THE EFFECT OF SENTIMENTS IN COMMUNICATION ON A TEAM PERFORMANCE

By
Abay Jumabayev

Submitted to
Central European University
Department of Economics and Business

In partial fulfillment of the requirements for the degree of
Masters of Arts in Economics

Supervisor: Gábor Békés

Vienna, Austria
2022

Abstract

There is a lot of literature exploring the determinants of team performance. Knowing how to improve performance could help in any field where people work in teams. I test how the sentiments in communication, for example, being rude or disrespectful towards the other person, affect the performance of a team. A team is a small group of four to six people that (i) are educated and highly paid, (ii) work together for a short period (like projects), and (iii) tend to work remotely. Especially for teams where the line-up changes frequently, it is beneficial to know the effect of toxicity in communication on teamwork because sentiments affect the mood of teammates and the mood affects the performance. As it is difficult to quantify the performance and communication between team participants, such as management consultants, I use Dota 2, an online game, to examine how being toxic affects the performance of a team. Apart from fitting the definition of a team, a team from Dota 2 has measurable performance, win or lose, and a chat log from which the derivation of sentiments is possible. With a dataset of 50,000 Dota 2 matches from an open-source platform, I first calculate whether the message sent was toxic. Then, I apply linear regression to predict the outcome of a match via toxicity and controls, such as game and player attributes. I found an economically and statistically significant negative link between a toxic team and its performance.

Keywords: sentiments, toxicity, communication, Natural Language Processing, performance, teamwork, linear regression.

Acknowledgments

Words cannot express my gratitude to my supervisor, Gábor Békés, for his invaluable patience and feedback. I am also grateful to my classmates and cohort members, especially Damira, Anna, and Che, for their editing help, late-night feedback sessions, and moral support. Thanks should also go to the university community, who impacted and inspired me.

Lastly, I would be remiss in not mentioning my family. Their belief in me has kept my spirits and motivation high during this process. I would also like to thank my fiancé who kept saying yes to all my endeavors while providing emotional support.

Table of Contents

List of tables

Introduction

I want to understand the effect of certain sentiments in communication on the performance of a team. Being rude to a team or just to one member can change the mood of the team, which in turn could deteriorate the way the team works. As people are social creatures and spend a significant amount of time performing a job, it is important to understand whether social behavior has an impact on team performance. I reduce a definition of a team to a small group (4 to 6 people) that consists of workers who are (i) high educated, better-paid workers, (ii) who frequently change teams, and (iii) who tend to work remotely. Job positions like a management consultant, software engineer, or researcher fit into the definition of workers in the team.

However, it is difficult to get the data on all the communication between team members and quantify the performance. That is why I focus on the Dota 2, a multiplayer online battle arena (MOBA) video game. In this game people, communicate and accomplish tasks in a team, which is similar to the work environment. Due to the recent pandemic, remote jobs are getting popular, and workers shift to using chats instead of verbal communication. This makes communication in work teams look more similar to the one in Dota 2. Moreover, the performance in the game is measurable and equal to the outcome of a match.

I use the dataset parsed from Opendota, an open-source platform, containing 50,000 matches. Using another dataset of texts labeled by toxicity I build a model that takes a text as an input and outputs whether the text was toxic or not. I define toxicity as a form of anti-social behavior, and a toxic text is a text that is rude and disrespectful to a reader. Using the model, I label every message sent in Dota 2 chats. Then I transform Dota 2 data to calculate teams' toxicity levels, in-game attributes, and teams' skill levels in order to use an OLS regression.

OLS results show that toxicity decreases the performance of a team. Having an extra toxic player in a team reduces the chances of winning a game by 1%. The coefficient is statistically significant at a 1% level. This result brings a new determinant of team success, which, to the best of my knowledge, was not considered before in the literature.

The rest of the paper is organized as follows: the next section provides a literature review that covers why Dota 2 teams are similar to job teams, how sentiments in communication affect team performance, and how toxicity in communication affects the performance in the game industry. Section 2 describes two main data sources that I use for my analysis. Section 3 describes the model that predicts toxicity, the data manipulations that are necessary for conducting an analysis, and the econometric method of the estimation. The last section provides the results and discussions, followed by a conclusion.

1. *Literature review*

Before proceeding to the description of data and methods it is important to understand what is known about three things: why Dota 2 teams are similar to the teams working in the industry, how toxicity affects team performance in the job industry, and how toxicity affects team performance in the game industry.

