Sentiment Analysis of User Comments on Reddit Posts: A Data-driven Approach Using R*

Joseph Chung

Apr 20, 2023

Abstract

This paper presents an in-depth analysis of the growth, engagement, and sentiment trends in the fields of Machine Learning, Artificial Intelligence, and Data Science based on post activity and discussions on a popular platform. Using various visualization techniques, sentiment analysis models, and topic modeling approaches like Latent Dirichlet Allocation (LDA), the study uncovers the dynamics, interconnectedness, and sentiments associated with these fields. The findings reveal a significant increase in post activity, distinct words per post, and predominantly positive sentiment, indicating growing interest, understanding, and appreciation for the potential applications and implications of these technologies. Furthermore, the study highlights the synergistic relationship between data science, artificial intelligence, and deep learning, emphasizing the importance of collaboration and interdisciplinary research for driving innovation and unlocking new potential applications. Despite some limitations, such as data source biases and potential inaccuracies in sentiment and topic modeling, the results provide valuable insights into the evolving landscape of Machine Learning, Artificial Intelligence, and Data Science, and their impact on various industries and the job market.

Table of contents

*Code and data are available at: https://github.com/UtopianYoungChung/Sentiment-Analysis-Using-R.git

List of Figures

List of Tables

1 Introduction

The rapid growth of online forums and communities discussing various topics, including Machine Learning(ML), has led to an abundance of user-generated content in the form of text data. Analyzing this data can reveal valuable insights into user perceptions, opinions, and sentiments toward specific subjects. Sentiment analysis, or opinion mining, is a natural language processing technique used to determine the sentiment expressed in a piece of text, such as positive, negative, or neutral. This paper focuses on employing sentiment analysis techniques to understand the sentiment polarity of user comments related to the topics of Machine Learning, Data Science, and Artificial Intelligence.

1.1 Background and motivation

Machine Learning, Data Science, and Artificial Intelligence have become popular fields of study and research due to their vast applications in various industries. As a result, online communities have emerged where people discuss these topics, share knowledge, and express their opinions. Analyzing the sentiment of these users' comments can provide insights into the perception of the Data Science community, identify common issues or concerns, and uncover emerging trends. This understanding could be valuable for researchers, educators, and industry professionals who aim to improve their work based on community feedback.

1.2 Research questions

This study aims to address the following research questions:

1. What is the overall sentiment polarity (positive, negative, or neutral) of user comments related to Machine Learning, Data Science, and Artificial Intelligence?
2. How is the sentiment polarity distributed across different posts in the dataset?
3. Are there any correlations between post attributes (e.g., post length, engagement) and sentiment polarity?
4. Can any trends or patterns in sentiment polarity be identified over time or in relation to specific subtopics within the topics of Machine Learning, Data Science, or Artificial Intelligence?

1.3 Scope and limitations

The scope of this study is limited to the analysis of a dataset containing post _ids and comments of the users related to the topics. The results and interpretations are based solely on the data provided and may not be generalizable to other online communities or topics. The sentiment analysis model employed in this study is subject to the limitations inherent in natural languages processing techniques, such as context ambiguity and handling of sarcasm or irony. Additionally, the study does not account for demographic information or biases present in the dataset, which could potentially influence the sentiment analysis results.

2 Literature Review

2.1 Sentiment analysis in online communities

Over the past decade, sentiment analysis has become increasingly popular as a tool to mine user opinions and emotions from online platforms, such as social media, blogs, and forums (Pang and Lee 2008). Numerous studies have focused on the application of sentiment analysis in various domains, such as politics (Tumasjan

et al. 2010), finance (Bollen, Mao, and Zeng 2011), and consumer reviews (Dave, Kushal and Lawrence, Steve and Pennock, David M. 2003). These studies demonstrate the potential of sentiment analysis as a valuable technique for understanding user-generated content and informing decision-making processes.

2.2 Sentiment analysis techniques in R

R is a widely used programming language for statistical analysis and data visualization, with several libraries and packages available for natural language processing and sentiment analysis (R Core Team 2020). Some of the most popular sentiment analysis packages in R include tidytext (Silge and Robinson 2016a), and syuzhet (Jockers 2020). These packages provide various algorithms and techniques for sentiment analysis, such as lexicon-based approaches, machine-learning models, and deep-learning methods. Researchers have employed these techniques to analyze sentiment in diverse contexts, ranging from Twitter data (Wijeratne 2017) to news articles (earney 2019)

2.3 Applications of sentiment analysis in Machine Learning research

Sentiment analysis has been employed in Machine Learning research to understand user perceptions, opinions, and trends related to the field. For example, Aliza (Sarlan, Aliza and Nadam, Chayanit and Basri, Shuib 2014) analyzed the sentiment of tweets related to Machine Learning conferences, revealing insights into the community's responses to specific presentations and events. In another study, Wang (Wnag 2019) investigated the sentiment of user comments on popular Machine Learning platforms, such as GitHub and Stack Overflow, to identify trends and patterns in the community's engagement with specific algorithms and techniques. These studies highlight the potential of sentiment analysis to uncover valuable insights into the Machine Learning community and inform future research directions.

The literature demonstrates the growing interest in sentiment analysis as a valuable tool for understanding user-generated content in various domains. R has emerged as a popular platform for sentiment analysis due to its extensive libraries and packages. However, there remains a need for more comprehensive studies that investigate the sentiment of user comments related to Machine Learning, particularly in the context of online communities where users share their knowledge and opinions on the topic. This study aims to address this gap by employing sentiment analysis techniques in R to analyze a dataset of post IDs and comments related to Machine Learning on the Reddit platform.

3 Data Collection and Preprocessing

This section describes the process of data collection, including web scraping and API requests, and the subsequent preprocessing steps applied to the dataset to prepare it for sentiment analysis.

3.1 Data Collection

The dataset used in this study was collected from Reddit, a popular online platform where users can share and discuss a wide range of topics. The data was obtained by performing web scraping and API requests on the topics of Machine Learning, Data Science, and Artificial Intelligence posts between 2013 to 2023. The Reddit API (https://www.reddit.com/dev/api/) was utilized to access and retrieve relevant posts and associated comments from the specified subreddits (e.g., r/MachineLearning, r/datascience, r/artificial).

To perform the web scraping and API requests, the Python package PRAW (Python Reddit API Wrapper) ("The Python Reddit API Wrapper" 2016) was used, which provides a convenient interface to interact with the Reddit API. The data collection process involved:

1. Authenticating with the Reddit API using a registered application's credentials.

2. Querying the API for posts related to the specified topics (Machine Learning, Data Science, and Artificial Intelligence) within the targeted subreddits.

