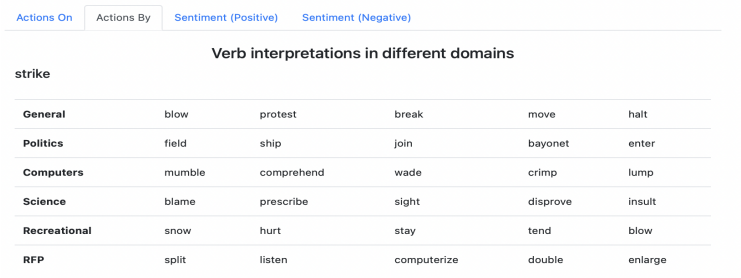
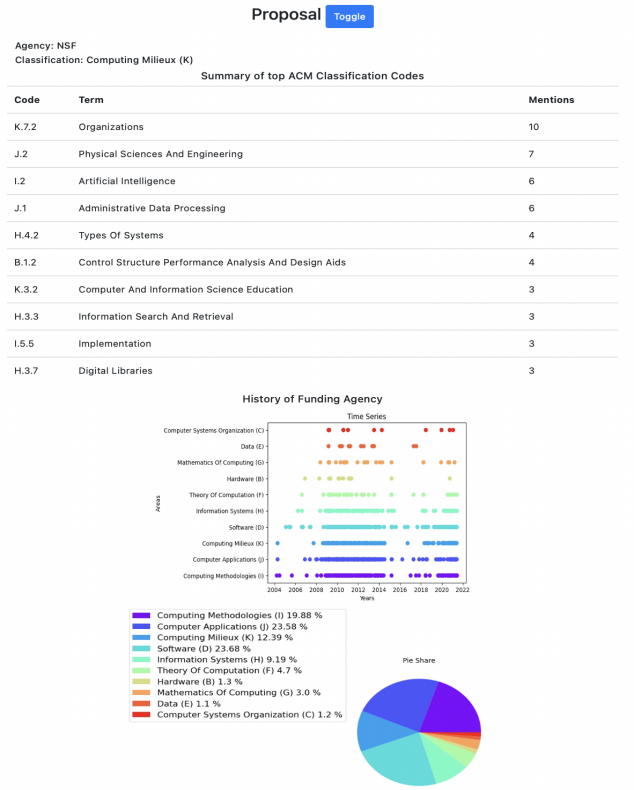
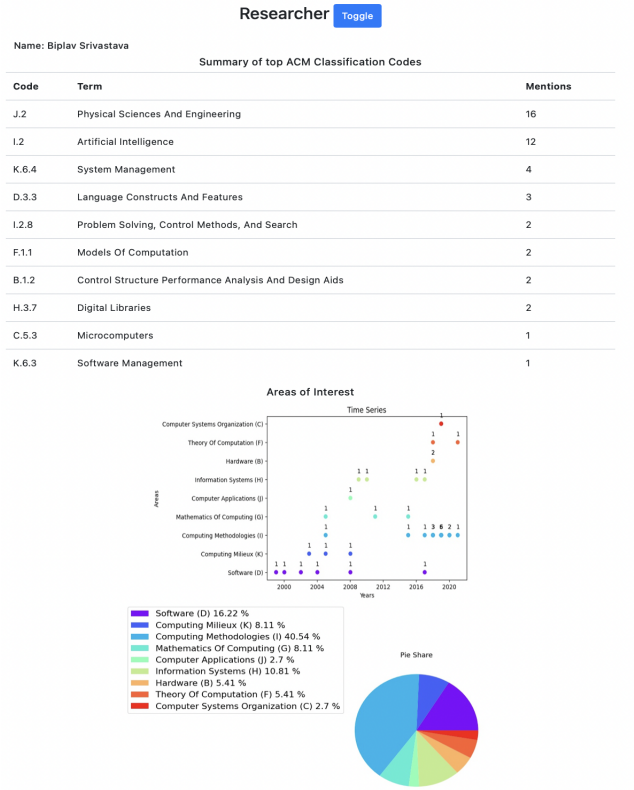


Figure 3: Visualizing interpretations of the verb *strike* in different domains



(a) Visualizing the Proposal at: https://www.nsf.gov/pubs/2021/nsf21585/nsf21585.htm. This displays the top ACM Classification *Codes* and history of the Funding Agency - NSF.