1.1 Similarity between Dota 2 teams and job teams

As the Dota 2 is an online game, the teams usually do not communicate with each other in person. That is why I want to focus on teams who do not see their colleagues in person. Due to COVID-19, a lot of employees had to shift to a work from home mode reaching a 47.7% in the US as of April 2020.[1] According to the US survey conducted by Mercer LLC (2021), 70% of employers plan to adopt a hybrid work model after the end of the pandemic.[2] That means that most of the team in jobs can be compared to the teams in Dota 2 in terms of non-face-to-face communication. Moreover, Bartik et. al (2020) found that remote work is much more common in industries with better educated and better-paid workers.[3] Dota 2 team can be compared to a job team also because both teams have a motivation to perform well. I analyze only ranked Dota 2 matches, which means each player earns points for winning and loses points for losing a match. Points in the ranking system play a role of respect, praise, and honor for players, which is a good motivation for players.[4] I want to compare Dota 2 teams to job teams that often change their composition. The reason is that in Dota 2 players are assigned to a team

[1] Shockley et al., "Remote Worker Communication during COVID-19."
[2] Mercer LLC, "US Flexible Working Policies & Practices Survey."
[3] Bartik et al., "What Jobs Are Being Done at Home During the Covid-19 Crisis?"
[4] Bostan, "Player Motivations."

randomly based on a skill level. In addition, given that in Dota 2 a team consists of five players, the comparable team should also be small in size.

Having all this information I come up with the definition of a team that is comparable to the Dota 2 team. A team is a small team (4 to 6 people) that consists of workers who are (i) high educated, better-paid workers, (ii) who frequently change teams, and (iii) who tend to work remotely. Job positions like a management consultant, software engineer, or researcher fit into the definition of workers in the team.

The major difference is that the Dota 2 games last usually no longer than one hour whereas job teams work for a long time. However, the first impression of individuals in a team affects future team interaction. Bizarro (2013) citing the work of Gersick mentioned that the first impression dominates the overall teamwork strategy for about half of the team's existence.[5] Thus, I assume being toxic to the team in the first hour affects the way the team operates in future times as well.

1.2 How sentiments in communication affect teams

There is a lot of literature on the effect of sentiments in communication on teams. Marlow et. al. (2017) argue that communication quality positively affects team performance.[6] According to the authors, communication quality is "clarity, effectiveness, accuracy, and completeness of communication".[7] According to the study by Alinor (2022), people experiencing microaggressions felt negative emotions which in turn limited their ability to use a more objective rationale for decision making.[8] Microaggressions are intentional or

[5] Bizarro, "The Distinct Roles of First Impressions and Physiological Compliance in Establishing Effective Teamwork."

[6] Marlow, Lacerenza, and Salas, "Communication in Virtual Teams."

[7] Marlow, Lacerenza, and Salas.

[8] Alinor, "Research."

unintentional behaviors that communicate negative racial slights and insults toward people of color.[9] However, I did not find any literature that assesses the effect of toxicity in communication on team performance.

1.3 Toxicity in online games

I found three papers that study the effect of toxicity in game chats on game success. Martens et al. (2015) found no strong correlation between the win rate of a player and the count of toxic words he/she sends to a chat.[10] The authors also found that the losing team starts using more toxic words in the late stage of a match, although the level of toxicity is consistent throughout the earlier periods. Traas (2017) continued the work of Martens et al. and focused on the impact of toxicity on the match outcome.[11] He found that the team that sends toxic messages has fewer chances to win a match. Monge and O'Brien analyzed another MOBA game called League of Legends and found that the teams in a toxic condition performed significantly poorer.[12]

In this work, I analyze how sending toxic messages affects the outcome of a match. My paper differs from the earlier works in three ways. First, I use newer and much larger data: 50,000 matches from 2015 versus 13,000 matches from 2012 used by Martens et al. and Traas. Second, I build a model that predicts whether the message is toxic based on a train set that was rated by human readers. Applying three vectorization techniques and six classifiers per vectorization technique I came up with 18 models. This machine learning technique identifies whether the message is toxic and performs better than labeling messages by the presence of

[9] Alinor.

[10] Märtens et al., "Toxicity Detection in Multiplayer Online Games," 4.

[11] Traas, "The Impact of Toxic Behavior on Match Outcomes in DotA," 20.

[12] Monge and O'Brien, "Effects of Individual Toxic Behavior on Team Performance in *League of Legends*."

profane words in it. Lastly, I use linear regression to evaluate how toxicity contributes to the prediction of match outcomes.

2. Data

For answering the question, I use two datasets: data on Dota 2 matches and data on labeled texts. Texts that are labeled as being toxic or not are required for building a model that predicts whether an inputted text is toxic. Then, I feed Dota 2 messages into the model and receive a label on whether the message is toxic. Finally, I use Dota 2 matches information, including the messages, to understand how being toxic affects the performance of a team.

2.1 Dota 2 matches

Dota 2 is a 5 vs 5 online battle arena, where two teams – radiant and dire – compete to destroy a large structure called an "Ancient", belonging to the opposing team. After the fall of the "Ancient" match is considered to be won and the player could search for the next match, where he/she will be matched up with a team against the other based on skill. I use the dataset parsed from Opendota, an open-source platform, containing 50,000 ranked matches played between November 6 and November 18 of 2015.[13] The more recent dataset will provide similar results because the game was not changed in terms of chat interaction. The dataset contains three sources of information that I use: (1) match data, which includes duration, the outcome of the match, the server where the game was played, whether there were players who left the game during the match, etc; (2) player data, which includes player attributes like skill and game attributes like gold, experience, last hits, kills, deaths, for each of ten players in a match; (3) chat data, which includes all messages sent to a chat by each player.