3. Combining the collected data into a structured dataset for further analysis.

Table 1: Ten observations from dataset of Reddit API

post_id	subreddit	score	created_year
k59xar	datascience	249	2020
gqdq2o	MachineLearning	392	2020
l8lqaw	datascience	414	2021
kv8hpb	datascience	392	2021
sx9o1g	datascience	500	2022
co37ut	MachineLearning	524	2019
10nodn4	MachineLearning	859	2023
wfkz9p	datascience	329	2022
mwur7p	datascience	455	2021
vc9l3r	artificial	238	2022

Table 1 shows ten observations of the dataset, with 4 selected attributes of 13. The data types of 13 attributes are explained below:

- post_id: VARCHAR
- subreddit: CHAR
- created_utc: INT
- selftext: CHAR
- post_url: VARCHAR
- post_title: VARCHAR
- link_flair_text: CHAR
- score: INT
- num_coments: INT
- upvote_ratio: FLOAT
- created_date: DATETIME
- created_year: YEAR
- comment: VARCHAR

It contains 223,781 observations and 13 variables in total.

3.2 Data cleaning and preprocessing

After obtaining the raw dataset, several preprocessing steps were performed to clean and prepare the data for sentiment analysis. These steps include:

3.2.1 Basic cleaning

There are different methods we can use to condition the data, but this paper will stick to the basics and use gsub() and apply() functions to do the cleaning (Becker and WLLKS 1998).

First, we got rid of those pesky contractions by creating a little function that handles most scenarios using gsub(), and then applied that function across all comments. Second, all those special characters that muddy the text were removed with gsub() function and a simple regular expression. Lastly, to be consistent, we converted everything to lowercase with tolower() function.

All the work was done on a separate R file called 02-data_preprocessing.R. It can be found in the 'script' folder.

Table 2: 10 Random observations were drawn from cleaned dataset from Reddit API

post_id	comment
yt4380	lolololol
ft5nsy	abraham wald
xwv9m3	thanks haha
k77sxz	deleted
u24lh7	where can i sign up
8augc6	neck urself if serious
4cv9ef	deleted
wp2vqk	what is hl7
jvwgq3	you are hired lol
g6og9l	i like example 1 2 4

3.2.2 Text Mining

Text mining and text analytics can be used interchangeably. The primary objective is to uncover significant data that may be concealed or unfamiliar beneath the surface. One of the techniques utilized in text mining is Natural Language Processing (NLP), which aims to unravel the intricacies in written language through various methods such as tokenization, clustering, entity extraction, and analyzing the relationships between words. Furthermore, NLP employs algorithms to detect patterns and quantify subjective data.

3.2.3 Tokenization

To unnest the tokens, we use the tidytext library (Silge and Robinson 2016b). From the tidy text framework, we broke the text into individual tokens and transform it into a tidy data structure. To do this, we used tidytext's unnest_tokens() function. unnest_tokens() requires at least two arguments: the output column name that will be created as the text is unnested into it ("word," in this case) and the input column that holds the current text (i.e. comments). We then took the Reddit dataset and pipe it into unnest_tokens() and then removed stop words. There are different lists to choose from, but here we used the lexicon called stop_words from the tidytext package.

After the tokenization, we used dplyr's anti_join() (Wickham et al. 2021) to remove stop words. Then we used distinct() to get rid of any duplicate records as well. Lastly, we examined the class and dimensions

of our new tidy data structure 'reddit_comments_filtered' is a data frame with 2914188 total words (not unique words) and 8 columns. The following is the snapshot with chosen token 'deepmind':

Table 3: Tokenized Format

word	post_id	subreddit	score	years
deepmind	w6kj9y	MachineLearning	1732	22-23
deepmind	wiqjxv	MachineLearning	1339	22-23
deepmind	y89xqw	MachineLearning	946	22-23
deepmind	vfl57t	MachineLearning	515	22-23
deepmind	wcug1f	MachineLearning	469	22-23
deepmind	xpe6bi	datascience	418	22-23
deepmind	w44lkv	datascience	407	22-23
deepmind	xw6r5k	datascience	408	22-23
deepmind	zetvmd	MachineLearning	368	22-23
deepmind	xwfvlw	MachineLearning	364	22-23

Table 3 shows us the tokenized, unsummarized, tidy data structure.

4 Descriptive Statistics

During this section, we will be using creative graphs from the ggplot2 (Wickham 2016), circlize (Gu et al. 2014), and yarrr (Phillips 2017) packages.

4.1 Shipshape: Word Count Per Post

The term "shipshape" is commonly used to indicate that everything is well-organized, neat, and tidy. In this context, an interesting perspective is presented on the orderly and organized data that demonstrates the lexical diversity, or the range of vocabulary, found within post comments over time. The pirate plot, a sophisticated technique for graphing a continuous dependent variable (such as word count) against a categorical independent variable (such as 'year'), is utilized to create a comprehensive and informative visual representation that incorporates both raw data points and statistical analysis.

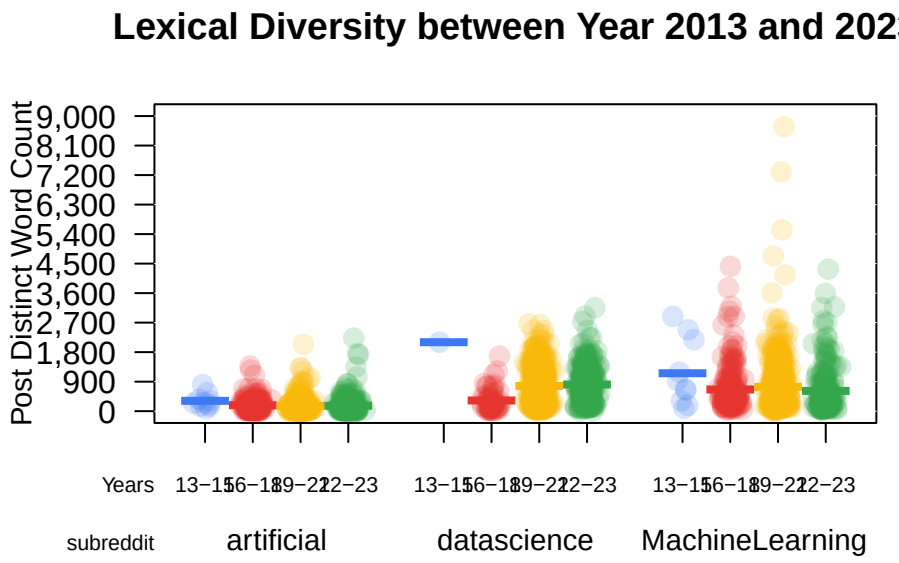

Figure 1: Lexical Diversity over the past 10 years, between '13 and '23

In this pirate plot displayed in figure Figure 1, each colored circle symbolizes a post. The dense clusters within each category indicate a considerable volume of posts in the dataset. Upon observation, it becomes evident that there were only a few posts during the period of '13-15 across all topics, whereas from '16-18 onwards, there was a surge in post activity. Additionally, it is worth noting that there is a slight upward trend in the number of distinct words per post in the Machine Learning topic. The solid horizontal line represents the mean word count for the corresponding years.

