Humorous and Professional – A Dive Into Social Media Text Classification

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science
in
Computational Linguistics
by Research

by

TANISHQ CHAUDHARY
2019114007
tanishq.chaudhary@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
May 2024

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled “**Humorous and Professional - A Dive Into Social Media Text Classification**” by **Tanishq Chaudhary**, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date

Adviser: Dr. Radhika Mamidi

To my parents and room 333

Acknowledgments

I would like to thank everyone who has been a part of my journey at IIIT-Hyderabad.

This thesis would not be possible without the immense support of my parents through these last 5 years. They were always with me, with their love and support. Without them, I would not have the success I have today. I thank my dad for the best advice always and my mom for always putting a smile on my face.

I would like to thank Dr. Radhika Mamidi, who supported me throughout my journey. Starting from mentoring a course project in the third semester, leading to a paper in the fourth. She has always been extremely supportive and understanding. Whenever I had a doubt or needed advice, she was always there to help me out. The biggest reason for my early research completion is the freedom she gave me to work at my own pace.

I would like to thank Mayank Goel and Pulak Malhotra for being amazing research partners. My work would not have been possible without their unique ways of thinking and our chemistry together.

I would also like to thank my friends, Shivansh Subramanian, Zeeshan Ahmed, Kushagra Garg and the entire room 333 for all the good times. Special thanks to Sahithi for sitting beside me as I write this thesis.

Abstract

Social media is now a deep-rooted part of our daily lives. Whether for entertainment or information acquisition, it serves as a communication hub for billions of users. In this thesis, we dive into the realm of text classification by taking a two-fold approach, namely, contrasting text in professional and humorous fields. First, we understand the nuances of human communication via a previously unexplored social media platform, Blind. Next, we identify how the nuances of human communication are exploited by looking at humor.

Our aim is to conduct a thorough analysis of these contrasting worlds to demonstrate that they work on the same underlying structures and goals. This provides a comprehensive analysis of the landscape within social media.

In the non-humorous domain, Blind has emerged as an anonymous platform with the unique goal of satisfying the growing need for taboo workplace discourse. Employees come on the platform to discuss issues ranging from layoffs, compensation, interview advice, career progression and more. In our work, for the first time, we explore the platform in detail by scraping and analyzing two datasets: **767,224** *Blind Posts* and **63,477** *Blind Company Reviews* containing seven years of industry data. Using the *Blind Posts* dataset, we dissect the popular discussion topics of employees, find mappings of global events like work-from-home, return-to-office, and layoffs, and aggregate the sentiments of the platform for a comprehensive temporal analysis. We then propose our novel content classification pipeline. We first filter relevant content with an accuracy of 99.25% and then further annotate relevant textual context into ten categories with an accuracy of 78.41% based on the *Blind Posts*. Using the *Blind Company Reviews*, we conduct content and metrical analyses on the data for a complete view of the platform and complete our novel content classification pipeline, by adding the ability to mine opinions of employees, with an accuracy of 98.29%.

For the humor domain, we utilize the *Short Jokes* dataset which has data from *r/jokes* and *r/cleanjokes* subreddits on Reddit, totaling **231,657** text jokes. After getting the humorous data, we use linguistically motivated features inspired by the Incongruity theory of humor and the General Theory of Verbal Humor (GTVH). These features allow us to consider humor instruments from the phonetic level to the pragmatic level, considering things like alliteration chain lengths, text polarity, slangs, etc. We train multiple machine learning and transformer models and achieve an accuracy of 63% and 98.90%, respectively. To understand the rift in the results better, we analyze the style and the semantics of the text in detail.

Finally, we formalize the results across tasks and explain the consistently superior results of transformers. We finally gain valuable insights into the common underlying structure of text classification tasks.

Contents

List of Figures

Figure Page

List of Tables

Chapter 1

Introduction

In this thesis, we expand our understanding of text classification by considering two diametrically opposing human communication domains. We consolidate our linguistically driven and technically robust explorations of the entertainment and the professional world.

This chapter introduces the reader to our research by describing this thesis's motivations, contributions, and organization.

1.1 Motivation: Social Media

In the last two decades, we have seen a tremendous rise in internet use. In 2022, social media collectively boasted a user base of 4.76 billion, with each user spending 150+ minutes daily.[1] Online Social Networks (OSNs) have effectively converted our limited offline social capabilities into a digital form with exponential reach [1].

1.1.1 Industry on Blind

Our work on Blind takes on the challenge of analyzing an unexplored social network and designing a custom content classification system to filter and annotate the entire social media internet for relevant content.[2]

Blind was our selected platform of choice for several reasons. Whether one is a student looking to join the corporate world or someone already in the industry looking to make a switch, there is a need for *honest* professional opinion and advice on topics like career growth, interview tips, etc. Moreover, one needs to know about individual companies as well – how do companies fare on critical metrics like compensation, work-life balance, etc. Blind has emerged as the leading social media platform, with over 7M+ employees and 300k+ companies.[3] A sample of how the Blind homepage looks is shown in figure

[1] https://datareportal.com/reports/digital-2023-global-overview-report

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

[3] https://www.teamblind.com/whyBlind

1.1. Due to the popularity of tech companies, Blind is now almost exclusively used by employees in tech, allowing us to analyze the roots of the tech industry. We collect Blind's posts and company reviews for relevant data.[4] We call the datasets *Blind Posts* and *Blind Company Reviews*, respectively.

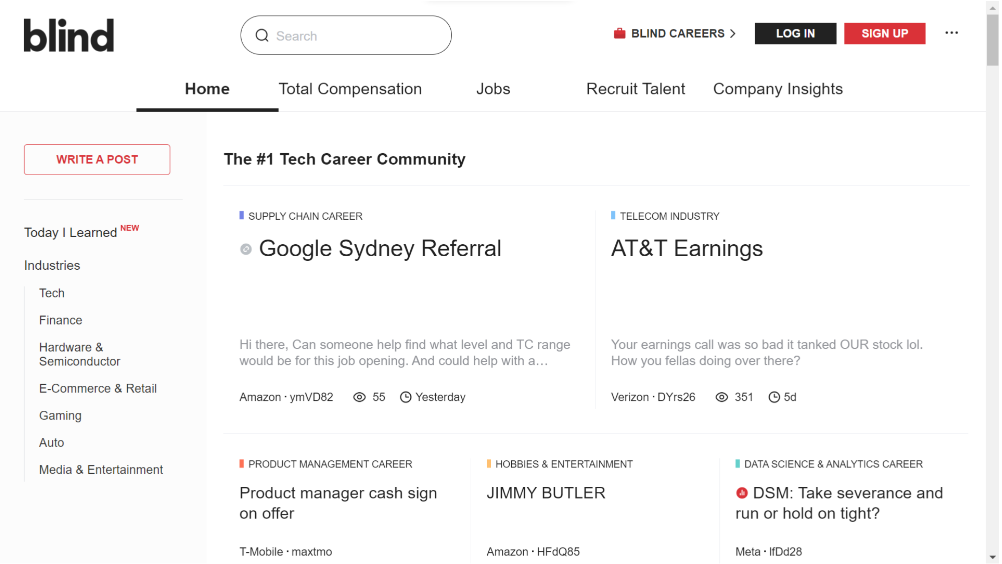
We use Blind for the following reasons.

- Blind is specific to career-related discussions. This is unlike Quora, X, subreddits on Reddit, and other social media websites with a plethora of non-career posts. Blind allows us to work with non-diluted career textual content.

- Blind does not have fake company reviews. Glassdoor is another popular website where people post company reviews.[5] However, it suffers from the problem of fake reviews [2]. Blind, on the other hand, has a strict verification check, allowing one to register, view, post, and comment only if they log in with a company e-mail ID.

- Blind is brutally honest. Because of anonymity, people on Blind are not scared to paint their employers in a negative light, if needed. This is in sharp contrast to LinkedIn, where people generally restrict their discussions to sharing positive news – getting an internship or a job offer, and so on [3, 4].

The Blind platform unlocks the door for classifying tech versus non-tech related content using the *Blind Posts*, allowing its extensions and applications to other social media platforms to filter out tech-related content. Moreover, using the *Blind Company Reviews*, we can further mine and understand opinions.



The #1 Tech Career Community

SUPPLY CHAIN CAREER
Google Sydney Referral
Hi there, Can someone help find what level and TC range would be for this job opening. And could help with a...
Amazon · ymVD82 55 Yesterday

TELECOM INDUSTRY
AT&T Earnings
Your earnings call was so bad it tanked OUR stock lol. How you fellas doing over there?
Verizon · DYrs26 351 5d

PRODUCT MANAGEMENT CAREER
Product manager cash sign on offer
T-Mobile · maxtmo

HOBBIES & ENTERTAINMENT
JIMMY BUTLER
Amazon · HFdQ85

DATA SCIENCE & ANALYTICS CAREER
DSM: Take severance and run or hold on tight?
Meta · IfDd28

Figure 1.1: A view of the Blind homepage

[4] All the data collection was done on and before 21 December 2022.

[5] `https://www.glassdoor.co.in/index.htm`

1.1.2 Humor on Reddit

Next, our work takes on the monumental challenge of identifying what makes a joke a joke. Understanding humor is a cornerstone of advancing Artificial Intelligence since it works not on the formal use of language but instead on exploiting the nuances of our communications [5, 6].

To identify whether a text is humorous, we need an annotated dataset with both labels for a supervised approach. While the non-humorous data is abundant, we turn to Reddit for the humorous part of the equation. Reddit, also called "the front page of the internet", is a social media content aggregation website (Figure 1.2).[6] It has over 430 million active monthly users and 52 million daily active users.[7] Reddit thrives because it has over 3.4 million subreddits.[8] Each subreddit is a separate discussion forum for every possible topic on Earth, inviting a vast variety of people. The subreddits range from *r/funny* (humorous side of Reddit) to *r/shittyprogramming* (talks about programming practices, humor related to code, etc.), *r/fatpeoplestories* (discussions about funny, embarrassing stories related to fat people) and everywhere in between.

In fact, *r/funny* is the biggest subreddit, with over 54 million subscribers.[9] People love to use this subreddit to share copy-pasted memes, funny videos, and stolen jokes. Even if we are looking at purely text jokes, there are 231k+ short jokes present as the *Short Jokes* dataset on Kaggle, taken from *r/jokes* and *r/cleanjokes*.[10]

We use Reddit's textual humor as the basis for our linguistically motivated experiments to spotlight the "jokiness" of a joke.

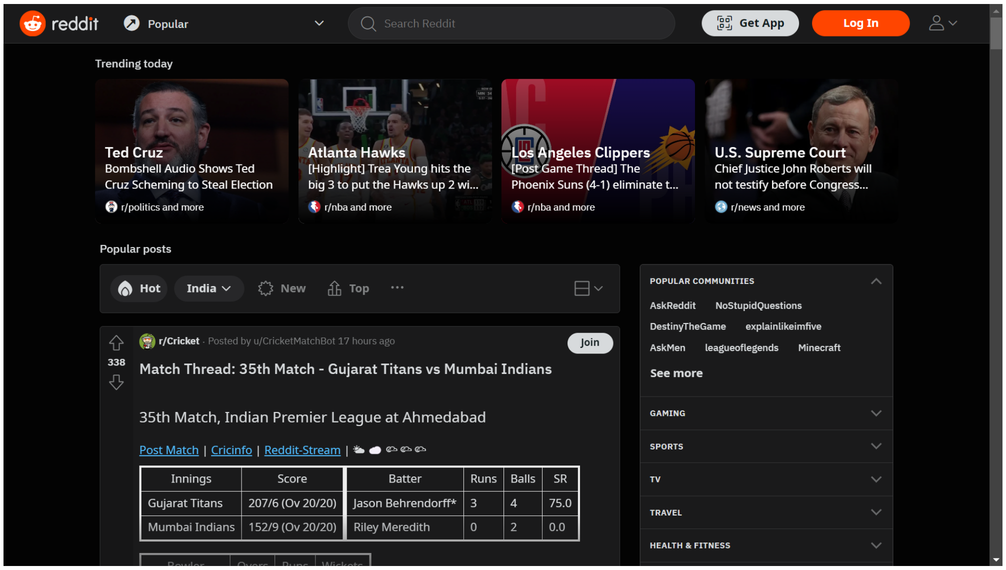
[6] `https://www.reddit.com/`

[7] `https://earthweb.com/how-many-people-use-reddit/`

[8] `https://www.businessdit.com/how-many-subreddits-are-there`

[9] `https://www.reddit.com/best/communities/1/`

[10] `https://www.kaggle.com/datasets/abhinavmoudgil95/short-jokes`



Ted Cruz Bombshell Audio Shows Ted Cruz Scheming to Steal Election — r/politics and more

Atlanta Hawks [Highlight] Trea Young hits the big 3 to put the Hawks up 2 wi... — r/nba and more

Los Angeles Clippers [Post Game Thread] The Phoenix Suns (4-1) eliminate t... — r/nba and more

U.S. Supreme Court Chief Justice John Roberts will not testify before Congress... — r/news and more

r/Cricket · Posted by u/CricketMatchBot 17 hours ago

Match Thread: 35th Match - Gujarat Titans vs Mumbai Indians

35th Match, Indian Premier League at Ahmedabad

Post Match | Cricinfo | Reddit-Stream |

Innings	Score	Batter	Runs	Balls	SR
Gujarat Titans	207/6 (Ov 20/20)	Jason Behrendorff*	3	4	75.0
Mumbai Indians	152/9 (Ov 20/20)	Riley Meredith	0	2	0.0

Bowler	Overs	Runs	Wickets

Figure 1.2: A view of the Reddit homepage

The two tasks of identifying and understanding humor and industry discussions look dissimilar on the surface. This thesis aims to conduct a thorough analysis of the humorous and non-humorous domains that demonstrate that they work on the same underlying structures and goals. In both cases, text classification aims to comprehend the nuances of human language, distinguishing between various tones, sentiments, and intents. The common thread lies in accurately recognizing and interpreting textual cues for amusement or informed decision-making. The dissimilarity turns into synergy, as we can understand not only what makes honest discussions but also see communication breaking down.

1.2 Contributions of the thesis

We first dive into the professional world by analyzing the content of the Blind platform.

- We collected all the *Blind Posts* from the inception of the Blind website, 20 October 2015, till 21 December 2022, by utilizing its public API. This amounts to **767,224** posts from 74 boards.

- We extract insights into the tech industry in the context of COVID-19, the layoffs and the new work-from-home, and, consequently, the return-to-office phenomenon.

- To identify tech-related posts anywhere on the internet, we train a binary coarse classification model with **99.25%** accuracy by merging the cleaned *Blind Posts* data with an equally large Reddit TL;DR dataset [7], totaling to **1,328,096** data points.

- Further, we enhance our content classification pipeline by training models to recognize the top ten boards (by posts) on Blind and get **78.41%** accuracy.

- We scrape another dataset, *Blind Company Reviews*, with **63,477** company reviews from the **55** most reviewed companies on Blind. We exploit the pro and con fields in the review to get automatically labeled data for classification experiments and achieve an accuracy of **98.29%** for mining opinions, completing our content classification pipeline.

Then, we explore the world of humor by analyzing the jokes from Reddit.

- We conduct linguistically motivated experiments for the first time on the Reddit Short Jokes dataset from (*r/jokes* and *r/cleanjokes*).
- We extract features on all levels of linguistic analysis: morpho-syntactic, lexico-semantic, pragmatic, affective, and improve over past works, with models achieving an accuracy of **63%**.
- We consolidate our findings with further experiments using transformer models, achieving an accuracy of **98.9**% with the RoBERTa model, to pinpoint the reasons for the disparity in results.

Finally, for each of the three tasks: *Blind Posts* classification, *Blind Company Reviews*, and humor classification, we provide detailed discussions and analyses for model performances and nuances of the data.

1.3 Organization of the thesis

The thesis is organized into six chapters, as follows.

Chapter 2: Related Work In this chapter, we explore the limited previous works on the Blind platform and the theories of humor driving our analyses in the later chapters. We also examine how social-media-related texts are processed and what models we use for our text classification experiments.

Chapter 3: Blind Posts Classification In this chapter, we explain how we acquired the *Blind Posts* data, then explain the general statistics of the dataset for a complete picture. We find trends in the tech-industry landscape due to the pandemic and other landmark events. Then, we explain our novel content classification pipeline and train and discuss nuances of the platform from the model results.

Chapter 4: Blind Company Review Classification In this chapter, we explain how we acquired the *Blind Company Reviews* data, then comprehensively analyze company metrics like work-life balance, compensation, and more. Finally, we complete the content classification pipeline.