(b) Visualizing the Person profile at https://sc.edu/study/colleges_schools/engineering_and_computing/faculty-staff/srivastava.php

Figure 4: Output of Kite in our Case Study for Research Training.

the *Code* which has a higher frequency in L. For example, *Iterative methods* corresponds to both, G.1.4 and G.1.5.

- We now choose the Top 10 most frequent *Terms* and their *Codes*.
- We define the *Topic* of a document as the most frequent letter (A-K) and its corresponding *Term* in L.

4.2 Dataset

We used Grants.gov to collect the abstracts of 50,072 proposals from all major funding agencies in the United States, from 2004 to 2021. We used the above keyword extraction algorithm to associate each proposal with a set of *Codes*.

4.3 Proposal

- Top 10 *Terms* and *Codes* are identified and displayed for *matching*.
- As shown in Figure 4a, a time-series and pie-chart of main *Topics* in all proposals from the particular Funding Agency is displayed to the user. This helps the user gain context about past funding inclinations of the agency, which can be

Table 3: *Topics* and their *Codes*

Topic	Code
General Literature	A
Hardware	B
Computer Systems Organization	C
Software	D
Data	E
Theory of Computation	F
Mathematics of Computing	G
Information Systems	H
Computing Methodologies	I
Computer Applications	J
Computing Milieux	K

Table 4: *Terms* and their hierarchy

Term	Code	Hierarchy
Organizations	K.7.2	Computing Milieux->The Computing Profession->Organizations
Physical Sciences and Engineering	J.2	Computer Applications->Physical Sciences and Engineering
Artificial Intelligence	I.2	Computing Methodologies->Artificial Intelligence

useful to determine chances of proposals being accepted for a particular *Topic*.

4.4 Person (Researcher)

A researcher profile on a university website is brief generally lacks valuable information about the researcher's publications. To provide valuable insights, we need to work with more information. After identifying their name, we scrape the researcher's Google Scholar page to achieve this.

- All publication titles of the researcher are identified and collected from Google Scholar. We find that the titles are concise and contain keywords which can be fed into our keyword extraction algorithm. Information about publication titles is then combined with the original profile text for all further analysis.
- Top 10 *Terms* and *Codes* are identified and displayed for *teaming*.
- As shown in Figure 4b, a time-series and pie-chart of topics identified from all publications of the author are displayed. This helps us to visualize how their research inclinations have evolved in the past.
- Additionally, links to some recent publications from the author are shown for further introspection.

5 EXPERIMENTAL RESULTS

In this section we discuss the results of two experiments carried out to evaluate the performances of *Kite*. We perform our experiments on the following components:

(1) System Output
(2) Usability of *Kite*

5.1 System Output

We have tested our system output for the case study done in Section 4 with well defined metrics. We choose to evaluate the performance of our keyword extraction algorithm shown in Figures 4a and 4b through manual annotaion of a well-defined dataset.

Approach
The annotation dataset consisted of Kite's output on 100 proposals and 105 persons (researcher profiles). We conducted a preliminary survey with 11 users from two Universities engaged in research, consisting of undergraduate and graduate students, post-docs and faculty. We presented them with the Top 3 keywords (top 3 of the 10 keywords in Figures 4a and 4b) identified by Kite. So, for the proposal in Figure 4a, they were presented with *Organizations*, *Physical Sciences and Engineering*, and *Artificial Intelligence*. Now, sometimes when keywords are presented to a user without context, there is ambiguity. Take the example of *Organizations*. This term can be interpreted in numerous ways. To counter this ambiguity, users were provided with the whole hierarchy of the keywords from the ACM Computing Classification System. For example, the *code* for *Organizations* is K.7.2, users were presented with *Terms* corresponding to *Codes* K, K.7 and K.7.2. Table 4 explains this with some examples.

These *Terms* and their hierarchy were then presented to users as a ranked list. Users were asked to go through the *Person* profile or *Proposal* and provide their input on the ranking. They could go to the Google Scholar page for *Persons* and the full proposal URL for *Proposals* to collect more information. Some examples of annotations are shown in Tables 5 and 6.

Inter-annotator Agreement
We presented 50 *Persons* and 37 *Proposals* to a different set of users for re-annotation. This was done to measure two aspects: (a) how easy it is for people to annotate, and (b) how well do two users agree on an annotation.