4.2 Assumption Checking and Methods

4.2.1 All Year Round: Post Count Per Year

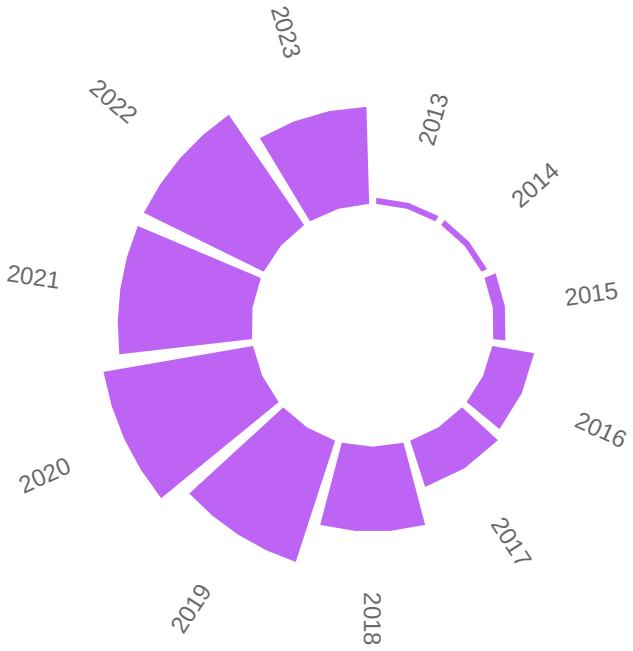

Figure 2: The total amount of posts on Reddit platform between Year '13 and '23

Figure 2 depicts the total number of posts in the areas of 'Machine Learning', 'Artificial Intelligence', and 'Data Science' from 2013 to 2022. During this time, there were many advancements in artificial intelligence, including the development of Stable Diffusion (2022), DALL-E (2021) ChatGPT (2022) and other important breakthroughs. Now that we are in 2023, the number of posts has already grown a lot compared to previous years. This suggests that people are more active in these topics than they were in the last 10 years.

4.2.2 Chords: Subreddit posts by Years

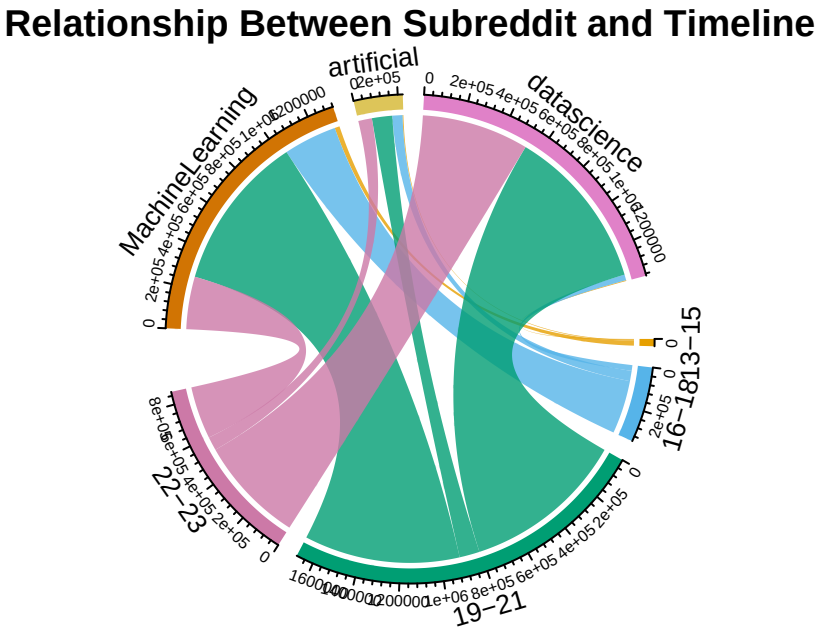

Figure 3: Relationship between Subreddit posts and timeline in semi-annual frames

As shown above in Figure 3, the subject Machine Learning is experienced significant growth between 2019 and 2021, and this trend has persisted into 2023.

4.3 Sentiment analysis model selection

4.3.1 Explore sentiment lexicons

The tidytext (Silge and Robinson 2016b) package includes a dataset called sentiments which provides several distinct lexicons. These lexicons are dictionaries of words with an assigned sentiment category or value. Tidytext provides three general purposes lexicons:

- AFINN: assigns words with a score that runs between -5 and 5, with negative scores indicating negative sentiment and positive scores indicating positive sentiment
- Bing: assigns words into positive and negative categories
- NRC: assigns words into one or more of the following ten categories: positive, negative, anger, anticipation, disgust, fear, joy, sadness, surprise, and trust

In order to examine the lexicons, we create a data frames.

4.3.1.1 Create sentiment datasets

Start off by creating Post sentiment datasets for each of the lexicons by performing an inner_join() on the get_sentiments() function. We pass the name of the lexicon for each call. For this paper, we use Bing for binary and NRC for categorical sentiments. Since words can appear in multiple categories in NRC, such as Negative/Fear or Positive/Joy, we will also create a subset without the positive and negative categories to use later on.

4.3.1.2 Mood changes over time

Since we are looking at sentiment from a polarity perspective, we might want to see whether or not it changes over time. We use geom_smooth() (Wickham 2016) with loses method for a smooth curve and another geom_smooth() with method=lm for a linear smooth curve.

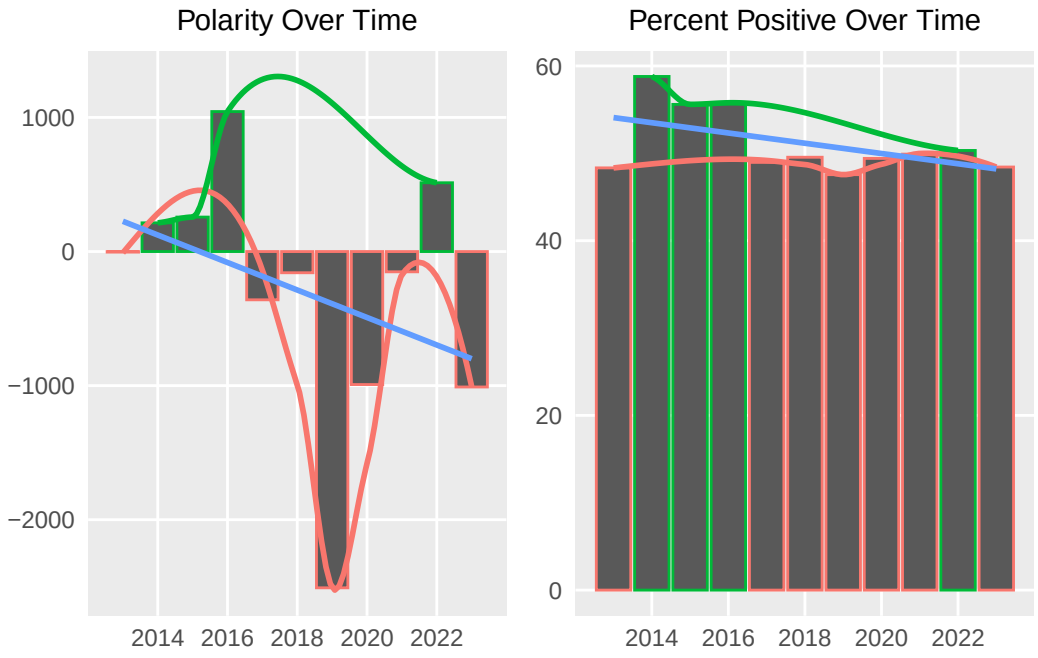

Figure 4: Sentiment ploarity over time and Percet postivie over time are descreasing

Figure 4 shows that the overall polarity trend over time is negative in both cases.