Chapter 5: Humor Classification In this chapter, we dive deeper into the realization of humor theories in practice, looking at the Reddit dataset. We explore linguistic feature extraction at various morphological, syntactic, semantic, pragmatic, and affective levels. Finally, we utilize the machine learning and transformer models and obtain insights into automatic humor classification.

Chapter 6: Conclusion and Future Work In this chapter, we consolidate our understanding of text classification from a social media perspective and look at future directions and applications possible in these fields.

Chapter 2

Related Work

In this chapter, we explore the past work done in context of the professional domain as well as the humorous. Following our aim of finding the synergy between the two contrasting domains, humor, and, non-humor, we follow a common pipeline of text processing, vectorizing, and, modeling.

2.1 Social Media Platforms

2.1.1 Reddit

There is a plethora of work on the Reddit social media platform, ranging from cyberbullying detection [8] and mental health discussions, detection, and classification [9, 10]. Although it has seen some light in recent years, work specifically on Reddit humor is limited.

Authors have applied different deep learning and transformer based architectures on the dataset, *Short Jokes*, with varying degrees of successes.[1] For example, authors have extensively experimented with methods using Recurrent Neural Network (RNN) to consider the sequential nature of text and improved them using Long Short-Term Memory (LSTM) neural networks in order to consider longer ranged dependencies within a text. Depending upon the contrasting dataset, that is, one used to label non-humorous data, accuracies range from 74.20% for LSTMs with dataset as Reuters, 52.2% with BNC, and, 55.6% with Proverbs [11, 12, 13].[2] Recent works have applied transformers to the same, and achieved upto 98.6% accuracies with a vanilla architecture and 98.2% while passing sentences in parallel to transformers to deeper layers and finally concatenating them [14, 15].

While the works showcase improving accuracies of the models over the years, a deeper discussion on the "why" behind these models is often lacking.

In our work, for the first time, we conduct analyses on the Reddit dataset with a linguistically motivated feature-based approach. We pull out and compare explicit features to what makes something funny. Going beyond, we outperform the current (state-of-the-art) SOTA transformer models and explore the

[1] https://www.kaggle.com/datasets/abhinavmoudgil95/short-jokes

[2] https://www.thomsonreuters.com/en.html

nuances of what truly makes something funny. We compare our linguistic results to other linguistically backed works on one-liners [16] and on Twitter data using an extensive list of features [17]. Finally, we dive deeper into critical reasons why the transformer model performs better than the past models.

2.1.2 Blind

Past work on the Blind social media platform is virtually non-existent, and only the anonymity aspect of the platform has been explored. The authors conduct a poll for Microsoft employees on the perceptions and uses of anonymity in IT organizations [18]. They extend their work further and find communication qualities and freedom of speech at work play a significant role in the work environment [19].

To the best of our knowledge, there is no other work on Blind focusing on large-scale platform characterization and text-based analyses.

2.2 Humor Theories

The work on humor classification in chapter 5 of this thesis relies upon the ideas set by the Incongruity theory and the General Verbal Theory of Humor, as explained below. All of the examples used to explain the theories are taken from the *Short Jokes* dataset we are working with.

2.2.1 Incongruity Theory

The incongruity theory of humor is the prevailing dominant theory, based on the idea of violation of expectation. The first well-known account of the theory is given by Aristotle, who explains that one of the ways to generate humor is to create an expectation in the audience and then violate it [20]. Numerous later philosophers have taken up this idea. This approach is still seen in modern comedy, whether it be on stage or on social media platforms. A joke is divided into a setup and a punchline. The "setup" *sets up* the expectation, which is then broken by the punchline – forcing the audience to backtrack and re-evaluate a joke.

"9/11 Jokes aren't funny.
The other 2 however, are hilarious!"

The above is an example of two different "ideas" coming into conflict with each other. The first line is serious, based on the "9/11" attacks in the United States of America. However, the second line forces us to reinterpret the first line since it says that nine jokes out of every 11 are hilarious, while the remaining two are not. At the heart, the joke exploits the literal meaning of "9/11," which means 9 "out of" 11, and the more commonly known interpretation, based on world knowledge, to remind us of the shocking event.

Similarly, consider the following joke, where the first part sets up an expectation that holding a funeral for a lost loved one is hard due to the emotional burden. However, the humor comes from the second part, where the wife is still alive and pesters her husband with questions, making it hard for him to arrange the funeral.

"Making the arrangements for my wife's funeral is tough.
She keeps asking what I'm doing"

This theory has been explored further by many other philosophers and psychologists. The incongruity of humor works at all levels and can manifest itself starting from the level of phones and going all the way up to the level of pragmatics.

2.2.2 General Theory of Verbal Humor

The General Theory of Verbal Humor (GTVH) was proposed by Victor Raskin and Salvatore Attardo in their article, "Script theory revis(it)ed: joke similarity and joke representation model" [21]. This theory uses 6 Knowledge Resources (KRs) to understand and model the jokes, as enumerated below.

2.2.2.1 Language

The first parameter to model a joke is noting the use of language. Specifically, looking at the choices made at "phonetic, phonologic, morphophonemic, morphologic, lexic, syntactic, semantic, and pragmatic levels". The idea is that the content of the joke remains the same, and there can be multiple "verbalizations" of the joke – each essentially being a paraphrase of the other. The most important factor that defines a joke is the punchline and the content of the joke works towards making the line "punch".

2.2.2.2 Narrative Strategy

The layout of a joke also makes a difference [22]. While the content of the two below jokes is the same, one is expository, while the other is set up as a question-answering sequence.

"Tweets are like your children: you love them all at first, you never know how they'll age, and most of them you regret creating."

"How are Tweets like your children?
How?
You love them all at first, you never know how they'll age, and most of them you regret creating."

2.2.2.3 Target

This parameter of a joke defines who is the target of the joke – a person/group of people [23]. The constant is the association of a stereotype associated with the target, regardless of reality. Interestingly, this is the only optional parameter; not all jokes have a "butt". For example, the following joke uses the stereotype that Jews are miserly, and so the father keeps on reducing the amount of money being discussed. Whether one believes the stereotype or not, we can agree there is a certain "jokiness" to the following.

"A young Jewish boy asks his father if he can borrow $50...
His father replies: 40 dollars! What could you possibly need to borrow 30 dollars for?!?"

2.2.2.4 Situation

The assumption with this parameter is that all jokes are about *something*. To define the situation, the joke uses certain props, whether they be people, things, places, etc. Note that the audience should be aware of the props. The following joke uses a standard "your mom" or "yo mama" template. The props used are iPhone and iPad, and the audience should know about the dimensions of the devices to make sense of the joke – without which the joke would not be funny.

"Yo mama so fat, she sat on my iPhone and turned it into an iPad"

2.2.2.5 Logical Mechanism

This parameter of a joke defines how different scripts interact with each other [24]. One case is that of "chiasmus", where the grammatical structures or concepts are reversed from the first to the second phrase or clause. In the following example, the first line says that the person does not believe in bigfoot, a mythical ape-like animal with debated existence. The second line reverses the concept of belief by saying that the reason the author does not believe in Bigfoot is that he never believed in the author.

"I don't believe in Bigfoot;
because he never believed in me."

2.2.2.6 Script Opposition

This parameter is based on the Script-based Semantic Theory of Humor (SSTH) [25]. The theory works at three levels of abstraction. In the topmost layer, we have a contrasting setup between real and unreal. In the middle layer, we have actual vs. non-actual, normal vs. abnormal, and possible vs. impossible. In the last layer, we have good vs. bad, life vs. death, sex vs. nonsex, money vs. no money, and high stature vs. low stature.

Many humor theories can explain some jokes better than others and no theory is either complete or perfect. However, humor itself is hard to define and pinpoint – it is a complex phenomenon with social, emotional, and cultural aspects.

2.3 Text Classification

2.3.1 Text Preprocesssing

The first step for any text classification task is to preprocess the data to ensure consistency. We preprocess it using the following steps.

1. Lowercase the text.
2. Tokenize using the NLTK's TweetTokenizer to preserve all hashtags and emojis [26].
3. Remove all the punctuations.

These steps allow us to streamline further evaluations for the machine learning models. For the transformer models, we use their respective tokenizers.

2.3.2 Vectorization

The next step is to convert the processed text into a numerical vector representation to feed to the models. Vectorizing allows us to represent simple and complex sentences alike in the form of dense meaning embeddings in the same higher dimensional space. We use the two following methods for vectorization.

2.3.2.1 Tf-Idf

TF-IDF, which stands for Term Frequency-Inverse Document Frequency, is commonly used in information retrieval systems [27]. There are three parts involved in calculating the Tf-Idf score.[3]

1. $tf(t, d)$ represents the *relative* frequency of term t in the document d with $f_{t,d}$ representing the raw count, as seen in the equation 2.1.
2. $idf(t, D)$ represents the *information* of term t in the the set of documents D and $N = |D|$, as seen in the equation 2.2.
3. $tfidf(t, d, D)$ is thus defined by multiplying both $tf(t, d)$ and $idf(t, D)$.

[3] Note that all of the equations above have minor implementation-dependent changes. We use the variant as described.

$$tf(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \tag{2.1}$$

$$idf(t,D) = \log\left(\frac{1+N}{1+|d \in D : t \in d|}\right) + 1 \tag{2.2}$$

$$tfidf(t,d,D) = tf(t,d) \cdot idf(t,D) \tag{2.3}$$

For the text classification task, Tf-Idf is helpful in two major ways. Firstly, it helps to reduce noise in the text. The higher frequency words like stop words (for example, “the”, “an”, “a”) do not contribute significantly to the text. Due to the high presence of these words in all documents, their idf is very low, reducing their contribution to sentence meaning. Secondly, for the same reasons, words that are unique or distinctive are boosted up in their sentence meaning contribution.

2.3.2.2 SBERT

To compare purely statistical approaches like Tf-Idf with embeddings designed to capture semantic knowledge, we employ SBERT [28]. SBERT stands for Sentence-BERT, a variant of BERT (explained in the following sections). The idea behind SBERT is to apply the BERT pre-trained model and fine-tune it on a sentence similarity task. The architecture is set up using Siamese and triplet networks, and it learns to embed the sentences in a space with an understanding of how similar or dissimilar the sentences are. We use two model variants, SBERT all-MiniLM-L6-v2 (80MB) and SBERT all-mpnet-base-v2 (420MB). This allows us to compare model sizes and see their impacts on the results.

2.3.3 Machine Learning Models

To form baselines for our approaches, we start our experiments with machine learning models. They form a solid baseline as they require a lot less data to perform at par with deep learning models. Moreover, they let us better judge the complexity of the task with the computational requirements.

2.3.3.1 Logistic Regression

At its heart, Logistic Regression uses the logistic function, also called the sigmoid function, as shown in the equation 2.4 [29]. The goal is to make the model learn a weight vector, which can be used for prediction on unseen data. We use the log loss function to train the model, given by the equation 2.5, where N is the number of classes, y is the true probability distribution, and $\hat{y}$ is the predicted probability distribution.

Note that Logistic Regression is a binary classification model, but it can also be extended to multiple classes. In the case of multi-class classification, we use the cross-entropy loss function to train the model, given by the equation 2.6.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.4}$$

$$L(y, \hat{y}) = -y \cdot \log(\hat{y}) - (1 - y) \cdot \log(1 - \hat{y}) \tag{2.5}$$

$$L(y, \hat{y}) = -\sum y \cdot \log(\hat{y}) \tag{2.6}$$

2.3.3.2 Linear Support Vector Classifier

Linear Support Vector Classifier (SVC) is another classification model, a variant of the Support Vector Machine (SVM) [30]. Once the features are given to the model, its goal is to create a separating hyperplane in a higher dimension. For example, if the data is in the d-dimensional space, the model raises it to $d + 1$ dimensions to separate it using the hyperplane. In the case of multiple classes, this model uses the ovr (one-versus-rest) scheme.

Mathematically, it solves the optimization equation as given in 2.7. w Represents the weight vector or the model parameter that will be trained. It is trained with the bias term, b. C represents the weighted penalty factor used to discourage making the errors for each ith data point (ϵ_i). The y_i and x_i are the ith data points. Note that for all the implementations in our work, we use the $l2$ penalty term and the squared hinge loss.

$$\begin{gathered} \text{minimize} : 0.5 \cdot \|\mathbf{w}\|_2^2 + C \sum_i \epsilon_i \\ \text{subject to} : y_i(\mathbf{w}^T \cdot x_i + b) \geq 1 - \epsilon_i \ \forall i \end{gathered} \tag{2.7}$$

2.3.3.3 Gaussian/Multinomial Naïve Bayes

The core of these algorithms is based on the Bayes theorem [31, 32]. For a given document d, we want to find the probability of it coming from a class c. That is, we find the $P(c \mid d)$. We find this using the formula given in the equation 2.8. $P(c)$ is the prior of the class c, which is the number of documents in the class divided by the total number of documents. $P(d \mid c)$ is the likelihood, modeled either as a Gaussian or multinomial distribution – depending upon the task. This leads to two different variations of the model. $P(d)$ is the marginal probability, which can be ignored during classification – since it is the same for all the classes and does not help in providing a comparative measure. The final mathematical equation can be given as 2.9. x is the feature vector of the document d. Note that the "naïve" in the model names comes from the naïve assumption that each feature contributes independently to the result of a particular class.

$$P(c \mid d) = \frac{P(c) \cdot P(d \mid c)}{P(d)} \tag{2.8}$$

$$classify(d) = \arg\max_{c \in classes} P(c) \cdot \prod_i P(x_i \mid c) \tag{2.9}$$

2.3.4 Transformer Models

Transformer models are a breakthrough innovation in the field of Natural Language Processing [33]. Each encoder "block" or layer in the transformer consists of a self-attention layer and a feed-forward neural network layer. At a high level, self-attention is a method used to weigh the contributions of each token in a sentence. Each self-attention layer has three matrices for each query, key, and value. For the input X, using the formula 2.10, we get the weights vector, Z. Here, d_k represents the dimension of the queries and the key vector.

$$\begin{aligned} Q &= X \times W^Q \\ K &= X \times W^K \\ V &= X \times W^V \\ \text{Attention}(Q, K, V) &= softmax(\frac{Q \cdot K'}{\sqrt{d_k}}) \cdot V \end{aligned} \tag{2.10}$$

Multiple attention heads allow the net to focus on different parts of the sentence jointly. This is visualized in the figure 2.1. These encoder blocks or layers can be stacked on top of each other, allowing for denser representations. In our work, since we focus on classification, we only use the encoder part of the models (leaving out the decoder part).

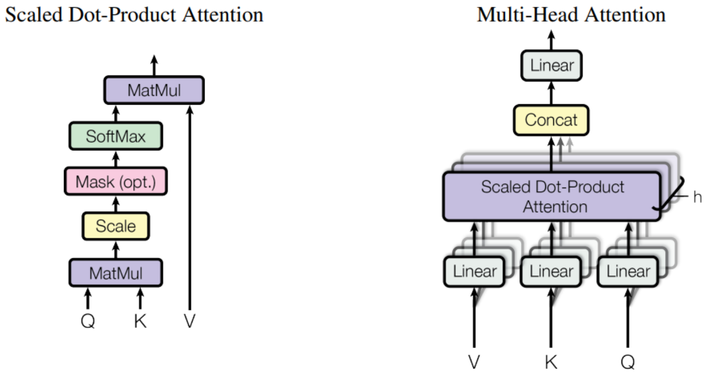


Figure 2.1: Visualization of scaled dot-product attention and multi-head attention

2.3.4.1 BERT

BERT stands for Bidirectional Encoder Representations from Transformers [34]. It is a pre-trained transformer architecture that has achieved SOTA performances on a variety of tasks. It uses a masked language modeling (MLM) objective, which has the model predict masked words in sentences – forcing it to "understand" the context better. Moreover, since it is bidirectional, it can look in both the right and the left directions to capture the bidirectional semantic dependencies.

2.3.4.2 DistilBERT

DistilBERT stands for a "distilled" version of BERT [35]. The model is lower in size due to its knowledge distillation process – the student model is DistilBERT, which learns to *distill* the knowledge from BERT, the teacher model. Overall, the model is 60% the size of BERT, 60% faster, and retains 97% accuracy. We use DistilBERT along with BERT to see if the model performance is retained for our datasets.

2.3.4.3 RoBERTa

RoBERTa stands for Robustly Optimized BERT Pretraining Approach [36]. It was found that BERT was significantly undertrained, and the performance could be made even better with more data and a longer training time. Applying the above with dynamic masking and extensive hyper-parameter tuning allows the RoBERTa model to perform much better BERT.