In Dota 2, there are two channels for sending a message: players either send a message to allies or to all. I use the data for both channels because it does not matter who the target of antisocial behavior is. Moreover, players can use "to all" chat even if they are mad at their

[13] "Dota 2 Matches."

teammates, blaming the teammate in front of opponents. The main limitation is that in the game players can communicate not only through chat but also through voice and pings, marks on the map, which can tell the players' intention: warn about enemies, express a desire to attack/retreat or indicate players' next movements. I do not have voice and pings data, so my analysis focuses on chat messages only. Nevertheless, it is reasonable to assume that the text chat reflects the sentiments of players through other channels. For example, if one of the players is very rude towards teammates in the chat, with a high probability the player and teammates would be also rude in the voice chat. Apart from the three mentioned information sources, there is a lot of information available that I ignore because of its irrelevance to the analysis: the characters players choose to play, the items they buy, the abilities they upgrade, the team fights information, etc. The raw data is then transformed in such a way that each observation is one team where most of the variables are the averages among five players in the team. As there are 50,000 matches between two teams, the data can contain a maximum of 100,000 observations: two observations for each match.

2.2 Labeled texts

For building a model that predicts whether a player sent a toxic message, I need train data with labeled texts. In 2018, the Conversation AI, a part of Google, announced a toxic comment classification challenge, which includes the definition of toxicity and the data labeled by human raters.[14] I define toxicity the same way it is defined by Conversation AI. Toxicity is a form of anti-social behavior, and a toxic message is a message that is rude and disrespectful to at least one player.[15] The dataset provided by Conversation AI includes about 160,000 comments in the English language from Wikipedia's talk page edits with dummy labels on

[14] Conversation AI, "Toxic Comment Classification Challenge."
[15] Conversation AI.

whether the comments are toxic. Wikipedia's talk page, also known as the discussion page, is the page where editors comment on improvements to Wikipedia articles. I use this dataset to label about 1.5 million chat messages sent in 50,000 games. Although Wikipedia comments are different from game chat messages, in general, they share some similarities. First, both texts are written online and may share a similar vocabulary of jargon; second, both texts are written to interact with participants in a discussion.

3. Methods

First, I build a model that predicts whether a text is toxic or not. Then I label each message sent by players with 1 if it is toxic and 0 otherwise. Out of the 50,000 games I have data on, I pick a subsample of 36,030 games that are ready for analysis. Lastly, I run an OLS regression of toxicity on the game performance.

3.1 Predicting toxicity

The first step in predicting toxicity is to preprocess the data, both train (Wikipedia comments) and test (Dota 2 chat messages). Text preprocessing is a method of data cleaning that is important for the model to efficiently determine whether the text is toxic. There are various ways people can deliver their thoughts through text. Not only language can differ, but also the way of writing it: punctuation, capitalization, or grammatical errors. I begin the preprocessing by removing all e-mails and URLs: I use regular expressions to get rid of information that is not related to toxicity. The second step is to remove all the alpha-numeric characters except white space, that is removing characters (including punctuation marks and new lines) except English letters and white spaces. The next step is to remove all the stop words. Stop words are the most common words in a language that do not add much information to the text. Stop words usually include articles, prepositions, pronouns, conjunctions, and topic-related words that often appear in the text. Examples of a few stop words in English are “the”, “a”, “I”, “so”, and “what”. Removing these type of words help two things: first, it helps the model to predict the toxicity by removing low-level information from the text, and second, it reduces the dataset size, which decreases the time for computations. I use the nltk library, which provides an extensive list of stop words, and stop words that are game-specific, like hero and

item names, that do not affect the toxicity of the message.[16] The last step of preprocessing is a text normalization technique – lemmatization – switching any kind of a word to its base root mode. I want the model to treat the words "play", "playing", and "playable" as one word because all of them convey the same meaning. Lemmatization also reduces the dataset size significantly.

After the text preprocessing, I need to convert each document, a comment for Wikipedia comments dataset and a message for Dota 2 dataset, to a vector of numbers. This process is called vectorization, which can be performed in different ways, and different vectorization algorithms affect the results of the model. That is why I use three techniques of vectorization: TF-IDF, Word2Vec, and GloVe.

TF-IDF stands for term frequency-inverse document frequency and it is a statistical measure that evaluates how relevant a word is to a document in a collection of documents. This is done by multiplying two metrics: how many times a word appears in a document, and the inverse document frequency of the word across a set of documents, a corpus. Mathematically:

$TF_{ij} = \frac{f_{ij}}{n_j}$, where f_{ij} is the frequency of term i in document j , n_j is the total number of words in document j.

$IDF_i = 1 + \log\left(\frac{N}{c_i}\right)$, where N is the total number of documents in the corpus, c_i is the number of documents that contain the word i.

$w_{ij} = TF_{ij} * IDF_i$, where w_{ij} is the TF-IDF score of term i in document j.