4.3.1.3 Sentiment in Machine Learning

Let's take a deeper look into the mood of a specific topic over time: Machine Learning

How would NRC model interpret the mood of posts under Machine Learning?

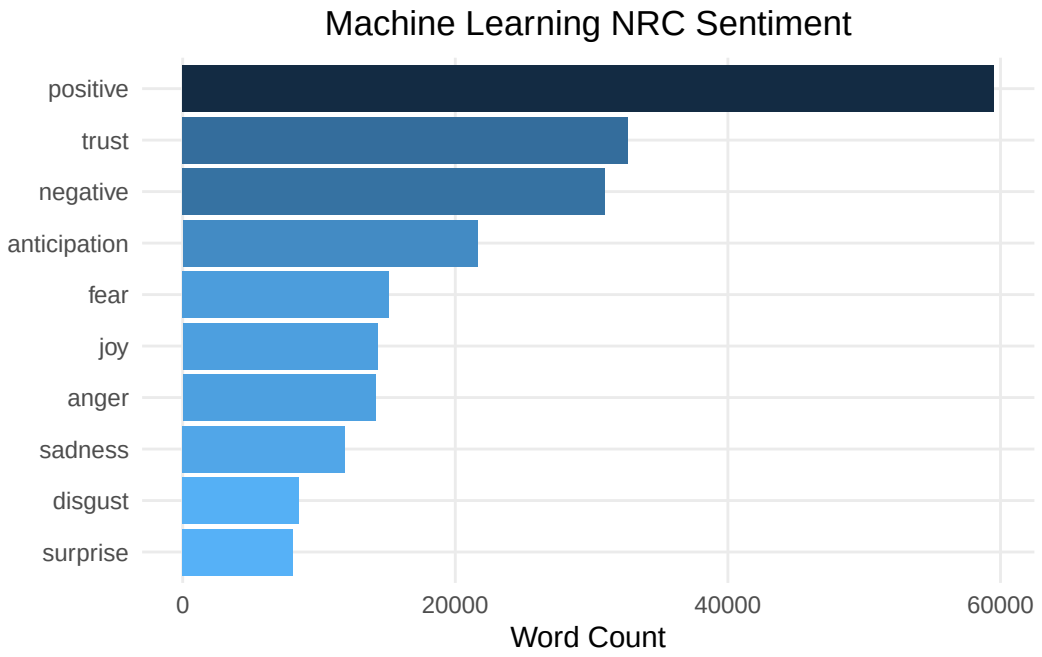

Figure 5: The interpretation of NRC model in mood of posts under discussion of Machine Learning in year '19-21

Although highly subjective, the results in Figure 5 appear to confirm that there is overwhelmingly positive and trust sentiment with Machine Learning. Let's go further in depth by looking at the sentiment categories with distinctive topics ('Career' and 'discussion') and see if they appear to be correlated.

Sub-NRC sentiments, Figure 6, show that anticipation is increasing under career category over time. Does this tell us that people are worrying about their jobs?

Bing sentiments, Figure 7, show many positive words from the posts under "discussion' category. Here, we can notice that some of the negative words included 'hard', 'wrong', and 'issue'.

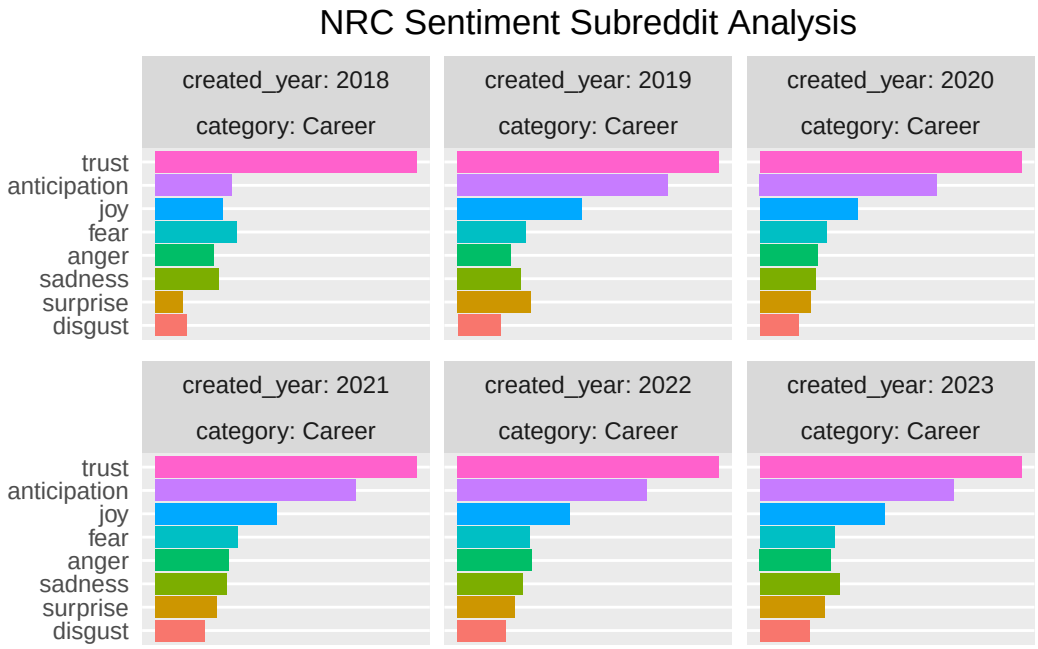

Figure 6: NRC sentiment showing the level of anticipation is increasing over time, under 'Career'

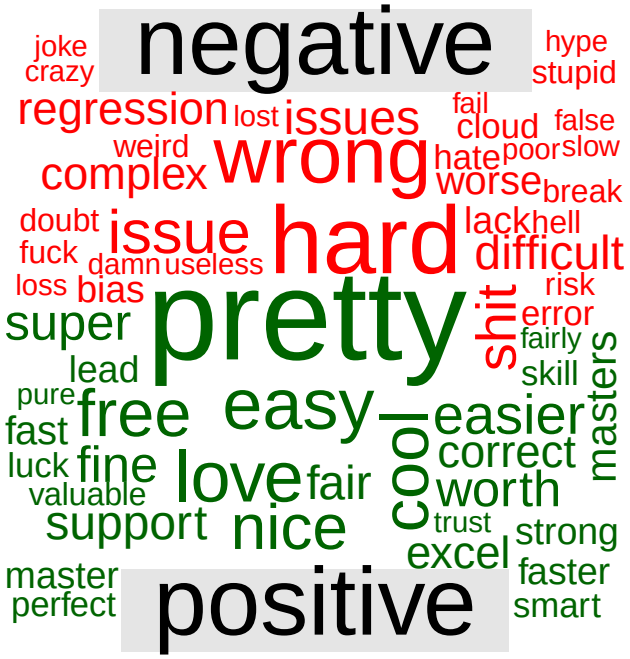

Figure 7: Bing sentiment showing the level of anticipation increasing over time, under 'Disscussion'

4.4 Model training and validation

4.4.1 Word embeddings

Word embeddings are a way to represent text data as vectors of numbers based on a huge corpus of text, capturing semantic meaning from words' context. First, let's filter out words that are used only rarely in this data set and create a nested dataframe with one row per post.