2.3.5 Evaluation

We use accuracy and F1-scores as two evaluation metrics all across our work to maintain consistency [37].

Accuracy is the most popular measure of model performance. It takes the ratio of the number of correct predictions to the total number of predictions made 2.11. Formally, the equation 2.12 describes the accuracy in terms of true or false positives or negatives.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \tag{2.11}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.12}$$

Since accuracy only takes into account the overall counts, in cases of an imbalanced dataset, a model can learn to predict the minority class. For example, in a binary classification task with unbalanced data, the model can regress to predicting the class with the higher number of data points.

To account for such nuances in the data, we also report the F1-score. F1-score is defined as the harmonic mean of precision P and recall R (Equation 2.14), as given in the equation 2.15.

$$P = \frac{TP}{TP + FP} \tag{2.13}$$

$$R = \frac{TP}{TP + FN} \tag{2.14}$$

$$\text{F1-score} = \frac{2 \cdot P \cdot R}{P + R} \tag{2.15}$$

Chapter 3

Blind Posts

This chapter describes a part of the work done in the paper titled "Blind Leading the Blind: A Social-Media Analysis of the Tech Industry", which has been published in the 20th International Conference on Natural Language Processing, 2023.

3.1 Introduction

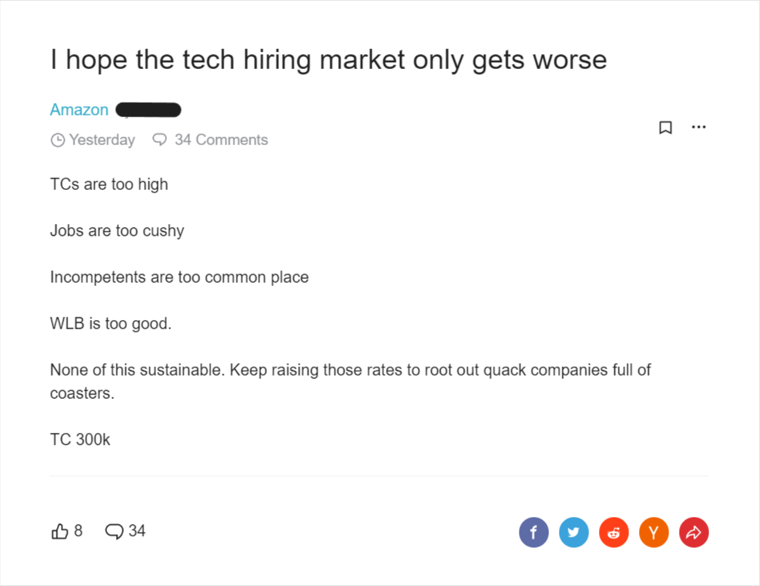
In this chapter, we explore the *Blind Posts*. We discuss their acquisition, data formatting, characteristics, and overall statistics. To gain a holistic overview of Blind, we track industry-level events like COVID-19, work-from-home, return-to-office, and layoffs. Then, we introduce our novel content classification pipeline and conduct two post-classification experiments: first at the coarse level and then at the fine-grained level. Finally, we do a manual analysis of the model results.

3.2 Blind Posts

The most treasured part of the Blind platform is the posts section. As introduced in chapter 1, Blind's anonymity fosters a safe space for honest discussions on any workplace topic. Employees openly share their thoughts and experiences on taboo topics, which would be impossible to discuss with identities attached. For example, people openly share compensation, help evaluate multiple job offers, give interview advice, ask for referrals, suggest workarounds around office politics, share candid opinions, etc. A sample Blind post has been shown in the figure 3.1.

Along with the textual contents, the post's author can assign a "board" (a "topic" or "category") to the post, allowing easy categorization. Post boards include, but are not limited to, *HR Issues*, *Layoffs*, *Software Engineering Career*, etc. One can also filter posts according to the selected board.

Each post also has other data like the like or the upvote count, the comments (nested to a maximum depth of two), and additional metadata information. A summary of all the crucial fields is given in the sub-section 3.2.2.



I hope the tech hiring market only gets worse

Amazon [redacted]

Yesterday 34 Comments

TCs are too high

Jobs are too cushy

Incompetents are too common place

WLB is too good.

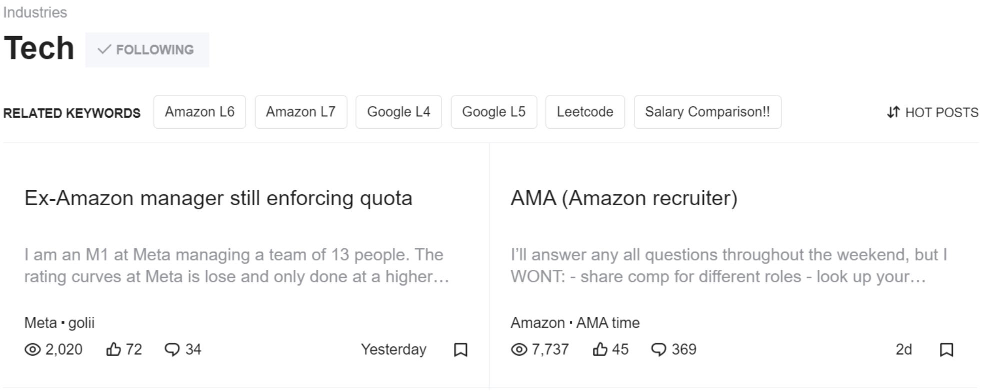
None of this sustainable. Keep raising those rates to root out quack companies full of coasters.

TC 300k

Figure 3.1: A sample Blind post

3.2.1 Blind API

To gain insights from the Blind posts, we scraped data from the platform. Blind currently does not provide an official Application Programming Interface (API), so we reverse-engineered the API used to serve the content. At the time of scraping, Blind allowed any non-logged-in user to access four posts (now Blind only allows two). The user can view any number of post tiles on the platform (each tile is the post's title and its first 100 characters of content). The count is accounted *after* a user clicks on the post tile on the webpage. Two tiles can be seen in figure 3.2.



Industries

Tech

Ex-Amazon manager still enforcing quota

I am an M1 at Meta managing a team of 13 people. The rating curves at Meta is lose and only done at a higher…

Meta • golii

2,020 72 34 Yesterday

AMA (Amazon recruiter)

I'll answer any all questions throughout the weekend, but I WONT: - share comp for different roles - look up your…

Amazon • AMA time

7,737 45 369 2d

Figure 3.2: A view of the Tech Industry page

Given that Blind does not increase the counter for viewing a tile (but only clicking on it), we realized that we could infinitely scroll and obtain all the post tiles, which have a link to their full posts. Although we are limited by what posts we can fully access (their full content), we can still enumerate all the posts – and get their hyperlinks. Then, it was a matter of scraping four posts per simulated non-logged-in user. Using these strategies, we scraped all the posts ever created on Blind since its inception. We found the first post, titled: "What's this?" with the content text present appropriately as "Hello world!", created on 20th October 2015. We scraped all posts till 21st December 2022, totaling **767,224 posts**, containing seven years of industry data.

We call this dataset as *Blind Posts*.

3.2.2 Data Description

Each post is a JavaScript Object Notation (JSON) object consisting of: *created_at*, *member_nickname* (anonymized), *board_name* (post category name), *title*, *content*, *like_cnt*, *view_cnt*, *comment_cnt*, *member_company_name*, and more. The important fields are explained in the table 3.1. Each post also had its author's company information as another JSON object, which was redundant as the same company information was present for all employees of a company.

We have a total of 74 boards (categories) like: *Tech Industry*, *Hobbies & Entertainment*, *Layoffs*, etc. *Tech Industry* and *Software Engineering Career* are the only two boards with over 100k posts. There are ten boards with over 10k posts, as shown in the table 3.2. The line plot (figure 3.3) shows the boards' distribution.

Field Name	Meaning
alias	A unique identifier for the post
article_type	The type of article: post, poll
member_nickname	Anonymized author name
member_company_name	Author's company name
created_at	Date of post creation
board_name	The post category
title	The title of the post
content	The content of the post
content_length	The length of the content
like_cnt	Like count
comment_cnt	Comments count
view_cnt	View count
company_page	Member's company information JSON

Table 3.1: Important fields in a Blind post JSON object

Board Name	Frequency
Tech Industry	389,947
Software Engineering Career	107,152
Investments & Money	38,058
Housing	22,110
Work Visa	21,831
Compensation	19,871
Product Management Career	16,094
Data Science & Analytics Career	12,058
Referrals	11,261
Finance Industry	10,053

Table 3.2: Ten most frequent boards in Blind posts data

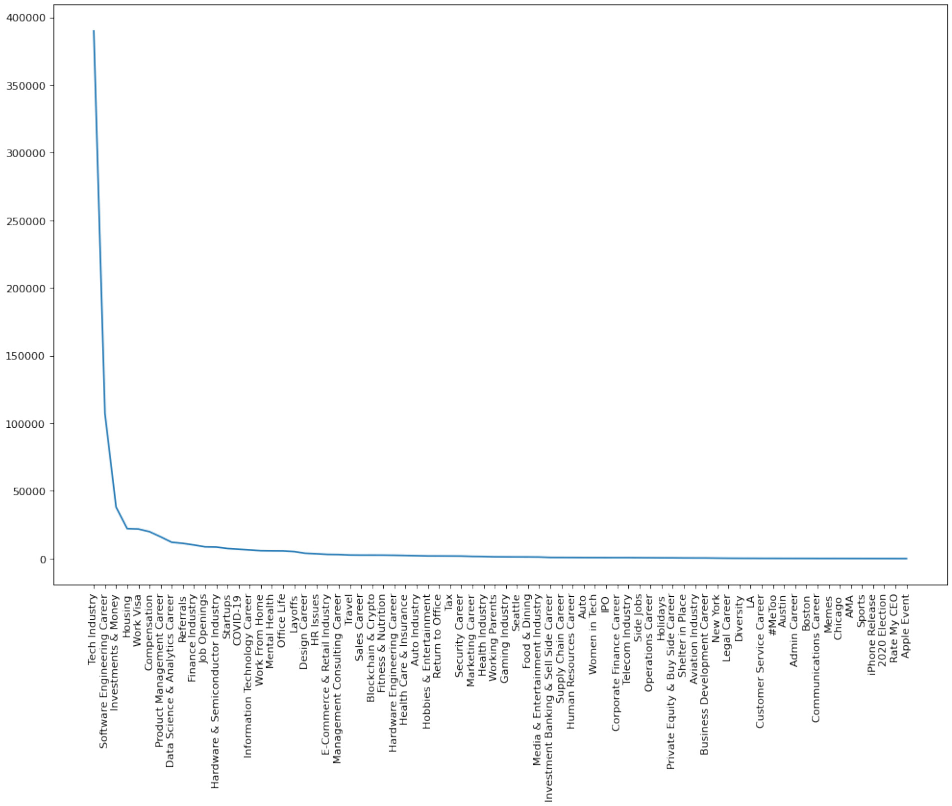


Figure 3.3: Line plot of Blind's board name frequencies

3.3 Preliminary Analyses

To get an overview of the content of discussions on Blind, we conduct preliminary analyses.

3.3.1 Content Analysis

Most Blind Posts are below 500 characters and between 80 to 120 words (Figure 3.4). To find the content of discussions, we first plot the top occurring uni-grams in a word cloud (Figure 3.5). Total compensation, abbreviated as TC, is the most frequently discussed topic mentioned in the posts. This is a symptom of the unique characteristic of the Blind platform – it is an unsaid "rule" to mention one's TC. We also observe that job offers and interviews are often mentioned, demonstrating the popularity of discussions about choosing between offers and asking for interview advice. We see a high frequency of mention of big-tech companies, including Amazon, Google, Microsoft, and Facebook (now Meta).

Two things become immediately clear from the above: first, employees are using Blind as a safe space to have otherwise stifled speech. More interestingly, in the case TC is not mentioned in the post, people usually end up replying with "TC or GTFO", standing for "*[share the] total compensation, or* *get the fuck out*", showing that Blinders (people on Blind) not only openly talk about taboo topics like compensation, but actively encourage it.

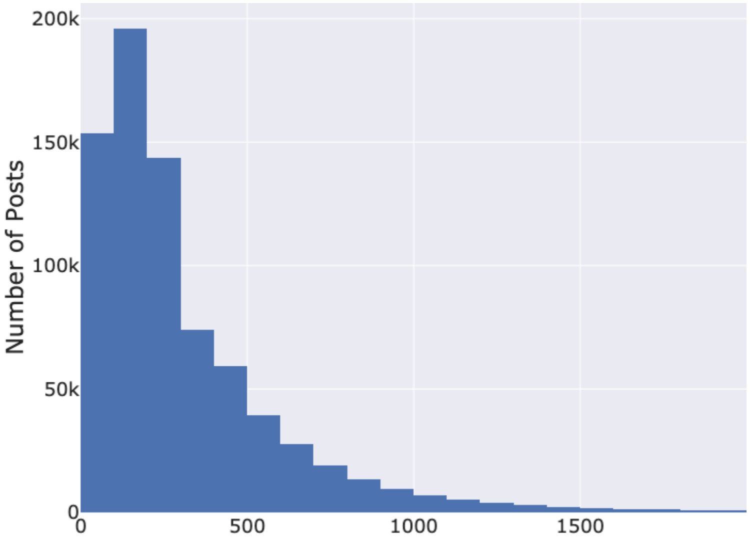


Figure 3.4: Histogram of Blind posts content length (in characters)

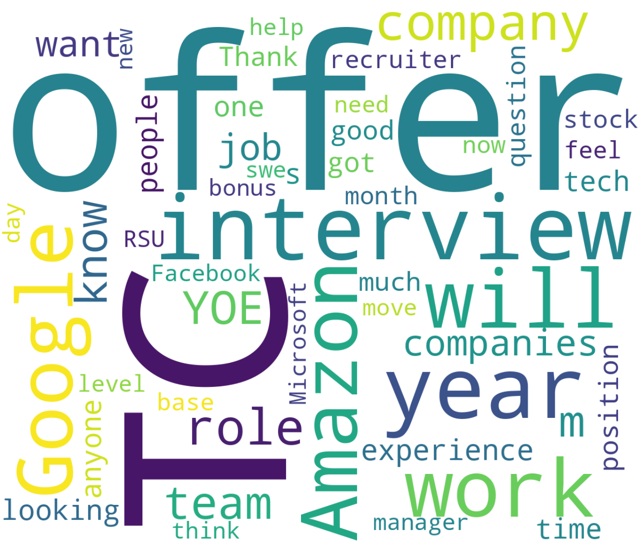


Figure 3.5: Wordcloud of Blind posts content unigrams

3.3.2 Hashtag Analysis

Hashtags on social media allow users to build communities around topics and promote opinions [38, 39]. We extract the most frequent hashtags from posts and plot them in a word cloud (Figure 3.6). We observe hashtags like "tech", "engineering", "software" and "swe" (stands for Software Engineering), associated with the tech industry and software careers in general. Further, all of the top 50 hashtags are associated with the tech industry, indicating that most content on Blind relates to tech and tech careers. Hashtags including “amazon”, “google”, “microsoft”, “facebook”, “apple”, “meta”, and “faang” are frequent, once again indicating that discussions about FAANG companies are popular. We see the prevalence of hashtags such as “interview” and “referral”, pointing to the popularity of asking for referrals and interview advice on the platform. Finally, we see “workvisa” and “h1b” also come up due to the inflow of talent in the United States of America (USA) – especially with the influx of Indians on the platform and in the country.[1]

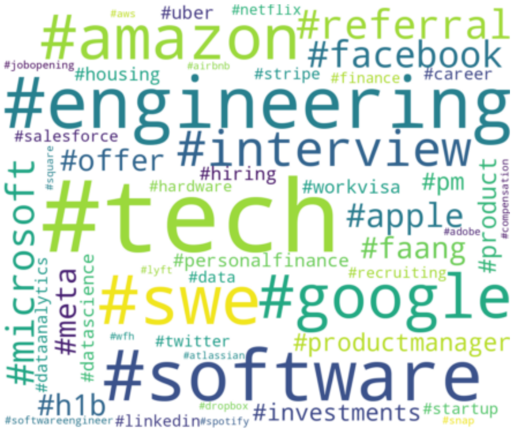


Figure 3.6: Wordcloud of Blind posts hashtags

Preliminary analyses of the *Blind Posts* reveal that Blind is biased towards big tech, which is in line with the highest frequencies of the *Tech Industry* and *Software Engineering Careers* boards (Table 3.2). We now dive into global events through the lens of tech.