The more frequent a word is in the document, the higher the TF-IDF score it has, but also the more frequent a word is across all documents, the lower TF-IDF becomes. For

[16] Bird, Klein, and Loper, "Natural Language Processing with Python."

example, the word “game” would appear often in every chat message, but also it appears often across all other messages, that is why the word should have a lower score. Meanwhile, the word “hate” can appear often in a particular message, and it is not expected to appear often across the corpus, leading to a word having a higher score. Performing TF-IDF would convert all text documents into a matrix with the shape $m * n$, where m is the number of documents and n is the number of unique words across the corpus.

Word2Vec is a two-layer neural network that is trained to reconstruct linguistic contexts of words. After training, Word2Vec models can be used to map each word to a vector of typically several hundred elements, which represent that word’s relation to other words. I use the SpaCy library to get a pre-trained model.[17] This pre-trained model allows to convert each word to a vector of length 300. As each document has several words, the vector for a sentence would be an average of vectors for each word in the document.

GloVe is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space. GloVe and Word2Vec are similar because semantically similar words have vectors close together in terms of cosine distance. The two models differ in the way they are trained and hence lead to word vectors with subtly different properties. GloVe model is based on leveraging global word-to-word co-occurrence counts leveraging the entire corpus. Word2vec on the other hand leverages co-occurrence within local context (neighboring words). For GloVe I also use pre-trained word vectors from Wikipedia (6 billion tokens, 100-dimensional vectors).[18]

[17] Honnibal et al., “SpaCy: Industrial-Strength Natural Language Processing in Python.”
[18] Pennington, Socher, and Manning, “GloVe: Global Vectors for Word Representation.”

After performing vectorization, I have documents converted to vectors of numbers that I can feed to the model to predict toxicity. I apply six classifiers for each type of vectorization algorithm to find the model that best predicts toxicity. Those classifiers are linear regression, logistic regression, naïve Bayes, decision tree, k-nearest neighbors, and support vector classifier. The metric for best prediction is precision, the ratio of true toxic comments divided by predicted toxic comments. I want to reduce the type II error, that is to decrease the number of false positives, non-toxic comments that were predicted toxic. Apart from that, the accuracy of each model is important. I split about 160,000 Wikipedia comments into train and test sets. I take a stratified sample of 80% of all data and use it to train the model. The rest of the data is a test set that is used to assess the performance of the models. I perform a stratified sampling because the data is unbalanced: out of 159,571 comments, only 15,294 are toxic, which is about 10% of all data. After performing stratified sampling, both train and test data sets include about 10% of toxic messages. The results of the models' performances can be seen in Table 1.

Table 1. Performances of models

	TF-IDF		Word2Vec		GloVe	
	Precision	Accuracy	Precision	Accuracy	Precision	Accuracy
Linear regression	0.57	0.91	0.92	0.94	0.87	0.93
Logistic Regression	0.93	0.96	0.84	0.95	0.76	0.93
Naive Bayes	0.75	0.94	0.47	0.90	0.39	0.87
Decision Tree	0.88	0.94	0.82	0.94	0.69	0.92
KNN	0.73	0.92	0.84	0.95	0.76	0.94
SVC	0.88	0.96	0.80	0.93	0.80	0.93

The results suggest that the TF-IDF vectorization algorithm within the logistic regression classifier has the highest accuracy of 0.96 as well as the highest precision of 0.93. The confusion table for that model can be seen in Table 2. Out of 1,998 comments that were predicted toxic 1,854 are actually toxic. That means, that the model wrongly predicted toxic messages 7% of the time. I use that model to predict whether the messages sent by players in Dota 2 are toxic. The reason GloVe has worse metrics than Word2Vec is probably associated

with the fact that using GloVe I transformed words into a 100-dimensional vector, while in Word2Vec, transformed them into a 300-dimensional one. The performance of TD-IDF and Word2Vec are similar across the classifiers.

Table 2. Confusion matrix for TF-IDF Logistic regression model

	Predicted non-toxic	Predicted toxic
Actual non-toxic	28,712	144
Actual toxic	1,205	1,854

Now that I have a trained model, I can move to the prediction part. In the chat data of Dota 2 games, I have 1,439,488 messages with the information about the match during which the message was sent, the player who sent it, and the time of sending. As mentioned above, for each text message I perform a preprocessing. The preprocessed text is then transformed into a TF-IDF vector based on the document-term matrix created from the Wikipedia comments data. The model gets the vector as an input and returns a value of 1 if the message is toxic and 0 otherwise.

3.2 Cleaning Dota 2 data

Before conducting regressions, it is important to have clean data to work on. There are three steps I want to do to have clean data: remove matches where at least one player left the game (unfinished matches), remove matches where people spoke a language different from English, and remove matches where no one wrote anything during the match in English.

To remove matches where a player left a game is straightforward: player data contains a variable called *leaver_status*, which has a value of 1 if a player left the game and 0 otherwise. It is important to remove those matches because a team that has a leaver, a player who left the game, has a big disadvantage. Statistics show that among 535 games with leavers, a team with a leaver has a 30% probability to win. Whereas, among the rest of the games, the *ex-ante* probability of winning is close to 50%. Therefore, I drop 535 games that had leavers.