Next, we create a slide_windows() function using the slide() function from the slider package (Vaughan 2022). This new function identifies skipgram windows in order to calculate the skipgram probabilities, how often we find each word near each other word.

Now we can find all the skipgram windows; let's calculate how often words occur on their own, and how often words occur together with other words. We do this using point-wise mutual information (PMI) (Alto 2020).

In statistics, probability theory and information theory, pointwise mutual information (PMI), or point mutual information, is a measure of association. It compares the probability of two events occurring together to what this probability would be if the events were independent.

$$pmi(x;y) \equiv \log_2 \frac{p(x,y)}{p(x)p(y)} = \log_2 \frac{p(x|y)}{p(x)} = \log_2 \frac{p(y|x)}{p(y)}$$

It's the logarithm of the probability of finding two words together, normalized for the probability of finding each of the words alone. We use PMI to measure which words occur together more often than expected based on how often they occur on their own. This step is the computationally expensive part of finding word embeddings. However, by using furrr package (Vaughan and Dancho 2022), we can take advantage of parallel processing because identifying skipgram windows in one document is independent of all the other documents.

When PMI is high, the two words are associated with each other, i.e., likely to occur together. When PMI is low, the two words are not associated with each other, unlikely to occur together.

We then determine the word vectors from the PMI values using singular value decomposition (SVD). SVD is a method for dimensionality reduction via matrix factorization ("Cholesky Decomposition: Matrix Decomposition," n.d.) that works by taking our data and decomposing it onto special orthogonal axes.

In our application, we will use SVD to factor the PMI matrix into a set of smaller matrices containing the word embeddings with a size we get to choose. We will be using the widely_svd() function in widyr (Robinson and Silge 2022), creating 100-dimensional word embeddings.

We now have our Reddit word embeddings.

4.5 Exploring Reddit word embeddings

Now that we have determined word embeddings for the data set of Reddit posts, let's explore them and discuss how they are used in modelling.

Each word can be represented as a numeric vector in the new set of 100-dimensional feature spaces. A single word is mapped to only one vector, therefore, all senses of a word are conflated in word embeddings. Because of this, word embeddings are limited to understanding lexical semantics.

In order to find out which words are close to each other, we created a simple function that will find the nearest words to any given examples using our newly created word embeddings. This function takes the tidy word embeddings as input, along with a word/token as a string. Let's explore what words are closest to "MachineLearning" in the data set of Reddit posts, as determined by our word embeddings. The results are shown on Table 4

Table 4: Top 10 related tokens with their probabilities to 'Machine Learning'

item1	value
machinelearning	1.0000000
reddit	0.7948791
remindmebot	0.7773240
datascience	0.7720273
comments	0.7587609
compose	0.7370770
message	0.7262611
remind	0.7137436
excludeme	0.7120444
totesmessenger	0.6957258

Since we have found word embeddings via singular value decomposition, we can use these vectors to understand what principal components explain the most variation in the Reddit posts.

In linear algebra, the singular value decomposition (SVD) is a factorization of a real or complex matrix ("Singular Value Decomposition," n.d.). It generalizes the eigendecomposition of a square normal matrix with an orthonormal eigenbasis to any $m \times n$ matrix. It is related to polar decomposition. Specifically, the singular value decomposition of an $m \times n$ complex matrix is the factorization of the form:

$$M = U\Sigma V$$

, where U is an $m \times m$ complex unitary matrix, is an m ×n rectangular diagonal matrix with non-negative real numbers on the diagonal, V is an n ×n complex unitary matrix, and V* is the conjugate transpose of V. Such decomposition always exists for any complex matrix. If M is real, then U and V can be guaranteed to be real orthogonal matrices; in such contexts, the SVD is often denoted:

$$U\Sigma V^T$$

The orthogonal axes that svd used to represent our data were chosen so that the first axis accounts for the most variance, the second axis accounts for the next most variance, and so on. We can now explore which and how much each original tokens contributed to each of the resulting principal components produced using SVD.

Figure 8 shows some interesting aspects of the relationships in how very common words are used together. For instance, component 6 shows us how 'statistics', 'algebra', 'mathematics' are often used together; component 3 shows 'vectors', 'parameters', and 'gradient' are often used together. Considering how these words are related to data science, artificial intelligence, and deep learning, this results are interesting.