3.4 Platform Analysis

In this section, we track how significant events in the world have shown reflections of the tech industry on the Blind platform through the content in the posts. Specifically, we track the events: the rise of COVID-19, the emergence of work-from-home, the mass layoffs, and the return-to-office phenomenon.

Before conducting analyses, an important aspect to consider is that the platform grew from its inception in 2015. To get a year-on-year (YoY) analysis of how events like the pandemic or the recession affected the tech industry, we must first normalize the Blind post activity. Using the *created_at* field, we determine how many posts were created year-by-year. Figure 3.7 shows a steady increase in the platform’s popularity as more and more people started creating discussions on Blind.

[1] `https://economictimes.indiatimes.com/nri/migrate/indians-are-leaving-the-country-in-droves-heres-where-they-are-headed-and-why/articleshow/96847173.cms`

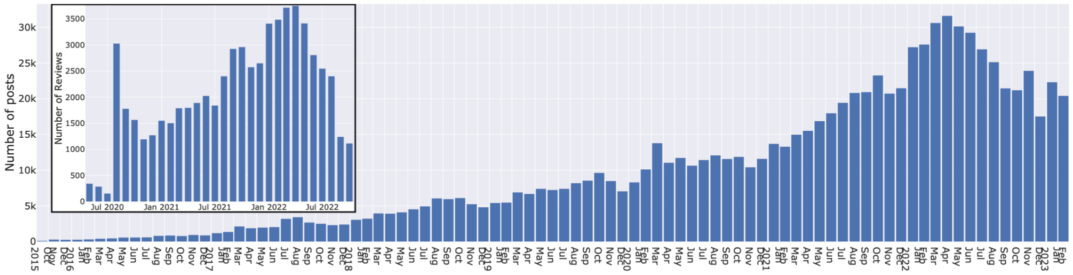


Figure 3.7: Year-Over-Year activity on Blind using *Blind Posts* (main graph) and *Blind Company Reviews* (inner graph)

3.4.1 Annotation

To track work-from-home phenomena, we identify all the posts containing the phrase "work from home" or the abbreviation "wfh". Similarly, for return-to-office, we find the phrase "return to office" or the common abbreviation "rto". We use the keyword "layoff" to get an idea of layoffs. Our manual analysis of 200 posts per category found that these phrases and keywords covered most texts.

To get a holistic view of the industry, we aim to quantify the sentiments and opinions of the tech industry employees. We use VADER to extract the overall sentiment scores [40]. VADER stands for Valence Aware Dictionary and sEntiment Reasoner. It uses manually created lexicon scores for positive and negative phrases, with scores from -4 (extremely negative) to +4 (extremely positive) and 0 being neutral. The model uses syntactic and semantic rules; for example, "not good" is labeled negative instead of "good" contributing to a positive score. We extract the *compound* field (a combination of positive, negative, and neutral scores) from the function outputted labels. The normalized score ranges from -1 (extremely negative) to +1 (extremely positive), with 0 indicating a neutral sentiment.

3.4.2 Analyses

3.4.2.1 Work From Home

We observe negligible mentions of WFH till 2019 (Figure 3.8a). They peaked in 2020 when the pandemic spread and are currently in a decline as firms call back employees [41].

3.4.2.2 Return To Office

Compared to WFH, an opposite trend is observed for return-to-office (RTO) (Figure 3.8b). Pre-2020, since there are no mentions of WFH, RTO is an alien concept. During 2020, most countries were in lockdown as COVID-19 spread uncontrollably.[2] Given this, we see no activity in RTO till 2020 end. In 2021, as companies saw growth adapting to the changing landscape, they started some form of RTO or

[2] `https://www.bbc.com/news/world-52103747`

the other, mostly coming in the form of a hybrid setup – we thus see a rise in mentions since 2021. In 2022, we see RTO mentions rise even more as companies call back employees, some even making office not optional (for example, X) [42].[3]

To put it definitively, only considering the years 2019 to 2022, we see a significant ($p < 0.05$) and strong negative correlation (r=-0.9976) between WFH and RTO.

3.4.2.3 Layoffs

For layoffs, we see mentions spike twice – first in 2020 and second in 2022 (Figure 3.8c). The first can be attributed to the initial global and national economic shocks due to the spread of COVID-19 [43]. This was followed by a short year of massive growth as companies adapted. The second is due to the bubble burst in mid-2022, with companies reporting slower gains due to delayed supply chain disruption effects and repeatedly announcing layoffs.[4] The situation worsened towards the end of 2022 as tech giants like Meta announced their first of many rounds of layoffs [44].[5]

A curious small peak in 2016 can also be observed, which also maps to tech giants' layoffs of 2016.[6]

3.4.2.4 Sentiments

We see the sentiments reflect the broader state of the tech industry in the real world (Figure 3.8d). 2020 and 2022 are the years of layoffs, as reflected by the sentiment dips. Even though 2020 was the biggest downfall globally, we hypothesize that positive sentiments due to WFH cushioning the fall a little. We see the sentiments at an all-time high in 2021 due to adopting and embracing the new tech. The increased profits and sentiments were consistent across all big corporations.[7] Even the rise of sentiments in 2018 and 2019 can be correlated with the high hiring pace of Meta and Alphabet at the time.[8]

[3] https://fortune.com/2023/03/24/return-to-office-elon-musk-twitter-tesla-layoffs/

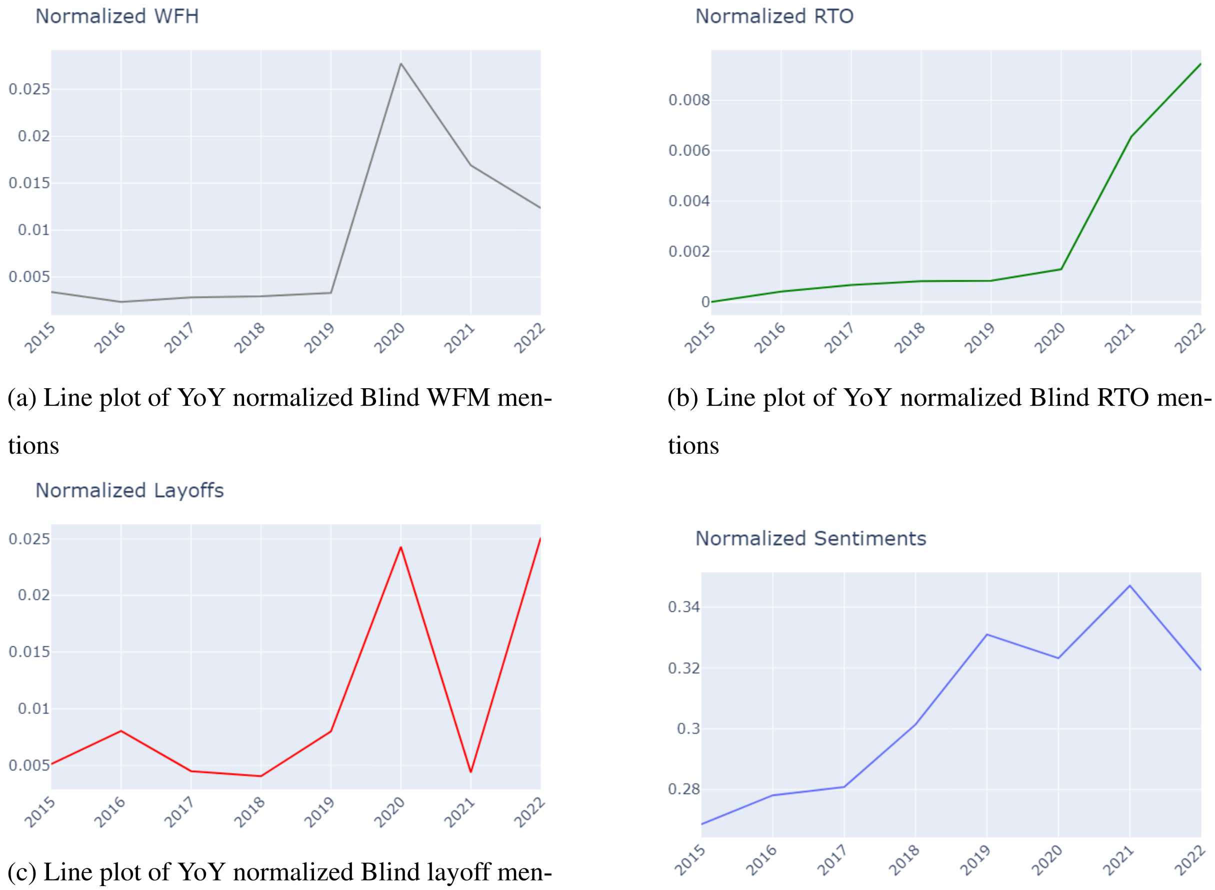
[4] https://www.computerworld.com/article/3679733/tech-layoffs-in-2022-atimeline.html

[5] https://about.fb.com/news/2022/11/mark-zuckerberg-layoff-message-toemployees/

[6] https://www.cio.com/article/218133/9-bloodiest-tech-giants-layoffs-of-2016.html

[7] https://www.bloomberg.com/news/articles/2022-03-30/2021-was-best-year-for-us-corporation-profits-since-1950

[8] https://www.geekwire.com/2018/facebook-hiring-record-pace-adds-10k-peopleheadcount-one-year/, https://www.forbes.com/sites/jackkelly/2023/01/25/techlayoffs-look-terrible-but-theyre-only-a-pullback-from-years-of-aggressive-hiring/



(a) Line plot of YoY normalized Blind WFM mentions

(b) Line plot of YoY normalized Blind RTO mentions

(c) Line plot of YoY normalized Blind layoff mentions

(d) Line plot of YoY normalized Blind sentiments

Figure 3.8: Lineplots of Blind activity, sentiments, layoffs and work from home

3.5 Classification Experiments

In this section, we exploit Blind's obsession with the tech industry and use it as leverage to extend our work beyond.

Our current analysis is entirely restricted to the confines of the Blind platform. However, discussions about the tech industry are spread across other social networks, such as X, Reddit, Quora, and more. The issue is a plethora of other discussions on the platforms not related to our interests. As an employee or even an employer, it would be highly productive to consolidate discussions and opinions from various platforms. This will allow us to get not just Blind-level insights but a social-media internet-level analysis.

In this effect, we construct a novel content classification pipeline (Figure 3.9). The end goal is understanding what it means for a piece of text to be labeled "tech" or "non-tech". This will let us filter out only tech-related content from an ocean of content. Going even further, we identify what are the kind of "tech" texts – by classifying them into the ten most popular categories.

Formally, in the next sections, we aim to create two classifier models. The first model aims to distinguish tech-related content from non-tech-related content – the coarse classifier. Once tech-related posts are identified, we create another classifier to specifically identify what kind of tech post it is, from a list of popular Blind boards (equivalently called categories) – the fine-grained classifier.

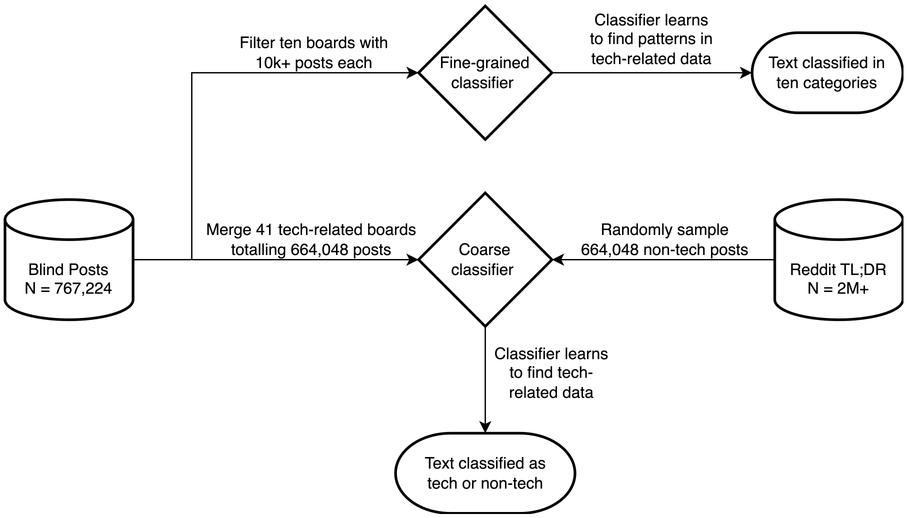

Filter ten boards with 10k+ posts each
Fine-grained classifier
Classifier learns to find patterns in tech-related data
Text classified in ten categories
Blind Posts N = 767,224
Merge 41 tech-related boards totalling 664,048 posts
Coarse classifier
Randomly sample 664,048 non-tech posts
Reddit TL;DR N = 2M+
Classifier learns to find tech-related data
Text classified as tech or non-tech

Figure 3.9: The content classification pipeline

3.5.1 Task 1: Coarse Classification

For the first task of coarse classification, we make use of the following datasets.

3.5.1.1 Blind Data

For the tech data, we use *Blind Posts*. While most posts on the platform belong to the tech domain, we filter out non-related boards such as “Hobbies & Entertainment”, “Holidays”, “Sports”, etc. We consider only the following categories: “Work Visa”, “Layoffs”, “Referrals”, “Job Openings”, “Work From Home”, “Return to Office”, “Compensation”, “Side Jobs”, “Startups”, “IPO” and every category with “Industry” or “Career” in its name. In total, we have 41 boards with a total of **664,048** posts out of the 767,224 in our dataset.

3.5.1.2 Reddit Data

The remaining number of posts is insufficient to represent the non-tech data. This is where we turn to Reddit once again, as introduced in chapter 1. Reddit is the choice for sourcing non-tech content for the following reasons.

- Reddit is the hub of users from all around the world, having different writing styles and vocabularies. This provides a realistic contrast to the population on Blind, which is narrowed to a small percentage of the population, having distinctive ways of writing and a unique vocabulary.
- As with the people on Reddit, the content of the platform is equally opposite to that on Blind. Reddit has content focused on humor, gaming, animals, music, arts, etc. While the content on Blind is exceedingly tech. Therefore, we label the data from Reddit collectively as non-tech.
- Perhaps most importantly, only one dataset – the Reddit TL;DR dataset [7] – is of comparable size as our Blind dataset.

We pick the 100 largest subreddits from the Reddit TL;DR dataset and then manually check all the subreddits to ensure no tech-related content is present. We remove 5 tech subreddits: "sysadmin", "Android", "techsupport", "talesfromtechsupport", and "technology" to ensure the cleanliness of the non-tech data. This totals 2,328,754 posts. We then randomly sample an equal number of posts (664k) to balance the data with tech-related posts obtained from Blind. We thus have a balanced dataset of **1,328,096** data points for the coarse classification task.

3.5.2 Task 2: Fine-Grained Classification

For the second fine-grained classification task, we use random sampling to create a balanced dataset of 10k points for each of the top ten post categories from the *Blind Posts* dataset.[9]

3.6 Methodology

For both of the tasks, we pre-process the text in the same way, as mentioned in the chapter 2. We use Logistic Regression, Linear Support Vector Classifier (shortened to SVC), and Multinomial Naive Bayes (shortened to Multinomial NB) algorithms with default parameters. For the first task of coarse classification, we use only the above machine learning models. To improve the accuracies of the second task, we leverage the transformer architectures beyond the machine learning models, experimenting with DistilBERT, BERT, and RoBERTa.

We conduct a 5-fold cross validation for the ML models, and split the data into train:test:validation as 80:10:10 for the transformers. We use the HuggingFace implementations for each transformer, labeled distil-bert-uncased, bert-base-uncased and roberta-base respectively. The best results for DistilBERT, BERT, and RoBERTa are found while varying the learning rate between 2e-5 and 5e-5, trained for two epochs and a batch size of 16.

[9] We conducted experiments without random sampling, and used all posts from the top ten boards each – however, due to the disparity in frequencies between the boards, the models degenerated to labeling all text as the highest frequency board.

3.7 Results

The table 3.3 showcases both the F1-score and the accuracies in each of the cases. For coarse classification, we see a stellar accuracy of **99.25%** by the Linear SVC model, followed by Multinomial Naive Bayes, and then finally by the Logistic Regression model. We do not experiment with transformer models for this case since the accuracies are satisfactory for a production use case.

For fine-grained classification, we see the accuracies are much lower, with Logistic Regression coming on the top with **72.32%** accuracy. However, when using transformer models for the same fine-grained task, we see a jump in accuracies 3.4. We see DistilBERT, BERT, and RoBERTa perform closely with each other, with minor differences.