As the model of predicting toxicity used English vocabulary from Wikipedia comments, I can only predict the toxicity of texts in English. Therefore, I need to focus on the games which included conversations in English. The matches data has cluster information from which I can retrieve information about the servers on which the match was launched. As the process of detecting a language is computationally heavy, I can only detect a fraction of the games. Therefore, from each game server, I randomly select 100 matches (or less if the server match count is less than 100) and detect the languages used in the text. I use the langdetect library which takes a text as an input and returns language as an output.[19] For each of the game servers, I calculate the prevalent language, which can be seen in Table 3. I keep only the games that were played on servers where English is a prevalent language. I drop extra 3,883 matches in this cleaning step. Nevertheless, there are still games where players texted in languages other than English. That's why I need to perform one more step of cleaning data.

Table 3. Game servers and prevalent language

Game server	Match count	Language
AUSTRALIA	2483	English
AUSTRIA	2242	English
BRAZIL	1816	Portuguese
CHILE	227	Spanish
DUBAI	18	Mixed
EUROPE	18222	English
JAPAN	87	English
PW TELECOM GUANGDONG	65	Chinese
PW TELECOM SHANGHAI	210	Chinese
PW TELECOM ZHEJIANG	62	Chinese
PW UNICOM	27	Chinese
SINGAPORE	7850	English
STOCKHOLM	1683	Russian
US EAST	10748	English
US WEST	3725	English

[19] Danilak, "Language Detection Library Ported from Google's Language-Detection."

The last step of cleaning data is to remove all matches where either players did not text anything or players did not text in English. As I have not only the raw texts sent but also a preprocessed version where all non-Latin characters are removed, I can drop the matches which had no preprocessed text sent. By performing data cleaning, I dropped 535 matches because of positive leaver status, 3,883 matches because of server location, and 9,552 matches because of the absence of English texts.

3.3 Running a regression

Overall, I have 36,030 matches to analyze or 72,060 observations, because two teams play in each match. To have all the relevant variables I perform data transformations between three sources, match data, player data, and text data. The dependent variable is performance or the outcome of the match. The match data contains a variable *win* which is equal to 1 when a team won and 0 otherwise. The variable of interest is toxicity. I have two metrics of toxicity – the number of toxic players in a team and the average toxicity rate of the team. The first one, *# of toxic*, is the number of toxic players in the game. A player is considered toxic if he/she sent at least one toxic message to the chat in the first half of the match. For most of the variables, I consider the information from the first half of the match only. The reason is that closer to the end of the game, players already have an understanding regarding their chances of winning. I might face a reverse causality issue if I consider information from the whole duration of the game, as I would expect a losing team to be more toxic due to resentment of defeat. In fact, Martens (2015) found that the losing team use more toxicity at the late stages of the match.[20] The second metric of toxicity, *toxicity rate*, is the toxicity rate averaged for a team. The toxicity rate is calculated by dividing the number of toxic messages sent by the total number of

[20] Märtens et al., "Toxicity Detection in Multiplayer Online Games."

messages sent in the first half of a match. If a player did not text anything, the toxicity rate is set to zero. To demonstrate, if a player had five toxic messages out of a total of ten messages sent during the first half of a match, the toxicity rate is 0.5. If the other teammates did not text anything toxic, their toxicity rate is 0. Thus, the *toxicity rate* for a team is $\frac{0.5+0+0+0+0}{5} = 0.1$. I also include a variable called *# no text*, which is equal to the number of players who did not text during the first half of a match. The reason for including is that there is a difference between a player who did not text anything and a player who did text but was not toxic.

The other variables that I include as controls are either related to in-game attributes or to players' attributes. In-game attributes include *gold ratio, experience (xp) ratio, last-hits (lh) ratio,* and *kill-death difference*. Those four variables indicate how much a team is powerful compared to the opponent team. I want to control for the in-game performance of the team to be sure that toxicity is not affected by the fact that the team is weak or strong compared to the opponent team. As with toxicity variables, I consider the information only from the first half of the match to avoid the issue of reverse causality. In the game, each player selects a hero, a powerful character, whom a player needs to improve with either stats or items. Items are bought for gold, the game currency which can be earned both passively over time and from killing other heroes, buildings, or non-player characters known as creeps. The upgrade of abilities and stats is achieved through gaining experience. Experience is earned when non-friendly creeps or heroes die nearby. The more gold and experience heroes have, the more powerful they are, which makes a team more capable of destroying the ancient and winning a game. Also, each player has a last-hit count, a count of creeps a hero kills. If a hero kills a creep, he receives more gold and experience compared to a hero who stayed nearby. Therefore, it is important for all heroes to be the last ones who hit a creep and take a gold and experience advantage. However, the sum of gold, experience, or last-hits a team has is not a good predictor of whether a team is winning. I need to know how powerful a team is relative to the opponent team.