Finding
We compared the rankings provided by users to our system output using the Pearson Correlation [1] metric. The Person Correlation coefficient has a range from -1 to 1: -1 being complete disagreement, and 1 being complete agreement. The Mean Pearson Correlation coefficients for our experiment, displayed in Table 7 are 0.61 (Proposal) and 0.69 (Person). The coefficients for Inter-annotator Agreement, displayed in Table 8 are 0.54 (Proposal) and 0.53 (Person).

5.2 Usability of Kite

Beyond our case study, we made *Kite* available to a wide group of people to learn how they perceive its output on various documents.

We also present the output of *Kite* on some inputs. In Figure 1, the tool was run on a news article. The system output shows entities which are of types persons or organizations (in a detailed tab view). The system accurately depicts *Chuck Schumer* and *House*

Table 5: Annotation example for Proposals. Notice that the user agrees with the ranking provided by the system.

URL	System Output	User Input
https://www.nsf.gov/pubs/2005/nsf05549/nsf05549.htm	Computer Applications->Physical Sciences and Engineering : 1 , Computing Milieux->The Computing Profession->Organizations : 2 , Computing Milieux->Management of Computing and Information Systems->System Management : 3	Physical Sciences and Engineering: 1 , Organizations: 2 , System Management: 3

Table 6: Annotation example for Persons. Notice that the user disagrees with the ranking provided by the system.

Name	Google Scholar	System Output	User Input
Jennifer Widom	https://scholar.google.com/citations?user=zdKmnYwAAAAJ&hl=en&oi=ao	Information Systems->Database Management->Database Administration : 1 , Computing Methodologies->Artificial Intelligence->Deduction and Theorem Proving : 2 , Software->Software Engineering->Distribution, maintenance, and enhancement : 3	Database Administration : 1 , Deduction and Theorem Proving : 3 , Distribution, maintenance, and enhancement : 2

Table 7: *Experiment Results*

Type	Number of documents	Mean Pearson Correlation
Proposal	100	0.61
Person	105	0.69

Table 8: *Inter-annotator Agreement Results*

Type	Number of documents	Mean Pearson Correlation
Proposal	37	0.54
Person	50	0.53

as closely related entities. The system identifies events but location is not mostly present.

Figure 2e shows the visualization of a legal document from the *Federal Court of Australia*. *Kite* points out that *Commonwealth* is an important entity, which can be expected from an Australian Court. It also highlights *Medicare* and *Dr. Selim* as top entities, which informs us that the document is about *Dr. Selim* and is also related to Medicine.

We have tested *Kite* with a wide variety of documents from different domains. Overall, our system can be used to analyze complex documents to give an entity-based and event-based view. In our entity-based visualization, we identify the main stakeholders, their mutual interaction, and the actions and sentiments related to them. We also relate the actions associated with entities to our event-based view, where we identify and display information on the main events. We currently use standard NLP techniques for various tasks such as Named Entity Recognition, Sentiment Analysis, and Word2Vec.

Evaluation results of our case study on Proposals and Persons (researchers) show that *Kite* can be useful and also be integrated into a larger system for team building and matching. It achieves this by extracting and matching keywords to the ACM Computing Classification System in an unsupervised way using fuzzy string matching. Currently, this is possible only for documents related to Computing. We plan to expand it's capabilities and provide support for multiple other domains.

6 DISCUSSION

There are many avenues for enhancing our method. *Kite* uses fuzzy string for tasks like matching to resolve references to the same entity by various names and keyword extraction. However, this approach can be improved to develop a more comprehensive algorithm, not solely based on string-similarity. The second direction is domain identification. In current work, we consider the content to be associated to only one domain and assign it to the domain classifier model with the highest confidence. One can relax this when multiple domain classifiers have similar confidence, and mix insights but it has to be evaluated for usability.

Third, one can do a larger user study of *Kite* to include input from more users. Fourth, in the research teaming case study presented, we focused on persons and proposals in computing domain and thus, worked with the ACM classification codes. This can be expanded to other scientific disciplines. Finally, one can augment the current capability with explanation about why certain domains, entities, events and sentiments are highlighted.

7 CONCLUSION

In this paper, we have presented a new approach for unsupervised exploration of documents spanning multiple domains. Our approach assumes no metadata and generates a visualization using various NLP techniques solely on the basis of information present in the text. We have also demonstrated *Kite's* effectiveness with a case study in research training by evaluating the system output on

multiple documents. *Kite* presents the user with quick insight into documents and can be seen as complement to text summarization, and a building block for advanced text exploration.