Model	**Coarse Classification**	**Fine-grained Classification**
Logistic Regression	0.9796, 0.9796	**0.7232, 0.7224**
Linear SVC	**0.9925, 0.9925**	0.7229, 0.7161
Multinomial NB	0.9800, 0.9800	0.6962, 0.6852

Table 3.3: Accuracy and F1 scores for all the machine learning models on both Blind post classification tasks

Model	**Fine-grained Classification**
DistilBERT	0.7767, **0.7819**
BERT	0.7823, 0.7772
RoBERTa	**0.7841**, 0.7778

Table 3.4: Accuracy and F1 scores for all the transformer models on the fine-grained classification task

3.8 Discussion

Let us consider the coarse classification problem first. The accuracy of all the models is significant considering they are working with a balanced dataset. This means that the models have learned useful features in the text. To analyze the text further, we take multiple examples and note how the best model (linear SVC) performs. First, we give the model the following texts as inputs.

"After spending almost 5 years at Facebook" → predicted: tech
"After spending almost 5 years at CSGO" → predicted: nontech

As expected, the model outputs the first text to be tech (tech-related). Changing "Facebook" to "CSGO" – a popular online game, results in the model predicting it to be non-tech. This implies that the model has learned to distinguish named entities (NEs). NEs refer to specific words or phrases that represent things like names of people, and organizations, along with dates, times, etc. Next, we give the input to the model as the following.

"I was removed from the company" → predicted: tech
"I was removed from the community" → predicted: nontech

The model again correctly outputs the first one as tech. Changing "company" to "community" flips the verdict to non-tech, which is again as expected. This example shows us that the model has not only learned NEs but also certain keywords. We also perform manual analysis of incorrectly labeled posts.

"My parents are getting me married" → predicted: tech

The above, for example, is a text that could be present in a relationship-based subreddit. The model, however, tags this as "tech". This can be explained by the inflow of Indians on the Blind platform, resulting in a significant number of discussions about marriage under the "Tech Industry" category on Blind.

For fine-grained models, we see the accuracy is much lower, even with the transformer models. Given that the classification problem is much harder, with ten classes of similar nature and limited data (10k per class), the models still perform relatively well. The transformer models improve upon machine learning models. We notice how all the accuracies of the transformer models are in a similar range, which can be attributed to a limited per-class data size.

3.9 Conclusion

In this chapter, we saw the work with *Blind Posts*. We first describe the procedure to procure the posts by finding a public Blind API. Then, we use VADER to annotate sentiments and find interesting reflections on global trends on the platform – whether it be COVID-19, work-from-home, return-to-office, or, layoffs. Finally, we conduct classification experiments at two levels: coarse and fine-grained for our novel content classification pipeline. For the former, we use a filtered version of Blind data for the tech category and Reddit's TL;DR dataset as the non-tech data. The linear SVC model achieves 99.25% accuracy on the balanced dataset. For the latter case, the RoBERTa achieves the highest accuracy, 78.41%.

Chapter 4

Blind Company Reviews

This chapter describes a part of the work done in the paper titled "Blind Leading the Blind: A Social-Media Analysis of the Tech Industry", which has been published in the 20th International Conference on Natural Language Processing, 2023.

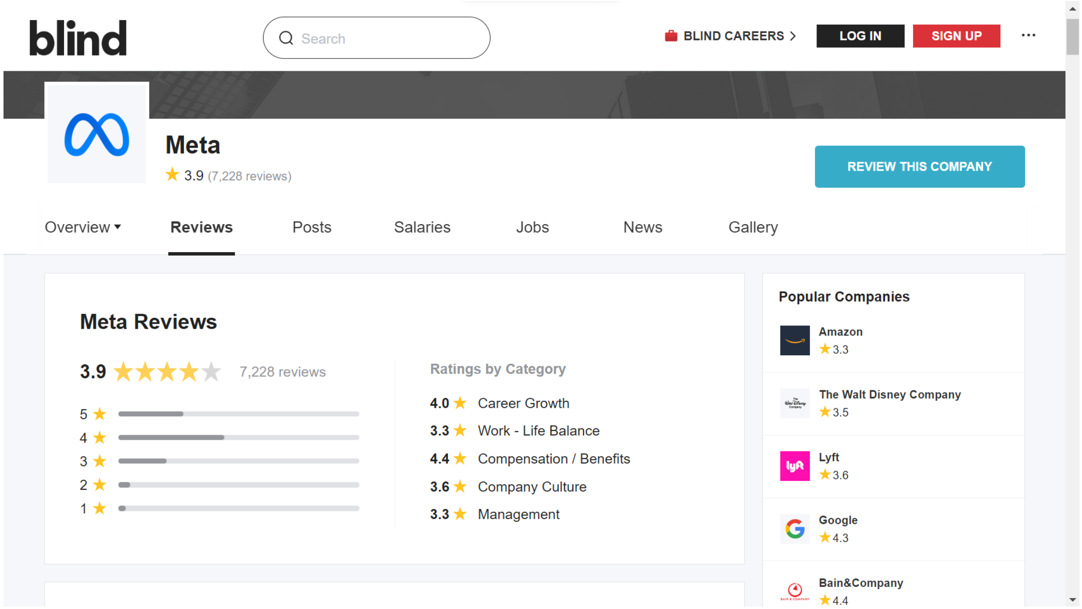
4.1 Introduction

In this chapter, we explore the *Blind Company Reviews*. We explain how they are acquired, the data fields, characteristics, and, overall statistics. Then, we explore the numerous metrics employees use to judge a company, like work-life balance, compensation, management, etc. We complete our content classification pipeline by conducting review classification experiments using different vectorization methods with machine learning models and then moving on to transformer-based architectures. Finally, we do a manual analysis of the results.

4.2 Blind Company Reviews

In this section of the Blind platform, users leave reviews about their current or past employer. Each company has its own *Reviews* section as seen in figure 4.1 for Meta. There is one "overall" rating of the company followed by ratings of "Career Growth", "Work-Life Balance", "Compensation / Benefits", "Company Culture", and "Management". These factors are considered the most critical of any workplace [45].[1] Each rating is the average rating of all user reviews, between 1 (lowest) to 5 (highest). Each user review consists of the pros/cons text fields as well as individual ratings for each of the aforesaid metrics (Figure 4.2a and 4.2b respectively).

[1] `https://www.indeed.com/career-advice/finding-a-job/what-makes-a-company-a-great-place-to-work`



Meta

★ 3.9 (7,228 reviews)

Meta Reviews

3.9 ★★★★★ 7,228 reviews

5 ★
4 ★
3 ★
2 ★
1 ★

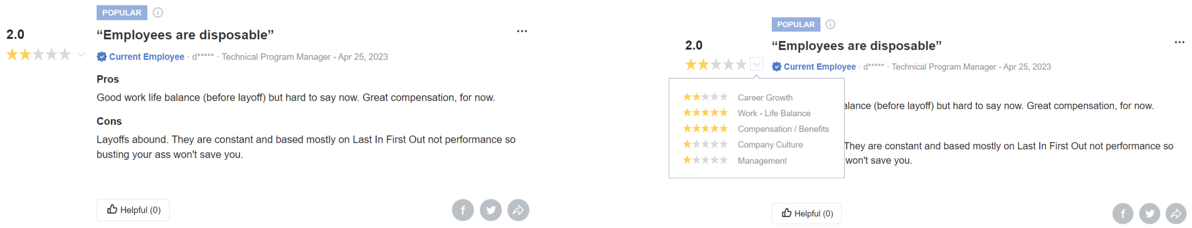
Ratings by Category

4.0 ★	Career Growth
3.3 ★	Work - Life Balance
4.4 ★	Compensation / Benefits
3.6 ★	Company Culture
3.3 ★	Management

Popular Companies

Amazon	★3.3
The Walt Disney Company	★3.5
Lyft	★3.6
Google	★4.3
Bain&Company	★4.4

Figure 4.1: The Meta review page



2.0 ★★☆☆☆

“Employees are disposable”

Current Employee · d***** · Technical Program Manager - Apr 25, 2023

Pros

Good work life balance (before layoff) but hard to say now. Great compensation, for now.

Cons

Layoffs abound. They are constant and based mostly on Last In First Out not performance so busting your ass won't save you.

(a) Review with pros/cons in text

★★☆☆☆	Career Growth
★★★★★	Work - Life Balance
★★★★★	Compensation / Benefits
★☆☆☆☆	Company Culture
★☆☆☆☆	Management

(b) Review with metric rating details

Figure 4.2: A sample Meta review, with pros/cons and metric ratings

4.2.1 Blind Scraping

Blind lets a non-logged-in user view only one company review for any company. To view all the remaining reviews from all companies, a Blind user is required to contribute at least one review for their company. Using one logged-in Blind account, we use Selenium to scrape all the other company reviews.[2] Selenium is a tool that allows dynamic scraping – allowing us to work with the paginated review page with ease. We collected **63,477 reviews** for the **55** most reviewed companies on Blind. We call this dataset the *Blind Company Review* dataset.

[2] `https://www.selenium.dev/`

4.2.2 Data Description

Each review contains numerous fields that we use to build a company's identity. All the important fields are mentioned in the table 4.1. Apart from the fields in the table, there are fields like "href" containing the link of the review, "date" containing the post date, "helpful_count" containing the number of "likes" on the review, "desc" containing an overall description of the review. In practice, we found the "helpful_count" had numbers insignificant for any analysis. The "resign" field, having resignation reason(s), was empty for most of the reviews.

Field Name	Meaning
company	The company of the review
pros	The positive text about the company
cons	The negative text about the company
resign	Author's resignation reason (blank for most)
rating	Overall company rating from 1 to 5
career_growth	Rating for how well one gets promotions
wlb	Rating for the work-life balance
comp-benefits	Rating for compensation and benefits
culture	Rating for company culture
management	Rating for management
author_name	The author's anonymized name
author_title	The author's job title

Table 4.1: Important fields in a Blind company review JSON object

4.3 Reviews Analysis

The number of reviews for each company is mentioned in figure 4.3. We see that well-known companies like Meta, Alphabet, Microsoft, Amazon, and Apple, collectively called MAMAA top the charts.[3]

[3] Note that in the Blind posts, it is also common to refer to the big tech companies as "FAANG". It stands for Facebook, Amazon, Apple, Netflix, and Google. Since the start of Blind in 2015, Facebook's name has been changed to Meta, Google's parent company became Alphabet, and opinions about Netflix plummeted, while opinions about Microsoft rose. To consider all the changes, the acronym has now become MAMAA.

We also look at the job titles (field name as “author_title”) of the employees who wrote the reviews. In figure 4.4, we see that most of the jobs are entry-level positions, like “Software Engineer I”, “Software Development Engineer (SDE)”, “Software Engineer”, etc.[4] Only 2 positions out of 20, “Product Manager”, and “Designer” were not software engineering related.

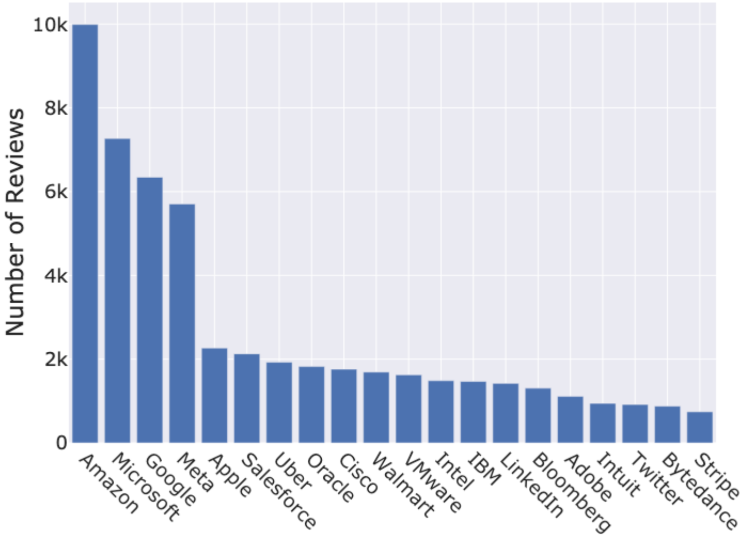


Figure 4.3: Histogram of number of reviews for each company

[4] Note that “Software Engineer” and “Software Engineer I” may or may not be the same. It might refer to any broad number of levels of “Software Engineer”, so we did not merge them. Similarly, “Software Engineer” and “Software Development Engineer” may or may not be the same, depending upon the company.

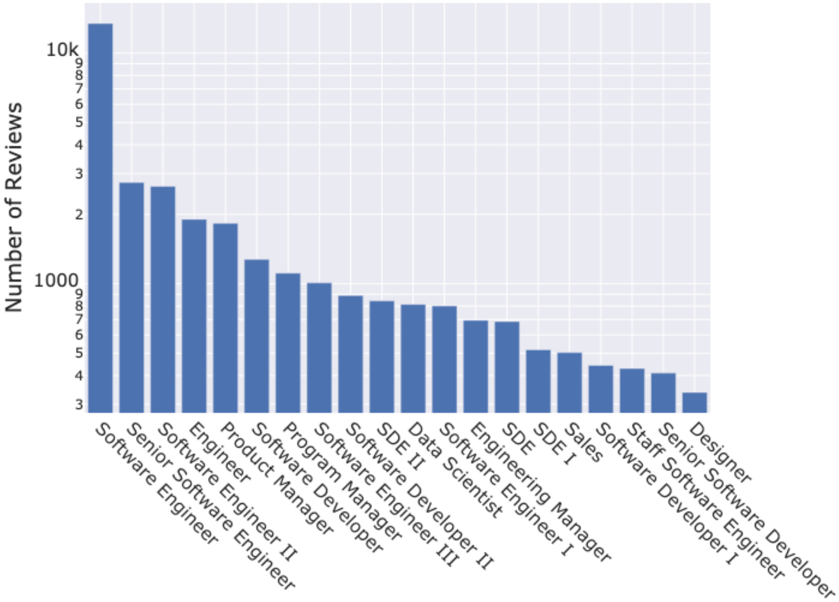


Figure 4.4: Histogram of job titles in company reviews

4.3.1 Content Analysis

Focusing on the contents of the pros and cons fields, we found that on average, users are more verbose in listing the cons of their company compared to the pros (Figure 4.5). To understand their statistical significance, we conducted a t-test and found that for 39 companies, this observation is statistically significant (p-value < 0.05). At the top, Amazon has 25.44% more length of cons than the pros. Amazon is closely followed by Meta (22.59%), Alphabet (Google) (18.46%), and Microsoft (16.10%). Only 5 companies out of 55 show an inversion of verbosity (with p-value > 0.05 however). Optiver has the most pros length than cons (27.57%), followed by Jane Street (15.20%), Discord, Hudson-River-Trading, and Snowflake. These results indicate the negativity bias phenomenon, which parallels findings in cognitive and social psychology [46].

To better understand the content of the pros and cons, we analyze the unigram and bigram frequencies. Across both fields, the bigrams “work life” and “life balance” emerge as the most frequent, highlighting that work-life balance (WLB) is an important factor when a user reviews a company. When analyzing pros, we see that apart from work-life balance, bigrams such as “career growth”, “good compensation”, “great work”, “great culture”, “smart people”, and “great benefits” are frequent. Bigram analysis of cons shows similar topics mentioned frequently, including “career growth”, “slow career”, “bad work”, and “low compensation”. When analyzing cons related to poor management, adjectives such as “middle”, “bad”, and “upper” are used. Others show hindrances of decision-making in large companies – “red tape”, “decision making”, “big company”, and old and unmaintained code – “tech stack”.

These findings further show that employees are openly talking about topics otherwise left as workplace taboos.