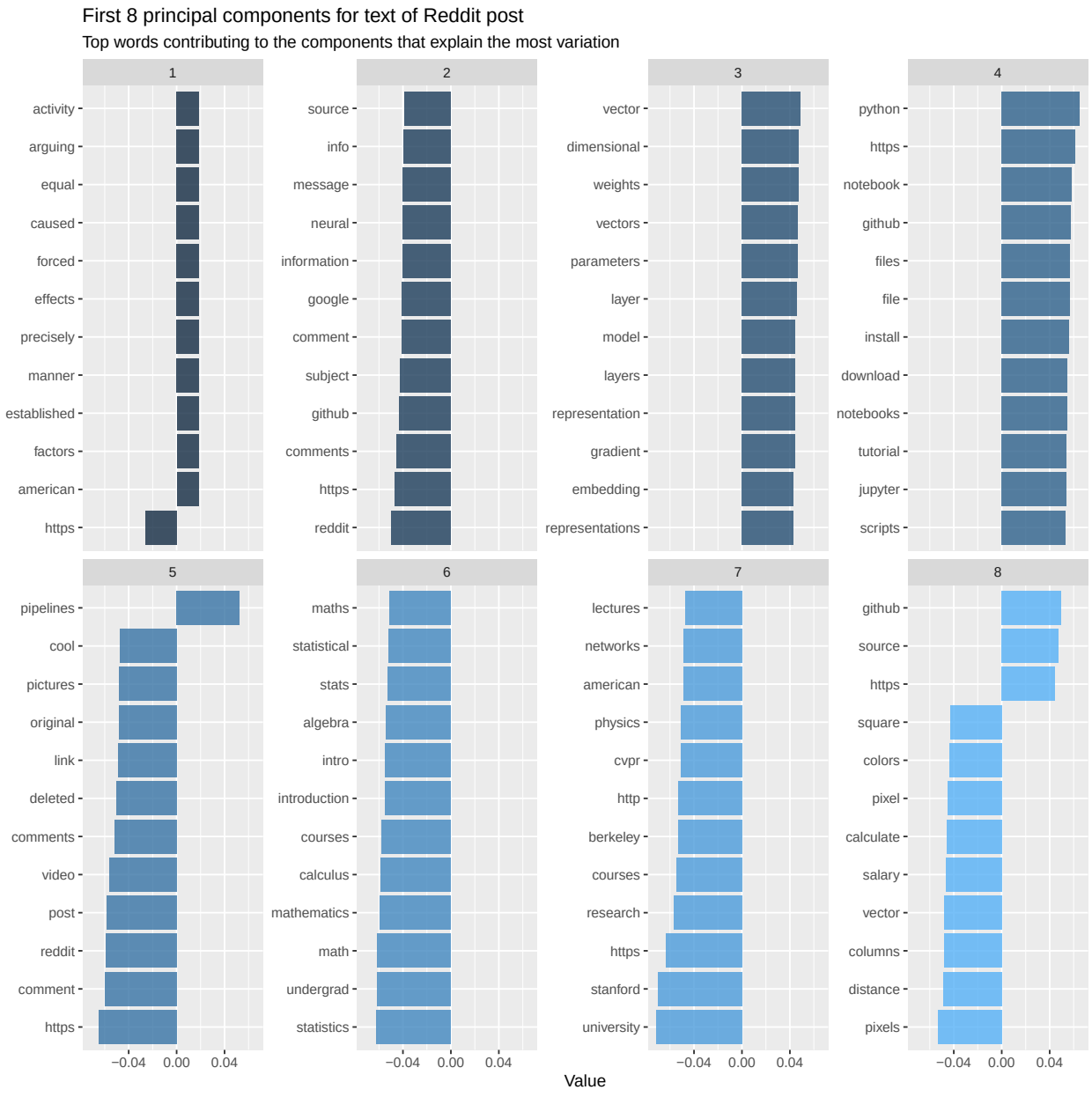

Figure 8: The produced principal components using SVD show tops words being used in the posts

4.6 LDA Model: 'Text as Data'

Latent Dirichlet Allocation (LDA) is a generative probabilistic model used in natural language processing and text analysis for uncovering hidden topics in a collection of documents. Introduced by David Blei, Andrew Ng, and Michael I. Jordan in 2003, LDA is a type of unsupervised machine learning algorithm that helps identify underlying structures in the data without any prior information about the documents' content [David Blei (2003)].

LDA is based on the assumption that documents are composed of a mixture of topics, and each topic is a probability distribution over a fixed vocabulary. In other words, LDA assumes that the words in a document are generated by a combination of latent topics, where each topic is represented by a distribution of words. The aim of LDA is to learn the latent topics and their distribution across documents.

To implement LDA, the following steps are typically followed:

1. Determine the number of topics (K) that you believe exist within the document collection.
2. Randomly assign each word in each document to one of the K topics.
3. For each document, update the topic assignments for its words by considering:
    a. The overall prevalence of each topic in the document (topic-document distribution).
    b. The overall prevalence of the word across all topics (word-topic distribution).
4. Iterate through step 3 multiple times, updating the topic assignments until they converge or a predefined stopping criterion is met.

After the model converges, you can interpret the topics by examining the most probable words associated with each topic. Additionally, you can compute the distribution of topics for each document, providing insights into the main themes present in the document collection.

It's important to note that LDA has some limitations:

- It assumes that the order of the words in the documents does not matter (bag-of-words assumption), which may not hold true in all cases.
- The choice of the number of topics (K) can significantly impact the results, and selecting the optimal K can be challenging.
- LDA may not always provide clear and interpretable topics, depending on the dataset and parameter settings.

Despite these limitations, LDA remains a widely used and powerful technique for topic modeling and extracting valuable insights from large text collections.