REFERENCES

[1] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. 2009. *Pearson Correlation Coefficient.* Springer Berlin Heidelberg, Berlin, Heidelberg, 1–4. https://doi.org/10.1007/978-3-642-00296-0_5

[2] Simon Brugman. 2019. pandas-profiling: Exploratory Data Analysis for Python. https://github.com/pandas-profiling/pandas-profiling.

[3] Jason Chuang, Christopher D Manning, and Jeffrey Heer. 2012. "Without the clutter of unimportant words" Descriptive keyphrases for text visualization. *ACM Transactions on Computer-Human Interaction (TOCHI)* 19, 3 (2012), 1–29.

[4] Neal Coulter, James French, Ephraim Glinert, Thomas Horton, Nancy Mead, Roy Rada, Anthony Ralston, Bernard Rous, Allen Tucker, Peter Wegner, Eric Weiss, and Carol Wierzbicki. 1998. Computing Classification System 1998: Current Status and Future Maintenance Report of the CCS Update Committee. 39 (02 1998).

[5] V. Dibia and Ç. Demiralp. 2019. Data2Vis: Automatic Generation of Data Visualizations Using Sequence-to-Sequence Recurrent Neural Networks. *IEEE Computer Graphics and Applications* 39, 5 (2019), 33–46. https://doi.org/10.1109/MCG.2019.2924636

[6] empty. empty. 20 Newsgroups Dataset. http://people.csail.mit.edu/jrennie/20Newsgroups/

[7] Diego Gomez-Zara, Miriam Boon, and Larry Birnbaum. 2018. Who is the Hero, the Villain, and the Victim? Detection of Roles in News Articles Using Natural Language Techniques. In *23rd International Conference on Intelligent User Interfaces* (Tokyo, Japan) *(IUI '18)*. Association for Computing Machinery, New York, NY, USA, 311–315. https://doi.org/10.1145/3172944.3172993

[8] M. Grobelnik and D. Mladenic. 2004. Visualization of News Articles. *Informatica (Slovenia)* 28 (2004), 375–380.

[9] Matthew Honnibal and Mark Johnson. 2015. An Improved Non-monotonic Transition System for Dependency Parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, 1373–1378. https://aclweb.org/anthology/D/D15/D15-1162

[10] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. *spaCy: Industrial-strength Natural Language Processing in Python.* https://doi.org/10.5281/zenodo.1212303

[11] Anette Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*. 216–223.

[12] Rianne Kaptein. 2012. Using Wordclouds to Navigate and Summarize Twitter Search Results.. In *EuroHCIR*. 67–70.

[13] Kostiantyn Kucher and Andreas Kerren. 2015. Text visualization techniques: Taxonomy, visual survey, and community insights. In *2015 IEEE Pacific Visualization Symposium (PacificVis)*. 117–121. https://doi.org/10.1109/PACIFICVIS.2015.7156366

[14] Yang Liu and Mirella Lapata. 2019. Text Summarization with Pretrained Encoders. arXiv:1908.08345 [cs.CL]

[15] Steven Loria. 2018. textblob Documentation. *Release 0.15* 2 (2018).

[16] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. arXiv:1301.3781 [cs.CL]

[17] Lisa J. Miller, Rich Gazan, and Susanne Still. 2014. Unsupervised Classification and Visualization of Unstructured Text for the Support of Interdisciplinary Collaboration. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work Social Computing* (Baltimore, Maryland, USA) *(CSCW '14)*. Association for Computing Machinery, New York, NY, USA, 1033–1042. https://doi.org/10.1145/2531602.2531666

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[19] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. 1532–1543. http://www.aclweb.org/anthology/D14-1162

[20] Delia Rusu, Blaž Fortuna, Dunja Mladenic, Marko Grobelnik, and Ruben Sipoš. 2009. Document Visualization Based on Semantic Graphs. In *2009 13th International Conference Information Visualisation*. 292–297. https://doi.org/10.1109/IV.2009.57

[21] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer. 2017. Vega-Lite: A Grammar of Interactive Graphics. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (2017), 341–350. https://doi.org/10.1109/TVCG.2016.2599030

[22] Franz Wanner, Andreas Stoffel, Dominik Jäckle, Bum Chul Kwon, Andreas Weiler, Daniel A Keim, Katherine E Isaacs, Alfredo Giménez, Ilir Jusufi, Todd Gamblin, et al. 2014. State-of-the-Art Report of Visual Analysis for Event Detection in Text Data Streams.. In *EuroVis (STARs)*. Citeseer.