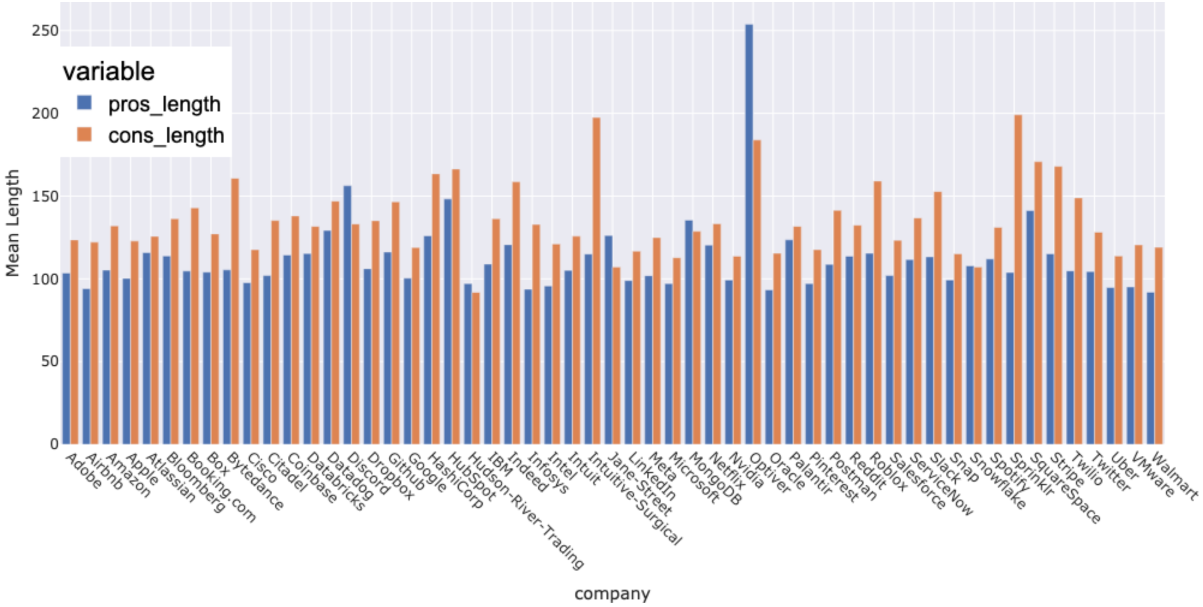
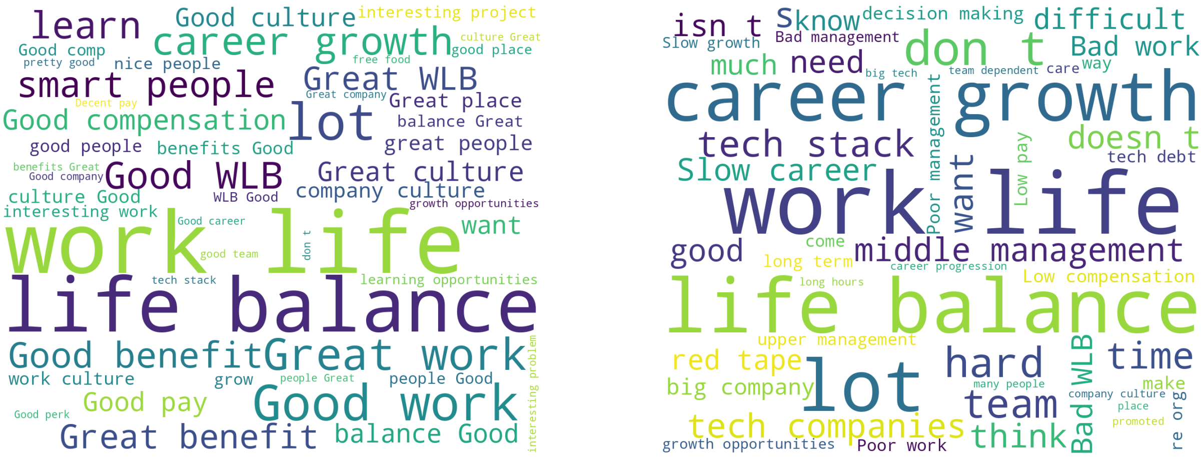


Figure 4.5: Review length for each company



(a) Pros

(b) Cons

Figure 4.6: Wordcloud of Blind company reviews unigrams and bigrams

4.3.2 Metrical Analysis

We find that a majority of reviews rate the company 4 or 5 stars overall, with minimal 1-star and 2-star reviews. This is surprising, given that employees are more verbose in listing the cons of their company.

Considering all ratings (including overall rating), we find significant correlations (p-value < 0.001) using Spearman's r for all the values. Surprisingly, even though WLB is mentioned the most frequently in both pros and cons in reviews, we find that the overall rating is the least correlated with WLB (r=0.49) and most correlated with culture (*r*=0.71) and management (*r*=0.7). This might mean poor management and toxic cultures are the biggest consistent reasons for a lower rating. Since WLB has the lowest *r*, we expect there to be cases with low WLB but high overall ratings.

We explore the causes for this anomaly by finding the top three companies by mean rating for each metric (Table 4.2). The mean and variance are mentioned for each company, and the median is 5 for all the values in the table. High-frequency trading (HFT) companies Hudson-River-Trading (HRT), Jane Street (JS), and Optiver are consistently at the top. What's more interesting is that these firms are known to have long and stressful hours – great compensation but low WLB, which explains the low *r* value for WLB. Only the WLB column shows a different composition of companies, with Indeed, SquareSpace, and Atlassian being the best. Even in the top ten, HFTs are nowhere to be found.

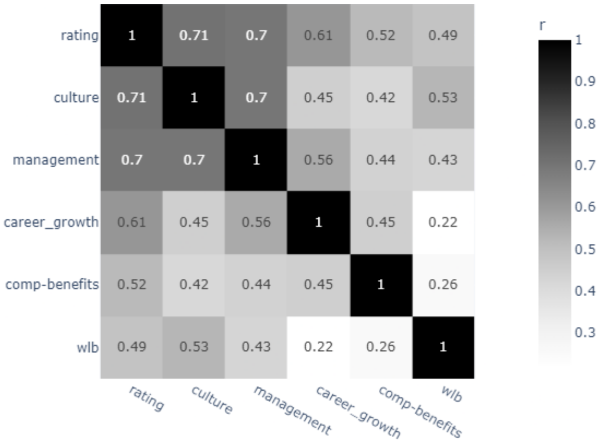


Figure 4.7: Correlation matrix of ratings

Rating	Culture	Management	Growth	WLB	Comp
HRT (4.76,0.23)	HRT (4.79,0.26)	HRT (4.47,0.49)	Optiver (4.64,0.53)	Indeed (4.58,0.67)	Optiver (4.90,0.08)
JS (4.73,0.40)	Discord (4.61,0.72)	JS (4.45,0.79)	JS (4.53,0.46)	SquareSpace (4.57,0.54)	HRT (4.88,0.15)
Optiver (4.63,0.53)	JS (4.61,0.61)	Optiver (4.14,1.36)	HRT (4.30,0.79)	Atlassian (4.56,0.66)	JS (4.84,0.18)

Table 4.2: Top three companies for the rating metrics

4.4 Classification Experiments

As explained in the previous chapter on Blind posts (Chapter 3), we tackle the problem of filtering out relevant tech-related posts from a plethora of unstructured texts on social media. We take our work one step further and aim to mine opinions with the full content classification pipeline (Figure 4.8). Understanding employee opinions allows us to discover trends at an aggregated level, without having to read every single text. Specifically, we exploit the "pros" and the "cons" fields in the company reviews, to train a classifier model that can accurately predict whether a piece of text is a “pro” or a “con” of a company.

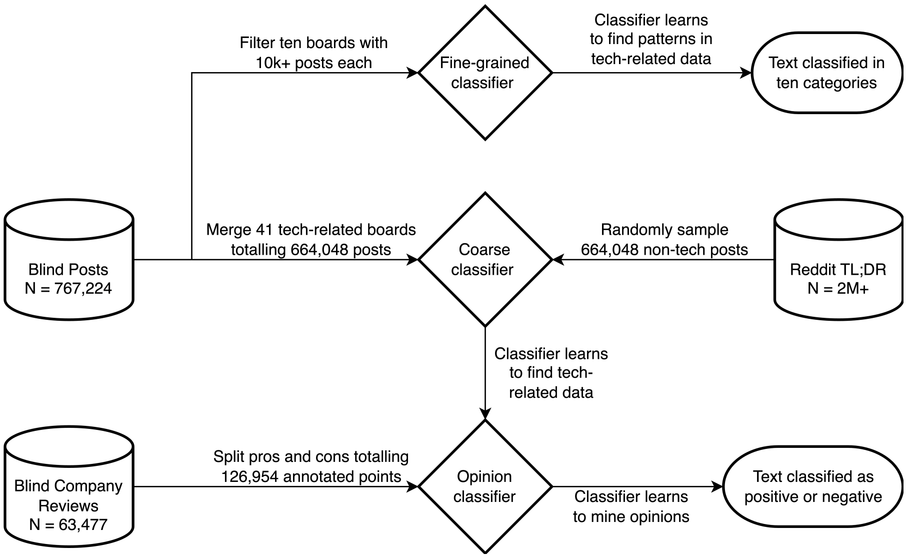

Filter ten boards with
10k+ posts each
Fine-grained classifier
Classifier learns to find patterns in tech-related data
Text classified in ten categories
Blind Posts
N = 767,224
Merge 41 tech-related boards totalling 664,048 posts
Coarse classifier
Randomly sample 664,048 non-tech posts
Reddit TL;DR
N = 2M+
Classifier learns to find tech-related data
Blind Company Reviews
N = 63,477
Split pros and cons totalling 126,954 annotated points
Opinion classifier
Classifier learns to mine opinions
Text classified as positive or negative

Figure 4.8: The full content classification pipeline

4.5 Methodology

We use the preprocessing steps outlined in chapter 2. We use three vectorizers, Tf-Idf, SBERT (miniLM), and SBERT (mpnet), followed by all the combinations of the machine learning models, Logistic Regression, Linear SVC, and Multinomial Naive Bayes. To improve our results even further, we experiment with the transformer models: DistilBERT, BERT, and RoBERTa.

We use 5-fold cross-validation for the ML models and 80:10:10 train:test:validation split for training and evaluation of the transformer models. All the machine learning models are trained with default parameters. We use the HuggingFace implementations for each transformer, labeled distil-bert-uncased, bert-base-uncased and roberta-base respectively. The best results for DistilBERT, BERT, and RoBERTa are found while varying the learning rate between 2e-5 and 5e-5, trained for two epochs and a batch size of 16.

4.6 Results

Table 4.3 summarizes the results of the machine learning models. We see that when using Tf-Idf vectorization, the Logistic Regression model performs the best, with an accuracy of 94.38% and an f1-score of 94.34%. SBERT (MiniLM) shows slightly lesser accuracy with 93.45%, but SBERT (mpnet) shows some improvements with 95.53% accuracy. Note that Multinomial Naive Bayes only allows non-negative inputs and the SBERT embeddings (for both the models) have negative values too – the results are thus dropped. We get the best results from the transformer models, with RoBERTa topping the list with a 98.28% accuracy, as given in table 4.4. BERT and DistilBERT follow closely, with 97.83% and 97.75% accuracies.

4.7 Discussion

We see that even with a purely lexical-level vectorizer, tf-idf, and a machine learning model, we achieve over 94% accuracy. This can be attributed to the abundance of adjectives like "good", and "great", for the pros, and "bad", "poor", "horrible", and "toxic" for the cons, used to quantify metrics like work-life balance and management. Growth and compensation are usually mentioned with "high" and "low", "limited". We see that SBERT (MiniLM) produces slightly worse results than Tf-Idf and SBERT (mpnet) produces slightly better results. Since the differences between each method are only a little, we can not draw reliable conclusions.

However, we do see a significant jump from the ML models to the transformer models. It is interesting to note that DistilBERT and BERT perform similarly, even though the former is 40% smaller. This might be because of the nature of the dataset, allowing both models to perform similarly. RoBERTa comes out as the best model and is perhaps due to its robust training procedure and access to a higher amount of data while model pre-training.

4.8 Conclusion

We saw the work with *Blind Company Reviews*. We first describe the procedure to procure the reviews by scraping review pages of 55 most reviewed companies. We then conduct a metrical and content analysis on the dataset. Finally, we complete our content classification pipeline by adding the ability to mine opinion from the text. After running a total of ten models, we achieve the best results with RoBERTa with 98.29% accuracy.

ML model	Tf-idf Vectorizer	SBERT (MiniLM)	SBERT (mpnet)
Logistic Regression	**0.9438, 0.9434**	0.9330, 0.9330	0.9524, 0.9526
Linear SVC	0.9427, 0.9427	**0.9345, 0.9345**	**0.9553, 0.9554**
Multinomial NB	0.9220, 0.9208	N/A	N/A

Table 4.3: Accuracy and F1 scores of machine learning models with different vectorization techniques

Transformer model	F1 Score	Accuracy
DistilBERT	0.9775	0.9777
BERT	0.9783	0.9786
RoBERTa	**0.9828**	**0.9829**

Table 4.4: Accuracy and F1 scores of transformer models

Chapter 5

Humor Classification

This chapter describes a part of the work done in the paper titled "Towards Conversational Humor Analysis and Design", which has been published in the 11th Humor Research Conference, 2021.

5.1 Introduction

In this chapter, we explore humor classification primarily through a linguistic lens, then, we conduct detailed analyses with the state-of-the-art models.

We first describe the kind of data we are working with. Then, we analyze linguistic features backed by the Incongruity theory and the General Theory of Verbal Humor, as detailed in chapter 2. Using these features, we train multiple machine learning and transformer models and get results on the Reddit dataset for the first time. Our aim with humor classification is to create a model to assess the "jokiness" of a joke – evaluate if a piece of text is humorous or not. We then discuss the limitations of the work and future improvements.

5.2 Data

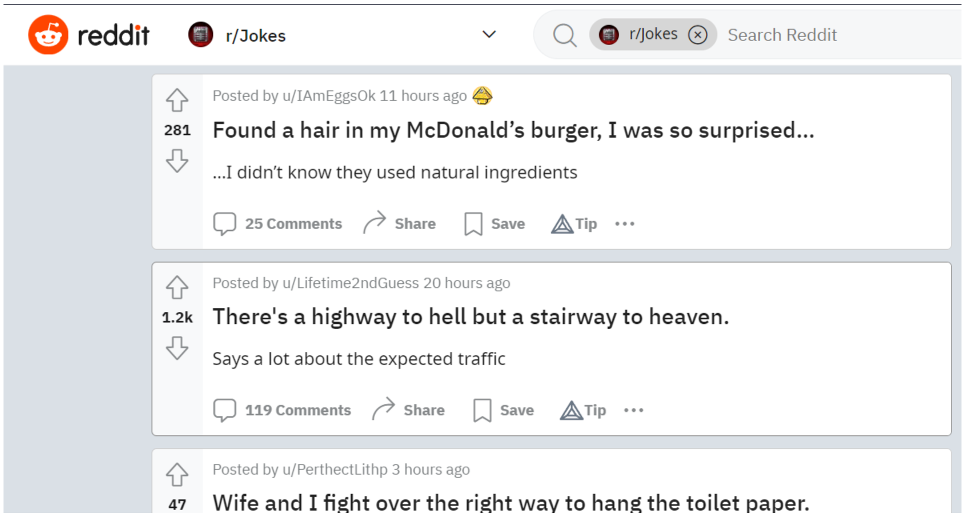
We use an existing dataset of Reddit jokes, called *Short Jokes* present on Kaggle.[1] The dataset majorly uses jokes from *r/jokes* and *r/cleanjokes* subreddits. A sample of the subreddit *r/jokes* can be seen in the figure 5.1. Jokes are scraped using PRAW, a Reddit API wrapper.[2] The data totals to **231,657** humorous samples. Each joke is of length 10 to 200 characters – all the jokes are short.

We use NLTK's *Gutenberg* and *Web and Chat Text* datasets to create a combined non-humorous dataset [26]. *Gutenberg* texts are a collection of 25,000 e-books, which we extend with *Web and Chat Text*, containing 10,000 posts. Note that any personally identifying information was already removed from the dataset. Since each book or post is longer as compared to the short jokes, we split the content by

[1] https://www.kaggle.com/datasets/abhinavmoudgil95/short-jokes

[2] https://github.com/praw-dev/praw

the punctuation, '.'. This allows us to not only get an appropriately lengthed sample but also allows us to get a balanced dataset in terms of non-humorous samples.



Posted by u/IAmEggsOk 11 hours ago

Found a hair in my McDonald's burger, I was so surprised...

...I didn't know they used natural ingredients

Posted by u/Lifetime2ndGuess 20 hours ago

There's a highway to hell but a stairway to heaven.

Says a lot about the expected traffic

Posted by u/PerthectLithp 3 hours ago

Wife and I fight over the right way to hang the toilet paper.

Figure 5.1: A view of Reddit's *r/jokes* subreddit

5.3 Feature Extraction

We use the six linguistic features that performed the best based on previous works [16, 17]. Each of them is explored below.

5.3.1 Alliteration chain length

At the first level, humor makes use of phones – specifically, we focus on alliteration. A phone is a distinct, smallest meaningful unit of speech sound. We use the CMU pronunciation dictionary to obtain the phones and extract alliteration chain length.[3] The hypothesis is that humorous texts have longer instances of alliteration chains [47]. In our general dataset, we might have some cases of alliteration from the Gutenberg texts, but we assume none from Web and Chat texts. The following example demonstrates a case of alliteration.

"Some people come here to sit and think I come here to shit and stink."