The subsequent activity is adapted from a chapter titled "Text as Data" in the book "Telling Stories with Data" by Rohan Alexander (Alexander 2022).

```
Beginning Spectral Initialization
     Calculating the gram matrix...
     Using only 10000 most frequent terms during initialization...
     Finding anchor words...
    ...
     Recovering initialization...
    ..................................................................................................
Initialization complete.
..................................................................................................
Completed E-Step (23 seconds).
Completed M-Step.
Completing Iteration 1 (approx. per word bound = -8.454)
```

```
.....................................................................................
Completed E-Step (23 seconds).
Completed M-Step.
Completing Iteration 2 (approx. per word bound = -8.201, relative change = 2.989e-02)
.....................................................................................
Completed E-Step (23 seconds).
Completed M-Step.
Completing Iteration 3 (approx. per word bound = -8.180, relative change = 2.590e-03)
.....................................................................................
Completed E-Step (25 seconds).
Completed M-Step.
Completing Iteration 4 (approx. per word bound = -8.171, relative change = 1.125e-03)
.....................................................................................
Completed E-Step (24 seconds).
Completed M-Step.
Completing Iteration 5 (approx. per word bound = -8.166, relative change = 5.692e-04)
Topic 1: can, just, time, good, really
 Topic 2: like, get, one, know, use
 Topic 3: data, people, think, https, also
.....................................................................................
Completed E-Step (25 seconds).
Completed M-Step.
Completing Iteration 6 (approx. per word bound = -8.163, relative change = 3.253e-04)
.....................................................................................
Completed E-Step (26 seconds).
Completed M-Step.
Completing Iteration 7 (approx. per word bound = -8.162, relative change = 2.031e-04)
.....................................................................................
Completed E-Step (24 seconds).
Completed M-Step.
Completing Iteration 8 (approx. per word bound = -8.161, relative change = 1.356e-04)
.....................................................................................
Completed E-Step (25 seconds).
Completed M-Step.
Completing Iteration 9 (approx. per word bound = -8.160, relative change = 9.534e-05)
.....................................................................................
Completed E-Step (25 seconds).
Completed M-Step.
Completing Iteration 10 (approx. per word bound = -8.159, relative change = 6.982e-05)
Topic 1: can, just, time, good, really
 Topic 2: like, get, one, know, use
 Topic 3: data, people, think, https, also
.....................................................................................
Completed E-Step (25 seconds).
Completed M-Step.
Completing Iteration 11 (approx. per word bound = -8.159, relative change = 5.280e-05)
.....................................................................................
Completed E-Step (25 seconds).
Completed M-Step.
Completing Iteration 12 (approx. per word bound = -8.158, relative change = 4.110e-05)
.....................................................................................
Completed E-Step (25 seconds).
Completed M-Step.
Completing Iteration 13 (approx. per word bound = -8.158, relative change = 3.284e-05)
```

```
..........................................................................................
Completed E-Step (25 seconds).
Completed M-Step.
Completing Iteration 14 (approx. per word bound = -8.158, relative change = 2.689e-05)
..........................................................................................
Completed E-Step (25 seconds).
Completed M-Step.
Completing Iteration 15 (approx. per word bound = -8.158, relative change = 2.253e-05)
Topic 1: can, just, time, good, really
 Topic 2: like, get, one, know, use
 Topic 3: data, people, think, https, also
..........................................................................................
Completed E-Step (25 seconds).
Completed M-Step.
Completing Iteration 16 (approx. per word bound = -8.158, relative change = 1.926e-05)
..........................................................................................
Completed E-Step (25 seconds).
Completed M-Step.
Completing Iteration 17 (approx. per word bound = -8.157, relative change = 1.674e-05)
..........................................................................................
Completed E-Step (26 seconds).
Completed M-Step.
Completing Iteration 18 (approx. per word bound = -8.157, relative change = 1.478e-05)
..........................................................................................
Completed E-Step (25 seconds).
Completed M-Step.
Completing Iteration 19 (approx. per word bound = -8.157, relative change = 1.328e-05)
..........................................................................................
Completed E-Step (25 seconds).
Completed M-Step.
Completing Iteration 20 (approx. per word bound = -8.157, relative change = 1.220e-05)
Topic 1: can, just, time, good, really
 Topic 2: like, get, one, know, use
 Topic 3: data, people, think, https, also
..........................................................................................
Completed E-Step (24 seconds).
Completed M-Step.
Completing Iteration 21 (approx. per word bound = -8.157, relative change = 1.147e-05)
..........................................................................................
Completed E-Step (23 seconds).
Completed M-Step.
Completing Iteration 22 (approx. per word bound = -8.157, relative change = 1.105e-05)
..........................................................................................
Completed E-Step (23 seconds).
Completed M-Step.
Completing Iteration 23 (approx. per word bound = -8.157, relative change = 1.093e-05)
..........................................................................................
Completed E-Step (23 seconds).
Completed M-Step.
Completing Iteration 24 (approx. per word bound = -8.157, relative change = 1.110e-05)
..........................................................................................
Completed E-Step (22 seconds).
Completed M-Step.
Completing Iteration 25 (approx. per word bound = -8.157, relative change = 1.147e-05)
```

```
Topic 1: can, just, time, good, really
 Topic 2: like, get, one, know, use
 Topic 3: data, people, think, https, also
..........................................................................................
Completed E-Step (22 seconds).
Completed M-Step.
Completing Iteration 26 (approx. per word bound = -8.157, relative change = 1.180e-05)
..........................................................................................
Completed E-Step (21 seconds).
Completed M-Step.
Completing Iteration 27 (approx. per word bound = -8.156, relative change = 1.179e-05)
..........................................................................................
Completed E-Step (20 seconds).
Completed M-Step.
Completing Iteration 28 (approx. per word bound = -8.156, relative change = 1.125e-05)
..........................................................................................
Completed E-Step (20 seconds).
Completed M-Step.
Completing Iteration 29 (approx. per word bound = -8.156, relative change = 1.028e-05)
..........................................................................................
Completed E-Step (20 seconds).
Completed M-Step.
Model Converged


Topic 1 Top Words:
     Highest Prob: can, just, time, good, really, even, s
     FREX: can, good, really, experience, thanks, right, now
     Lift: begat, baths, breeder, chewer, mender, enos, avdol
     Score: can, good, really, even, time, well, now
Topic 2 Top Words:
     Highest Prob: like, get, one, know, use, ai, work
     FREX: ai, like, google, one, code, know, use
     Lift: buffalo, 0035, 0078202577577, 0145394000, 014833888, 017133374, 028969191
     Score: like, one, get, know, use, ai, way
Topic 3 Top Words:
     Highest Prob: data, people, think, https, also, com, t
     FREX: com, r, reddit, https, data, python, model
     Lift: sneakpeekbot, adience, afw, animal10n, aqua, cacd, carpk
     Score: data, people, https, think, com, also, t
```

In light of the findings presented above, it is evident that numerous positive discussions related to the subjects: 'Deep Learning', 'Data Science', and 'Artificial Intelligence' have taken place on the platform. These results corroborate the conclusions drawn from previous sentiment analyses.

5 Discussion

5.1 Interpretation of results

The pirate plot in figure Figure 1 reveals a significant increase in post activity in the areas of Machine Learning, Artificial Intelligence, and Data Science between 2016 and 2018 compared to the period of 2013-2015. This surge in post activity is indicative of the growing interest and advancements in these fields.

The slight upward trend in the number of distinct words per post in the Machine Learning topic suggests that discussions are becoming more in-depth and diverse. Furthermore, the increase in total number of posts from 2019 to 2022 as shown in Figure 2 highlights the continued growth in these topics.

The upward trend in Machine Learning, as seen in Figure 3, is evidence of the significant growth experienced in the field between 2019 and 2021, which has persisted into 2023. Despite the overall negative polarity trend observed in Figure 4, Figure 5 shows that there is a predominantly positive and trust sentiment associated with Machine Learning.

Delving deeper into sentiment categories within specific topics like 'Career' and 'Discussion', we find interesting correlations. For instance, the increasing anticipation under the career category over time, as shown in Figure 6, might indicate that people are becoming more concerned about their jobs in the face of advancing technology. On the other hand, Bing sentiments in Figure 7 reveal many positive words within the 'Discussion' category, although some negative words like 'hard', 'wrong', and 'issue' are also present.

Figure 8 illustrates how common words related to data science, artificial intelligence, and deep learning are used together, reflecting the interconnectedness of these fields.

The top words analysis from Topic 1, Topic 2, and Topic 3 reveal that conversations on the platform are predominantly positive and focused on various aspects of Machine Learning, Artificial Intelligence, and Data Science. This further supports the findings from the sentiment analysis models, emphasizing the increasing interest and engagement in these areas over the years.

5.2 Implications for Machine Learning community

The implications for the Machine Learning community based on the analysis are as follows:

1. Growing interest and engagement: The increasing post activity and positive sentiment indicate that more people are joining the community and actively participating in discussions. This can lead to a more diverse and inclusive environment, fostering collaboration and innovation.

2. Expanding knowledge base: The upward trend in the number of distinct words per post suggests that conversations are becoming more in-depth and covering a wider range of topics. This can help expand the collective knowledge base and facilitate the development of new ideas and techniques in Machine Learning.

3. Focus on applications and impact: The increasing anticipation in the 'Career' category and the interconnectedness of common words related to data science, artificial intelligence, and deep learning highlight the growing awareness of the real-world applications and implications of Machine Learning. The community can leverage this understanding to address potential challenges and explore new opportunities across various industries.

4. Need for continuous learning and adaptation: As Machine Learning continues to evolve and integrate with other fields, it is essential for community members to stay up-to-date with the latest research and advancements. This will help them adapt to the changing landscape and ensure that their skills remain relevant in the job market.

5. Importance of addressing concerns and misconceptions: The negative polarity trend observed in some cases underscores the need for the Machine Learning community to address concerns and misconceptions surrounding the technology. By engaging in open and honest discussions, the community can build trust and promote a better understanding of Machine Learning's potential benefits and limitations.

6. Collaboration and interdisciplinary research: The synergistic relationship between data science, artificial intelligence, and deep learning emphasizes the importance of collaboration and interdisciplinary research. By working together and drawing from diverse perspectives, the Machine Learning community can drive innovation and unlock new potential applications.