Therefore, I can pick either difference in team gold (experience, or last-hits) or a ratio of the gold one team has to the gold the other team has. The ratio better represents the relative performance, especially when comparing different games. Suppose I compare two games with durations of 20 minutes and 60 minutes. The difference in team gold in the first game is more likely to be smaller than the difference in the second because teams earn more gold with time. In the meantime, the ratio takes into account the time difference. The player data contains information about the gold, experience, and last hits for every minute of the match. To get the values for the half time, I take a match duration time, divide it in half, and round to the closest minute to get a round half-match time. Then I take values for each player from the corresponding minute and sum them. Finally, I calculate the ratios: *gold ratio* – the total gold team earned divided by the total gold the opposing team earned; *xp ratio* – the total experience team earned divided by the total experience the opposing team earned; and *lh ratio* – the total number of last hits (killing non-player characters) divided by the total number of last hits by the enemy team. The last in-game variable, *kill death diff* is the difference between the number of kills one team made and the number of kills the opponent team made. The greater the difference is, the more a team dominates a match, the more likely it will win. I do not take ratio here because there are games with zero kills from a team during the first half of a match.

Apart from in-game performance, it is important to control for gamers' skills. Players' attributes variables include *skill mean* and *skill st.d*. In the game, each player has a matchmaking rating (MMR) which represents a skill level. Unfortunately, the information about MMR is not publicly available. Therefore, for each gamer, a TrueSkill rating is calculated based on about 900,000 matches that occurred prior to the time 50,000 matches data was parsed. TrueSkill is a Bayesian skill rating system that can be viewed as a generalization of the

Elo system used in chess.[21] TrueSkill ranking system skill is characterized by two numbers: the average skill of the gamer (mean) and the degree of uncertainty in the gamer's skill (standard deviation). The new gamer starts with a mean of 25 and a st.d. of $25/3$; whereas the mean increases after a win and decreases after a loss. After getting the skill level and standard deviation for each gamer, for each match, I calculate the average skill and average uncertainty of a team. In some matches, I have gamers whose history was unavailable. Therefore, those gamers were given a rating of a new gamer with a mean of 25 and a st.d. of $25/3$. I also create a variable *# no rank* which is equal to the number of players in a team whose history of games was unavailable. The reason for adding the skill is to remove the bias from the coefficient of toxicity. After adding controls for skill, I expect the coefficient of toxicity to increase because I assume that players that have more skill tend to be less toxic. Moreover, I expect the coefficients of skill to be insignificant because of the matchmaking mechanics in the game: players are pooled in teams in a way that the average skill levels are equal. That means that for every skill level, the probability of winning should stay the same.

To know the effect of toxicity on performance I use the following linear regression:

$$win = \alpha + \beta Z + \delta_1 toxicity + \epsilon,$$

where Z are the control variables. I have three models, the first one doesn't include control variables, the second one includes in-game characteristics, and the third one includes in-game characteristics as well as players' skill levels. Thus, I run three regressions for each toxicity measure (*# of toxic* and average *toxicity rate*):

$$win = \alpha + \delta_1 toxicity + \epsilon \quad (1)$$

$$win = \alpha + \delta_1 toxicity + \delta_2 gold\ ratio + \delta_3 lh\ ratio + \delta_4 kill\ death\ diff + \epsilon \quad (2)$$

[21] Herbrich, Minka, and Graepel, "TrueSkill™."

$$win = \alpha + \delta_1 toxicity + \delta_2 gold\ ratio + \delta_3 lh\ ratio + \delta_4 kill\ death\ diff \quad (3)$$
$$+ \delta_5 skill\ mean + \delta_6 skill\ st.d. + \delta_7 \#\ no\ rank + \epsilon$$

According to the correlation table below, toxicity is weakly correlated with the outcome of the match, although the sign is as expected negative. In-game characteristics are strongly positively correlated to the win of the match. As the *gold ratio* and *xp ratio* are highly correlated (0.89), I decided to keep only the *gold ratio* in the regression to avoid a problem of multicollinearity. I expect the coefficients to be positive and statistically significant. Player performances, both *skill mean* and *skill st.d.*, are weakly correlated with the win of the match. I expect the coefficients to be small in magnitude and not statistically significant.

Table 4. Correlation table of main variables

	win	# toxic	toxic rate	gold ratio	xp ratio	lh ratio	kill death diff	skill mean
# toxic	-0.01	1.00						
toxic rate	-0.01	0.81	1.00					
gold ratio	0.49	-0.01	-0.009	1.00				
xp ratio	0.48	-0.01	-0.008	0.89	1.00			
lh ratio	0.31	0.01	0.01	0.66	0.61	1.00		
kill death diff	0.26	-0.01	-0.003	0.51	0.57	0.22	1.00	
skill mean	0.02	0.04	0.03	0.02	0.02	0.02	0.01	1.00
skill std	-0.04	-0.08	-0.07	-0.04	-0.04	-0.03	-0.02	-0.36