Figure 5.2a shows the alliteration chain length distributions. We see that there are some differences in the distribution, with the humorous texts having an overall increased length of the chain. The longer chains, while low in number, are especially significant to differentiate.

[3] `http://www.speech.cs.cmu.edu/cgi-bin/cmudict`

5.3.2 POS ratios

Going a level above, we focus on the morpho-syntactic level. Since most of the data points are of short length (200 characters or less), a full syntactic parse becomes unsuitable for a majority of instances [17]. Instead, we focus on Part-Of-Speech (POS) tags.

We consider the POS tag ratios of nouns, pronouns, verbs, and proper nouns. That is the number of specified tags to the total count of tokens in the sentence. The hypothesis is that humorous texts have a different probability distribution of POS tags. We expect this, especially when it comes to pronouns since humor is often on, or, about someone or something.

Figures 5.2b, 5.2c, 5.2d, 5.2e shows the distribution of all the POS-tag ratios. We see the differences emerge in the case of nouns, pronouns, and verbs. The ratios of proper nouns show very few differences. This might be because the Gutenberg texts have character names and place names, as much as the humorous texts have external name entity references. We explore this idea further in the section 5.6.

5.3.3 Antonym pairs

Next, we experiment with the lexico-semantic level by counting the antonym pairs. We use WordNet to find the antonym relations using WordNet's synsets [48]. Directly following from the incongruity theory of humor, the hypothesis is that humorous texts have a higher number of antonyms, or, "incongruities". The following example demonstrates the use of antonym pairs: "clean" and "cluttered" and "modest" and "proud".

*"A **clean** desk is a sign of a **cluttered** desk drawer. Always try to be **modest** and be **proud** of it!"*

Figure 5.2f shows a small difference between general and humorous samples of data.

5.3.4 Discourse markers

Next, we move to the pragmatic level, giving attention to the discourse connectives [49]. While the text length is uniformly distributed, for the most part, we need to be able to handle longer humorous samples. To do this, we create a custom list of discourse connectives, containing cases of elaboration – "in addition to", contrast – "whereas", etc. While the aforesaid are more "formal" in nature, social media text contains numerous "non-formal" cases. For example, "like", "sort of", "and then", and "to be honest", are examples of unigram, bigram, and trigram discourse markers. Note that discourse connectives like "sort of" can be also used as "I like that sort of candy", but we expect the overall distributions of the connectives to be different for humorous and non-humorous texts. The following joke shows the use of a discourse connective.

*"Some day, Canada will take over the world. **And then** we'll all be sorry."*

Figure 5.2g shows a small difference between general and humorous samples of data.

5.3.5 Polarity

We now take a look at the affective level, by using polarity. The hypothesis is that humor evokes some emotion – whether it is positive or negative [50]. We found a lot of examples of negative polarity jokes (and rarely any feel-good or wholesome jokes). We use SenticNet version 6.0 to calculate the polarity of the sentence, which is a score between -1 (extreme negativity) and +1 (extreme positivity) [51]. In our implementation, we take the summation of the absolute polarities of each of the words in the sentence. This allows us to look at the overall strength of the sentence, without worrying about the positives and negatives canceling out. For words that do not have a mapping, we default to 0. This way, we expect the humorous samples to have overall higher scores than non-humorous ones. Following is an example of a negative polarity joke.

*"I'm **deathly afraid** of elevators. I take a lot of steps to **avoid** them."*

Figure 5.2h shows a significant difference between general and humorous samples of data, especially the ones with more than a score of 3.

5.3.6 Slangs

On platforms like Reddit, a sizeable number of instances of adult humor is present [52]. To detect and appropriately handle such cases, we curate a list of slang words, and words commonly used in adult humor. "Slangs" is a broad umbrella, including words like, "sex", and "fuck", along with gender terms like, "gay" and "lesbian", etc. The hypothesis is that slangs and words like the ones mentioned above are present more in humorous texts than in non-humorous texts. Figure 5.2i shows a difference between general and humorous samples of data.

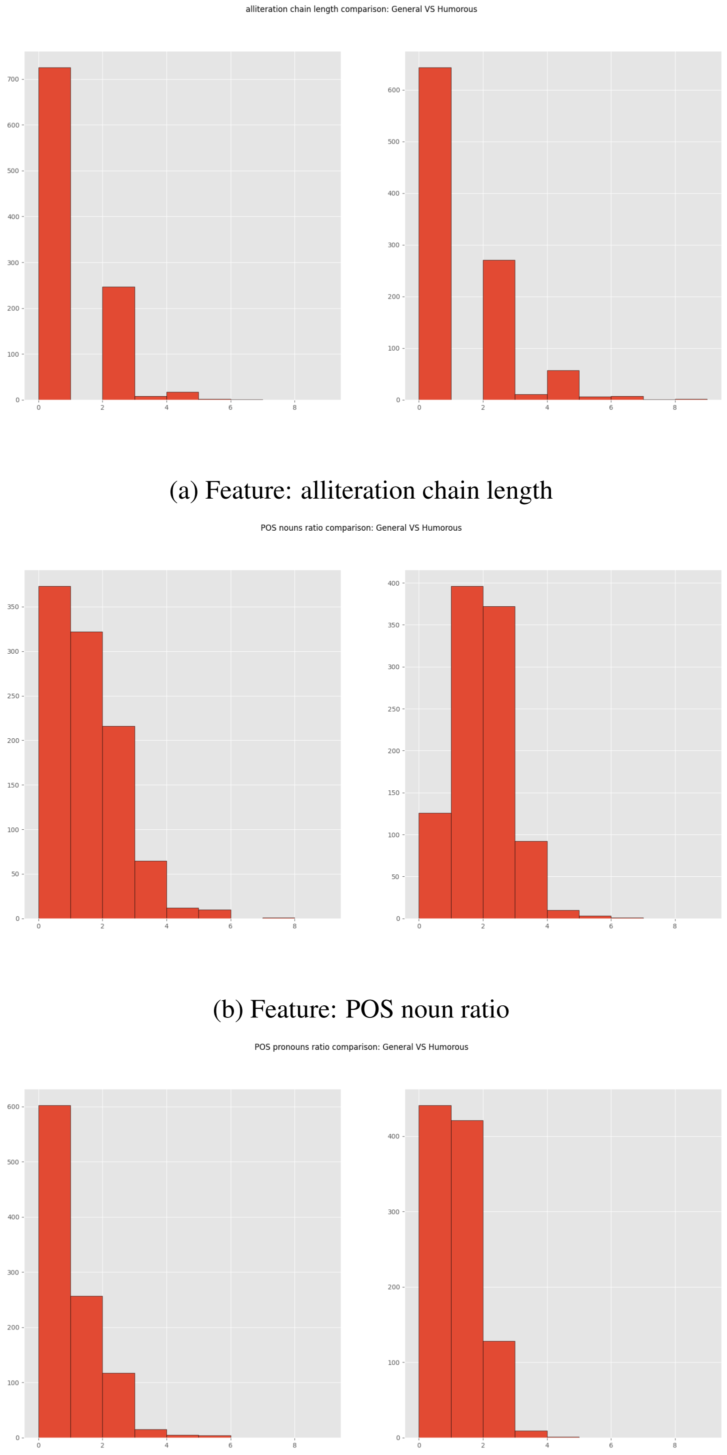
Following are two examples of the abundance of slang and the presence of adult humor in the dataset.

*"My mates keep calling me **gay**, so to prove them wrong I went out and **fucked** this **sexy** nurse.*
*He definitely wasn't **gay**."*

"A man calls in sick and his boss replies and asks 'How sick are you?'
*Well, I am **fucking** my sister so pretty sick."*

5.4 Experiments

Some of the features above have large differences in the distributions of the general and humorous texts and can be used directly. However, the features that have lesser differences can be paired or combined alongside other features to generate a better prediction. Therefore, to classify humor, we feed the above features to machine learning (ML) models as described in chapter 2. We train Logistic Regression, Gaussian Naive Bayes, and Linear Support Vector Classifier models.

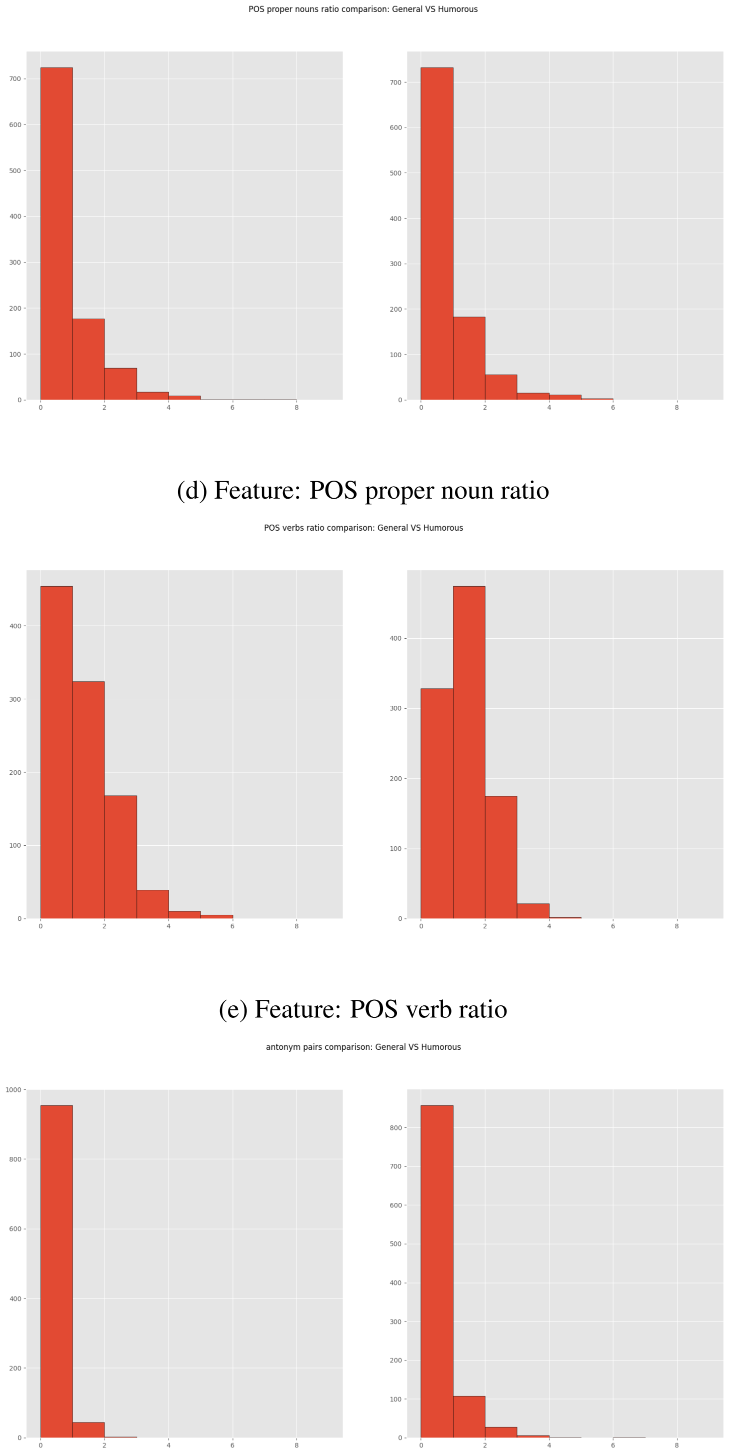


(a) Feature: alliteration chain length

(b) Feature: POS noun ratio

(c) Feature: POS pronoun ratio

Figure 5.2: Histogram comparisons of different features in non-humorous vs humorous texts

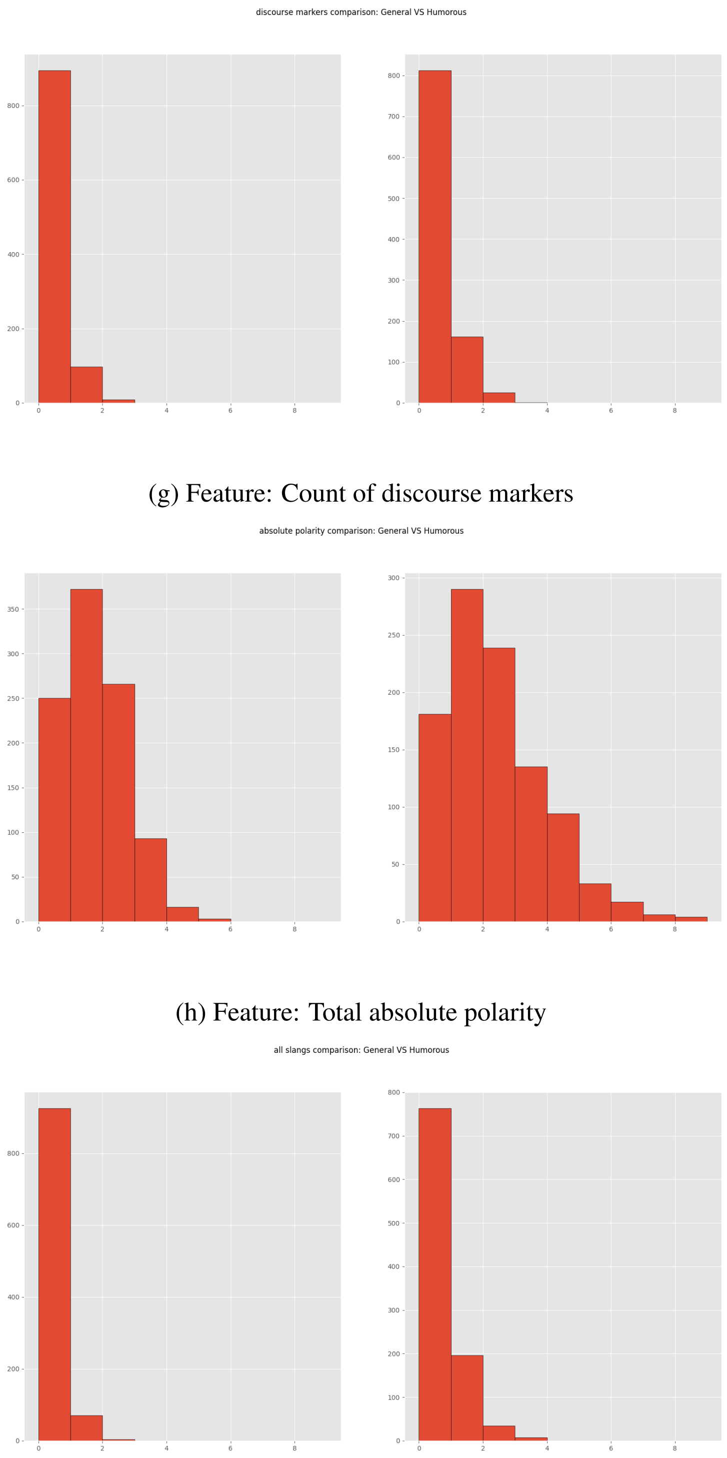


(d) Feature: POS proper noun ratio

(e) Feature: POS verb ratio

(f) Feature: Count of antonym pairs

Figure 5.2: Histogram comparisons of different features in non-humorous vs humorous texts (contd.)



(g) Feature: Count of discourse markers

(h) Feature: Total absolute polarity

(i) Feature: Count of slangs

Figure 5.2: Histogram comparisons of different features in non-humorous vs humorous texts (contd.)

We also compare the performance of ML models with transformers: DistilBERT, BERT, and RoBERTa.

We use 5-fold cross-validation for training and evaluation of the models. All the ML models are trained with default parameters. For the transformers, we vary the learning rate between 2e-5 and 5e-5 and train them for 2 epochs, with 16 as the batch size.

5.5 Results

For the ML models, we see the highest accuracy with the Gaussian Naive Bayes model: 62.5%. Support Vector classifier has the results in a similar range, with an F1-score of 63%. Logistic regression performs the worst, barely touching the 60% mark in terms of accuracy. Table 5.1 summarizes these.

For the transformer models, we see accuracies jump and reach up to 98.90% from the RoBERTa model. The DistilBERT and BERT models show accuracies in a similar range. Table 5.2 summarizes all the results.

Model	**F1**	**Accuracy**
Logistic Regression	58%	60.5%
Gaussian Naive Bayes	59%	**62.5%**
Support Vector Classifier	**63%**	62%

Table 5.1: Accuracy and F1 scores of ML models after feature extraction

Model	**F1**	**Accuracy**
DistilBERT	98.66%	98.65%
BERT	98.73%	98.73%
RoBERTa	**98.90%**	**98.90%**

Table 5.2: Accuracy and F1 scores of transformer models

5.6 Discussion

The ML and transformer models show a chasm in accuracies. Given that the dataset is balanced, the ML results show that specific linguistic features are indeed useful for determining the jokiness of a joke, since the model performs better than random. However, the model performs poorly in comparsion to others with similar tasks. For example, previous works on Twitter data [17] and stock humor data [16] achieve close to 84% and 80% accuracies with the same ML models. We explore the reasons for low ML and high transformer accuracies below.