5.3 Limitations of the study

There are several possible limitations of the study that should be considered:

1. Data source: The study have relied on data from a single platform or a limited number of sources. This could result in a biased representation of the Machine Learning community and its discussions. Including data from a wider range of sources and platforms could provide a more comprehensive view.
2. Temporal limitations: The study's data only covered a specific period, potentially it will missing out on more recent developments or trends in the Machine Learning community in the future. Continuous updates to the data and analysis would be necessary to maintain relevance and accuracy.
3. Sentiment analysis limitations: Sentiment analysis is often based on algorithms that may not accurately capture the nuances and complexities of human emotions and language. Sarcasm, irony, and context can be challenging for these algorithms to interpret, which could result in misclassification of sentiment.
4. Topic modeling limitations: The study's topic modeling approach might not perfectly capture the underlying themes and topics discussed within the community. The choice of parameters, the number of topics, and the algorithm itself could influence the results, leading to a different representation of the data.
5. Limited scope: The study might focus on a specific aspect of Machine Learning, Artificial Intelligence, or Data Science, potentially overlooking other important factors or areas of interest. A broader scope could provide a more comprehensive understanding of the community and its discussions.
6. Generalizability: The findings of the study might not be generalizable to other communities or fields. Differences in culture, language, and context could limit the applicability of the results to other areas.
7. Potential biases: The study might be subject to biases, such as selection bias, confirmation bias, or researcher bias. These biases could influence the interpretation of the data and the conclusions drawn from the analysis.

To address these limitations, future research could incorporate data from diverse sources, continuously update the dataset, refine sentiment and topic modeling approaches, broaden the scope of the study, and rigorously assess potential biases to provide a more accurate and comprehensive understanding of the Machine Learning community and its discussions.

6 Conclusion

In conclusion, the analysis of the various figures and sentiments demonstrates a significant growth in interest and engagement in the fields of Machine Learning, Artificial Intelligence, and Data Science over the years. The increase in post activity, distinct words per post, and predominantly positive sentiment reveal a deeper understanding and appreciation of these technologies and their potential applications.

The interconnectedness of the common words related to these fields underscores their synergistic relationship, which will likely continue to drive further advancements and breakthroughs. Moreover, the increase in anticipation within the 'Career' category suggests that people are both excited and concerned about the potential impact of these technologies on the job market.

The growing interest and engagement in Machine Learning, Artificial Intelligence, and Data Science reflect the recognition of their importance in shaping the future of various industries. As these technologies continue to advance, we can expect even more in-depth and diverse conversations, fostering greater understanding, collaboration, and innovation in these areas.

References

Alexander, Rohan. 2022. *Telling Stories with Data*. https://www.tellingstorieswithdata.com/index.html.

Alto, Valentina. 2020. "Understanding Pointwise Mutual Information in NLP." *Medium*. https://medium.com/dataseries/understanding-pointwise-mutual-information-in-nlp-e4ef75ecb57a.

Becker, Chambers, R. A., and A. R. WLLKS. 1998. *The New s Language*. https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/grep.

Bollen, Johan, Huina Mao, and Xiaojun Zeng. 2011. "Twitter Mood Predicts the Stock Market." *Journal of Computational Science* 2 (1): 1–8. https://doi.org/https://doi.org/10.1016/j.jocs.2010.12.007.

"Cholesky Decomposition: Matrix Decomposition." n.d. *Geeks for Geeks*. https://www.geeksforgeeks.org/cholesky-decomposition-matrix-decomposition/.

Dave, Kushal and Lawrence, Steve and Pennock, David M. 2003. "Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews." In *Proceedings of the 12th International Conference on World Wide Web*, 519–28. WWW '03. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/775152.775226.

David Blei, Michael I. Jordan, Andrew NG. 2003. "Latent Dirichlet Allocation."

earney, Michael W. 2019. "Rtweet: Collecting and Analyzing Twitter Data." https://doi.org/http://dx.doi.org/10.21105/joss.01829.

Gu, Zuguang, Lei Gu, Roland Eils, Matthias Schlesner, and Benedikt Brors. 2014. "Circlize Implements and Enhances Circular Visualization in r." *Bioinformatics* 30: 2811–12.

Jockers, Matthew. 2020. "Syuzhet: Introduction to the Syuzhet Package." In. https://cran.r-project.org/web/packages/syuzhet/vignettes/syuzhet-vignette.html.

Pang, Bo, and Lillian Lee. 2008. "Opinion Mining and Sentiment Analysis" 2 (1–2): 1–135. https://doi.org/10.1561/1500000011.

Phillips, Nathaniel. 2017. *Yarrr: A Companion to the e-Book "YaRrr!: The Pirate's Guide to r"*. https://CRAN.R-project.org/package=yarrr.

R Core Team. 2020. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Robinson, David, and Julia Silge. 2022. *Widyr: Widen, Process, Then Re-Tidy Data*. https://CRAN.R-project.org/package=widyr.

Sarlan, Aliza and Nadam, Chayanit and Basri, Shuib. 2014. "Twitter Sentiment Analysis." In *Proceedings of the 6th International Conference on Information Technology and Multimedia*, 212–16. https://doi.org/10.1109/ICIMU.2014.7066632.

Silge, Julia, and David Robinson. 2016a. "Tidytext: Text Mining and Analysis Using Tidy Data Principles in r." *Journal of Open Source Software* 1 (3): 37. https://doi.org/10.21105/joss.00037.

———. 2016b. "Tidytext: Text Mining and Analysis Using Tidy Data Principles in r." *JOSS* 1 (3). https://doi.org/10.21105/joss.00037.

"Singular Value Decomposition." n.d. *Geeks for Geeks*. https://www.geeksforgeeks.org/singular-value-decomposition-svd/.

"The Python Reddit API Wrapper." 2016. https://praw.readthedocs.io/en/stable/.

Tumasjan, Andranik, Timm Oliver Sprenger, Philipp G. Sandner, and Isabell M. Welpe. 2010. "Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment." *Proceedings of the International AAAI Conference on Web and Social Media*.

Vaughan, Davis. 2022. *Slider: Sliding Window Functions*. https://CRAN.R-project.org/package=slider.

Vaughan, Davis, and Matt Dancho. 2022. *Furrr: Apply Mapping Functions in Parallel Using Futures*. https://CRAN.R-project.org/package=furrr.

Wickham, Hadley. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. https://ggplot2.tidyverse.org.

Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. 2021. *Dplyr: A Grammar of Data Manipulation*. https://CRAN.R-project.org/package=dplyr.

Wijeratne, Balasuriya, s. 2017. "EmojiNet: An Open Service and API for Emoji Sense Discovery." https://doi.org/https://doi.org/10.48550/arXiv.1707.04652.

Wnag, Chen, Y. 2019. "A Comparative Study of Machine Learning Algorithms and Their Applications." https://doi.org/https://doi.org/10.1109/ICRITO.2018.8748793.