4. Results and discussions

I start this section with the description of the variables, followed by explaining the results of the main estimations and concluding with limitations. Table 5 shows the characteristics of the main variables between the two teams: radiant and dire. Each team has 36,030 observations and it can be seen that the teams' characteristics are not different. The number of toxic players in each team is 0.27 on average, meaning that a player could bump into a toxic player every third game, which is consistent with the survey results mentioned in Ingersoll's report (2019).[22] The survey result showed that 38% of people who play Dota 2 regularly experience toxicity. The number of people who did not text to the chat is 3.5 on average. This number is quite high and the possible reason for that is the preference of players to communicate via voice chat. Nordlander (2018) found that 90% of players prefer voice chat, mainly because it is an easier and faster way of communication.[23] Nevertheless, this does not pose a problem for answering the main question. *gold ratio, xp ratio, lh ratio*, and *kill death diff* values are not far distributed. That means that during the first half of the match teams' characteristics are similar on average. The average skill value of a team is equal to the average value of skill in the TrueSkill system, which is 25. That is a good sign that the *skill mean* variable is calculated correctly. The average number of players whose rank was not retrievable is 2.15 on average, meaning that there are 3 players on average whose rating was available.

[22] Ingersoll, "Free to Play?"

[23] Nordlander, *The Different Emotional Effects of Voice and Text Communication in a Game Environment*.

Table 5. Mean and standard deviation comparison between two teams

variable	Team radiant		Team dire		difference in means
	mean	std	mean	std	
win	0.51	0.50	0.49	0.50	0.02
# toxic	0.27	0.55	0.28	0.56	-0.01
toxic rate	0.02	0.06	0.02	0.06	0.00
# no text	3.51	1.15	3.49	1.15	0.03
gold ratio	1.02	0.20	1.01	0.20	0.01
xp ratio	1.02	0.19	1.02	0.19	0.00
lh ratio	1.04	0.30	1.04	0.29	0.00
kill death diff	-0.11	4.62	0.11	4.62	-0.22
skill mean	25.48	1.38	25.48	1.39	0.00
skill std	6.65	1.37	6.62	1.38	0.03
# no rank	2.15	1.51	2.13	1.53	0.02

Table 6 provides an OLS regression estimation of the number of toxic players on the performance of the game, i.e. whether a team won the match. In the first specification, I included only the number of toxic players in a team and a number of players who did not text as independent variables. The variable of interest, the number of toxic players, has a coefficient both statistically and economically significant. On average, having an extra toxic player in a team reduces the probability of winning by 2%. That is, having five toxic players in a team reduces the chances to win a match by 10%. Similarly, on average, having an extra player who does not communicate decreases the probability of winning by 1.3%. The coefficient is also statistically significant at the 1% level. In the second model, I add in-game characteristics. After controlling for the *gold ratio, last-hits ratio,* and *kill-death difference*, the coefficient of *# toxic* decreased in half. The sharp decrease is associated with the negative correlation of in-game attributes and *# toxic*. Therefore, the decrease is associated with the positive bias a variable had. All three control variables are significant at the 1% level. On average, if the gold ratio is increased by 0.1, the probability of winning a match increases by 12.5%. Suppose two teams have 20,000 gold each and then one team receives an extra 2000. Then the gold ratio

would change from 1 to 1.1 increasing the chance of winning by 12.5%. The coefficient size and sign are as expected. However, the sign of the *last-hits ratio* coefficient is negative.

Table 6. The effect of the number of toxic players on performance

	Dependent variable: win		
	(1)	(2)	(3)
# toxic	-0.020***	-0.009***	-0.010***
	(0.004)	(0.003)	(0.003)
# no text	-0.013***	-0.006***	-0.005***
	(0.002)	(0.002)	(0.002)
gold ratio		1.250***	1.248***
		(0.013)	(0.013)
last-hits ratio		-0.026***	-0.026***
		(0.007)	(0.007)
kill-death diff		0.002***	0.002***
		(0.000)	(0.000)
skill mean			0.001
			(0.001)
skill st.d.			-0.004*
			(0.002)
# no rank			-0.003
			(0.002)
Observations	72,060	72,060	72,060
R^2	0.001	0.241	0.242
Adjusted R^2	0.001	0.241	0.241
Residual Std. Error	0.500	0.436	0.435
F Statistic	28.073***	4580.232***	2868.138***
Note:		*p<0.1; **p<0.05; ***p<0.01	

The explanation for this is that controlling for gold if the team focuses on killing creeps instead of killing heroes or buildings, it tends to lose. The last variable, the *kill-death difference* has a coefficient of 0.002 which means if the team has five more kills relative to the opponent team, the chances of winning a game increase by 1%. The third model is similar to the second one apart from adding the controls for players' characteristics. As was explained in a previous subsection, the coefficient of *# toxic* should increase. In fact, it was increased in magnitude by 0.001. The interpretation is that a team with an extra toxic player instead of non-toxic decreases

the probability of winning a match by 1%. The coefficients of player characteristics are insignificant even at 5% which is also expected. A team with higher skills has the same probability of winning as the team with lower skill levels, because of the game matchmaking mechanism.