5.6.1 Style

To understand the results from a stylistic angle, we look at the paper working on the Twitter data. Authors find that the feature called "Twitterspeak_url" – which counts the number of URLs present in the data, has the most significant contribution to the model accuracy. Other features that exploit '#'s and '@'s are also found important – which are not at all present in the Reddit dataset – making our task harder, stylistically.

The ML model thus loses out on social-media-specific features. On the other hand, transformers use subword tokenization, allowing them to not only work with out-of-vocabulary (OOV) data but also produce a fine-grained representation that can be used to build specific contexts [53, 54].

5.6.2 Semantics

From a semantic perspective, there are a lot of differences between the Reddit dataset and the one-liners dataset. A deeper analysis shows that the jokes on Reddit require more world knowledge to understand than stock humor. For example, the following joke from Reddit draws on a lot of external information.

"I just saw the Assassins Creed Movie Trailer...
I did not expect The Spanish Inquisition."

This requires one to know what Assassin's Creed is (a popular game franchise), and a specific game in the series, which involves the Spanish Inquisition. Moreover, this joke is a spinoff of the popular line, "Nobody expects the Spanish Inquisition!" from Monty Python.

The jokes are also inspired by a conversational and sitcom style, as shown below [55]. The humor not only emerges from the raw text content but also from the setup; as if the conversation took place. The author even mentions the location in the very first line in square brackets.

"[Touring Italy]
Guide: Bathroom anyone?
Me: I peed at the Tower of Pizza
Guide: That's Pisa
Me: Sorry. I took a pisa at the Tower of Pizza"

While the ML models struggle with longer, narrative jokes and jokes requiring world knowledge, transformers ace once again. Due to extensive pre-training with vast amounts of data from the Internet, they become sources of information and act as databases.

We explore the reasons for high transformer accuracies further in chapter 6.

5.7 Conclusion

We conducted our experiments on 231k jokes from Reddit augmented with negative samples from Gutenberg and Web and Chat texts data. We identified six crucial high-performing linguistic features from past works at all levels – from phonetic to pragmatic – and found their distributions for the combined dataset. Training multiple ML models, we see the highest accuracy of 62.5% by the Linear Support Vector Classifier. While better than random, the accuracy hints at the unique style and semantic challenges of the Reddit datasets and the high complexity of the task itself – detecting humor. As we move to the transformer architectures, we see a significant jump in the model accuracies, reaching 98.90%, due to their ability to capture OOV data, conduct sub-word tokenization and most importantly, act as natural language databases.

Chapter 6

Conclusion & Future Work

6.1 Cross-Task Model Analysis

We now analyze model results for all the classification tasks – fine-grained *Blind Posts*, *Blind Company Reviews*, and Reddit humor. We note a **consistent** jump in the accuracies from the best ML models to the best (in fact, even the worst) transformer models (Figure 6.1).[1] We note several reasons are possible for this jump in accuracy.

Both the Tf-Idf vectorization and the machine learning models could be improved. Tf-Idf is only based on counts and does not have any "meaning" other than the one derived from a purely lexical level. Then, the machine learning model is trained from scratch with the vectors. In contrast, transformers use multi-headed self-attention to understand sentence structure [56, 57]. The self-attention mechanism allows the model to focus on certain parts of the sentence more than others, by weighting them higher or lower. With multiple heads of this mechanism, the model can learn to focus on multiple important parts of the sentence [58]. Moreover, as we stack the encoders on top of each other, they mimic the natural hierarchy of syntax in language, allowing for a better internal representation of language in the model [59].

The transformer models are all pre-trained large language models (LLMs). The "pre-trained" part refers to the fact that the model has been trained on a large amount of text. In this way, the model effectively captures facts about the world, which are exploited in downstream tasks [60]. In the case of humor classification, we had to first extract the features and then train models on the same. In selecting the features, we might have missed out on other features, which can be derived from the same piece of text. As explained in chapter 5, LLMs can exploit their world knowledge to "understand" the jokes. Even for post and review classification, "understanding" the bigger context of company reviews is helpful since opinions can be sarcastic or ironic, demanding a flip in opinion polarity.

We thus see how both vectorization and models are improved when it comes to transformers – the model learns contextual word (token) embeddings and its pre-trained nature lets it understand language better.

[1] Note that we consider all machine learning models with the Tf-Idf vectorization.

model↓ task→	**Humor**	**Blind Posts**	**Blind Company Reviews**
Best ML model	GNB: 62.5%	L-SVC: 72.32%	LR: 94.38%
Best transformer model	RoBERTa: 98.9%	RoBERTa: 78.41%	RoBERTa: 98.29%

Table 6.1: Machine learning and transformer models accuracy comparison for each classification task

6.2 Conclusion

In this thesis, we explored the world of text classification in social media texts, by taking a dual perspective.

We first explored the previously untapped platform, Blind, which rose in popularity due to its promise of anonymity for workplace discussions. The two novel datasets, *Blind Posts* and *Blind Company Reviews* contain a total of 767k and 63k data points from seven years of industry data, respectively. These datasets proved to be of high validity as user anonymity fostered and encouraged productive discussions on otherwise taboo subjects. We explored and exploited the platform's bias toward the tech industry, by finding reflections from landmark events like COVID-19, work-from-home, return-to-office, and layoffs, and conducted a full temporal analysis. We also found surprising correlations and revelations about what employees say is important, versus what they quantitatively vote to be. Combining the utility of the two novel datasets, we proposed a novel content classification pipeline with accuracies of 99.25% for filtering relevant data, 78.41% for annotating, and 98.29% for mining opinions, showing a high level of practicality.

Turning to humor, we leveraged the linguistic richness of humor theories and tested their validity and applicability in modern humor as seen on Reddit. We found six features from past works, ranging from the phonetic level to the pragmatic level – allowing us to accurately assess the humor theories. We found the machine learning models are severely restricted and perform at an accuracy of 62.5%, due to unique stylistic and semantic issues present in Reddit, not present in the past works. We improved the results using transformers, getting up to 98.90% accuracy, deliberating the benefits of rigorous pre-training and acting as a knowledge database, as seen in the section above.

6.3 Future Work

As detailed in the chapters 3 and 4, the novel content classification pipeline can be productionized. This pipeline can live on a separate platform that aggregates news/information/opinions of tech from all across the internet. Specifically, the coarse classifier model can be used on Twitter, Quora, and Reddit (and similar social media) platforms. Other model architectures can be used to improve the accuracies of the fine-grained classification model after more data is scraped in the next years. Furthermore, since the opinions on the platform represent the honest opinions of the industry, even across years, trading bots can be implemented to track the live aggregated sentiments from the community and make decisions before news makes it to mainstream media.

One could also try finding correlations from the transformer model to the humor theories – to see what the model learns, how it learns it, what it lacks, and the empirical reasons for high accuracy as qualitatively explored in chapter 5. It would be interesting to see if the performance of transformers can be enhanced even further by combining them with rule-based linguistic features.

Related Publications

1. **Chaudhary, Tanishq**, Mayank Goel, and Radhika Mamidi. "**Towards Conversational Humor Analysis and Design.**" In Proceedings of the Humor Research Conference, 2021.
2. **Chaudhary, Tanishq**, Pulak Malhotra, Radhika Mamidi, and Ponnurangam Kumaraguru. "**Blind Leading the Blind: A Social-Media Analysis of the Tech Industry.**" In Proceedings of the 20th International Conference on Natural Language Processing, 2023.

Bibliography

[1] Ravi Kumar, Jasmine Novak, and Andrew Tomkins. Structure and evolution of online social networks. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 611–617, 2006.

[2] Ping Gong and Jacob K Thomas. Thumb on the scale: Do employers manage glassdoor reviews? *Available at SSRN 4460625*, 2023.

[3] Xiao Yan, Jaewon Yang, Mikhail Obukhov, Lin Zhu, Joey Bai, Shiqi Wu, and Qi He. Social skill validation at linkedin. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2943–2951, 2019.

[4] Joanna Davis, Hans-Georg Wolff, Monica L Forret, and Sherry E Sullivan. Networking via linkedin: An examination of usage and career benefits. *Journal of Vocational Behavior*, 118:103396, 2020.

[5] Graeme Ritchie. Can computers create humor? *AI Magazine*, 30(3):71–71, 2009.

[6] Sean Kanuck. Humor, ethics, and dignity: Being human in the age of artificial intelligence. *Ethics & International Affairs*, 33(1):3–12, 2019.

[7] Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. Tl; dr: Mining reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 59–63, 2017.

[8] Tazeek Bin Abdur Rakib and Lay-Ki Soon. Using the reddit corpus for cyberbully detection. In *Intelligent Information and Database Systems: 10th Asian Conference, ACIIDS 2018, Dong Hoi City, Vietnam, March 19-21, 2018, Proceedings, Part I 10*, pages 180–189. Springer, 2018.

[9] Munmun De Choudhury and Sushovan De. Mental health discourse on reddit: Self-disclosure, social support, and anonymity. In *Proceedings of the international AAAI conference on web and social media*, volume 8, pages 71–80, 2014.

[10] Zheng Ping Jiang, Sarah Ita Levitan, Jonathan Zomick, and Julia Hirschberg. Detection of mental health from reddit via deep contextualized representations. In *Proceedings of the 11th International Workshop on Health Text Mining and Information Analysis*, pages 147–156, 2020.

[11] Faraz Faruqi and Manish Shrivastava. “is this a joke?”: A large humor classification dataset. In Gurpreet Singh Lehal, Dipti Misra Sharma, and Rajeev Sangal, editors, *Proceedings of the 15th International Conference on Natural Language Processing*, pages 104–109, International Institute of Information Technology, Hyderabad, India, December 2018. NLP Association of India.

[12] BNCXML Edition. Version 3 (bnc xml edition), 2007.

[13] Faraz Faruqi and Manish Shrivastava. “is this a joke?”: A large humor classification dataset. In *Proceedings of the 15th International Conference on Natural Language Processing*, pages 104–109, 2018.

[14] Orion Weller and Kevin Seppi. Humor detection: A transformer gets the last laugh. *arXiv preprint arXiv:1909.00252*, 2019.

[15] Issa Annamoradnejad and Gohar Zoghi. Colbert: Using bert sentence embedding in parallel neural networks for computational humor. *arXiv preprint arXiv:2004.12765*, 2020.

[16] Rada Mihalcea and Carlo Strapparava. Making computers laugh: Investigations in automatic humor recognition. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 531–538, 2005.

[17] Renxian Zhang and Naishi Liu. Recognizing humor on twitter. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*, pages 889–898, 2014.

[18] Heewon Kim and Craig R Scott. Going anonymous: Uses and perceptions of anonymous social media in an it organization. In *Proceedings of the 9th International Conference on Social Media and Society*, pages 335–339, 2018.

[19] Heewon Kim and Craig Scott. Change communication and the use of anonymous social media at work: Implications for employee engagement. *Corporate Communications: An International Journal*, 2019.

[20] Lisa Glebatis Perks. The ancient roots of humor theory. *Humor*, 25(2):119–132, 2012.

[21] Salvatore Attardo and Victor Raskin. Script theory revis (it) ed: Joke similarity and joke representation model. *Humor*, 1991.

[22] Maxim Petrenko. *The narrative joke: Conceptual structure and linguistic manifestation*. PhD thesis, Purdue University, 2007.

[23] Christie Davies. *Jokes and targets*. Indiana University Press, 2011.

[24] Christian F Hempelmann and Salvatore Attardo. Resolutions and their incongruities: Further thoughts on logical mechanisms. 2011.

[25] Willibald Ruch, Salvatore Attardo, and Victor Raskin. Toward an empirical verification of the general theory of verbal humor. 1993.

[26] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.", 2009.

[27] Juan Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48. Citeseer, 2003.

[28] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[29] Xiaonan Zou, Yong Hu, Zhewen Tian, and Kaiyuan Shen. Logistic regression model optimization and case analysis. In *2019 IEEE 7th international conference on computer science and network technology (ICCSNT)*, pages 135–139. IEEE, 2019.

[30] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20:273–297, 1995.

[31] James Joyce. Bayes' theorem. 2003.

[32] Gurinder Singh, Bhawna Kumar, Loveleen Gaur, and Akriti Tyagi. Comparison between multinomial and bernoulli naïve bayes for text classification. In *2019 International Conference on Automation, Computational and Technology Management (ICACTM)*, pages 593–596. IEEE, 2019.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[34] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[35] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[36] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[37] Margherita Grandini, Enrico Bagli, and Giorgio Visani. Metrics for multi-class classification: an overview. *arXiv preprint arXiv:2008.05756*, 2020.

[38] Kate Scott. The pragmatics of hashtags: Inference and conversational style on twitter. *Journal of Pragmatics*, 81:8–20, 2015.

[39] Hanadi Buarki and Bashaer Alkhateeb. Use of hashtags to retrieve information on the web. *The Electronic Library*, 36(2):286–304, 2018.

[40] Clayton Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the international AAAI conference on web and social media*, volume 8, pages 216–225, 2014.

[41] Alexander Bick, Adam Blandin, Karel Mertens, et al. Work from home after the covid-19 outbreak, 2020.

[42] Brodie Boland, Aaron De Smet, Rob Palter, and Aditya Sanghvi. Reimagining the office and work life after covid-19. 2020.

[43] Abel Brodeur, David Gray, Anik Islam, and Suraiya Bhuiyan. A literature review of the economics of covid-19. *Journal of economic surveys*, 35(4):1007–1044, 2021.

[44] Ahmed El-Deeb. The first tech layoff wave after years of hypergrowth: How this affects the industry? *ACM SIGSOFT Software Engineering Notes*, 48(1):4–5, 2023.

[45] Timothy R. Hinkin and J. Bruce Tracey. What makes it so great?: An analysis of human resources practices among fortune's best companies to work for. *Cornell Hospitality Quarterly*, 51(2):158–170, 2010.

[46] David E Kanouse and L Reid Hanson Jr. Negativity in evaluations. In *Preparation of this paper grew out of a workshop on attribution theory held at University of California, Los Angeles, Aug 1969.* Lawrence Erlbaum Associates, Inc, 1987.

[47] Maharani Widya Putri, Erwin Oktoma, and Roni Nursyamsu. Figurative language in english stand-up comedy. *English Review: Journal of English Education*, 5(1):115–130, 2016.

[48] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[49] Ksenia Shilikhina. Discourse markers as guides to understanding spontaneous humor and irony. In *The Dynamics of Interactional Humor*, pages 57–76. John Benjamins, 2018.

[50] Sven van den Beukel and Lora Aroyo. Homonym detection for humor recognition in short text. In *Proceedings of the 9th workshop on computational approaches to subjectivity, sentiment and social media analysis*, pages 286–291, 2018.

[51] Erik Cambria, Yang Li, Frank Z Xing, Soujanya Poria, and Kenneth Kwok. Senticnet 6: Ensemble application of symbolic and subsymbolic ai for sentiment analysis. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 105–114, 2020.

[52] Leonard Tang, Alexander Cai, and Jason Wang. The naughtyformer: a transformer understands and moderates adult humor (student abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 16348–16349, 2023.

[53] Marcio Inácio, Gabriela Wick-pedro, and Hugo Gonçalo Oliveira. What do humor classifiers learn? an attempt to explain humor recognition models. In *Proceedings of the 7th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 88–98, 2023.

[54] Eugene Kharitonov, Marco Baroni, and Dieuwke Hupkes. How bpe affects memorization in transformers. *arXiv preprint arXiv:2110.02782*, 2021.

[55] Anna Šmilauerová. Tv sitcom friends: Analysis of character humor strategies based on the violation of grice's conversational maxims. 2012.

[56] Dichao Hu. An introductory survey on attention mechanisms in nlp problems. In *Intelligent Systems and Applications: Proceedings of the 2019 Intelligent Systems Conference (IntelliSys) Volume 2*, pages 432–448. Springer, 2020.

[57] Xiaobing Sun and Wei Lu. Understanding attention for text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3418–3428, 2020.

[58] Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. Multi-head attention: Collaborate instead of concatenate. *arXiv preprint arXiv:2006.16362*, 2020.

[59] Antoine Simoulin and Benoit Crabbé. How many layers and why? an analysis of the model depth in transformers. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 221–228, 2021.

[60] Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. *arXiv preprint arXiv:2104.08696*, 2021.