Table 7 provides an OLS regression estimation of the average team toxicity rate on the performance of the game, i.e. whether a team won the match. Three model specifications are identical to the one in table 6 except the variable of interest is replaced with the average team toxicity rate instead of the number of toxic players in a team. In the first model, the coefficient of toxicity rate is negative and statistically significant at the 1% level. The coefficient means that if we compare two teams with average toxicity rates of 0% and 20%, the team that sends 20% of toxic messages has 2% fewer chances of winning a game. The second variable *# no text* has the same magnitude as in the previous regression with another toxicity definition. In the second model, the coefficient of toxicity rate decreased by half after controlling for the in-game characteristics. However, the coefficient became statistically insignificant even at the 10% level. The reason for that is the small magnitude of the coefficient relative to the magnitude of standard errors. All the other coefficients have a similar magnitude and all are significant at the 1% level. The third model is similar to the second one apart from adding the controls for players' characteristics. As with the previous regression, the coefficients related to players' skills are not significant at the 5% level. However, the coefficient of toxicity rate increased in size and became statistically significant at the 10% level. However, the size of a coefficient is not economically significant. If a team has all players that only send toxic messages to chats, the probability of winning is decreasing by 5%. The possible reason for the low magnitude of the coefficient might lie in the players' habit of texting. In online game chats, people usually text a lot of short messages instead of long ones. The chat data suggests that the length of a message sent is around 2.5. As I analyzed the messages separately, the toxicity rate

is very low as the denominator, the total number of messages sent is big. As Table 5 shows, the actual mean of average team toxicity rate is 0.02, which affects the magnitude of the toxicity rate coefficient.

Table 7. The effect of average toxicity rate players on performance

	Dependent variable: win		
	(1)	(2)	(3)
toxic rate	-0.097***	-0.043	-0.050*
	(0.034)	(0.030)	(0.030)
# no text	-0.010***	-0.004***	-0.004***
	(0.002)	(0.001)	(0.001)
gold ratio		1.251***	1.249***
		(0.013)	(0.013)
last-hits ratio		-0.026***	-0.026***
		(0.007)	(0.007)
kill-death diff		0.002***	0.002***
		(0.000)	(0.000)
skill mean			0.001
			(0.001)
skill st.d.			-0.003*
			(0.002)
# no rank			-0.003
			(0.002)
Observations	72,060	72,060	72,060
R^2	0.000	0.241	0.241
Adjusted R^2	0.000	0.241	0.241
Residual Std. Error	0.500	0.436	0.435
F Statistic	17.634***	4578.771***	2867.078***
Note:		*p<0.1; **p<0.05; ***p<0.01	

Overall, I can conclude that the toxicity negatively affects the performance of a team. After controlling for players' skill and in-game characteristics, a team that has one toxic person more decreases the chance of winning a game by 1%.

Nevertheless, my work contains several limitations. Firstly, I consider only text communication and lack the data for other types of communication like voice chat or pings.

Although, there is a high probability that the text chat and other types of communication overlap in terms of toxicity levels, having more data would give a more accurate effect of toxicity. The other limitation is that I analyze only the English language in communication. I removed all texts written with non-Latin characters. Although I considered only matches played on servers with English as a prevalent language, there were still messages written in the other languages. A better solution would be to detect the language of each message sent and consider matches where messages were sent in English. However, this solution is computationally expensive as there are about 1,5 mln messages sent.

Conclusion

In this paper, I have estimated the effect of sentiments in communication on team performance using a dataset of Dota 2 matches. The OLS regression has shown a statistically significant negative correlation between the number of toxic players in a team and the performance of a team. The effect is also economically significant. As I have drawn parallels between a Dota 2 team and a job team, I propose that being toxic in a job team also negatively affects the team's performance. A job team could be a team of management consultants, software engineers, or researchers.

This study can be extended in a couple of ways. First, getting rid of limitations would improve the results. That is including other types of communication channels and excluding matches where players communicated in different languages. Nevertheless, I believe that getting rid of limitations would not change the sign of the coefficient, but could provide a more accurate estimate of the coefficient. The work can also be extended by splitting the definition of toxicity into sub-categories. For example, toxic texts could be split into subcategories by the writers' mood. The effect could differ if the person sends a toxic joke or an angry text. There are many other ways to look at toxicity. The analysis could focus on the effect of threat, insult, or identity hate on team performance.

References

Alinor, Malissa. “Research: The Real-Time Impact of Microaggressions.” *Harvard Business Review*, May 17, 2022. https://hbr.org/2022/05/research-the-real-time-impact-of-microaggressions.

Bartik, Alexander W., Zoe B. Cullen, Edward L. Glaeser, Michael Luca, and Christopher T. Stanton. “What Jobs Are Being Done at Home During the Covid-19 Crisis? Evidence from Firm-Level Surveys.” Working Paper. Working Paper Series. National Bureau of Economic Research, June 2020. https://doi.org/10.3386/w27422.

Bird, Steven, Ewan Klein, and Edward Loper. “Natural Language Processing with Python,” n.d. https://github.com/nltk/nltk.

Bizarro, Andrea M. “The Distinct Roles of First Impressions and Physiological Compliance in Establishing Effective Teamwork,” n.d., 139.

Bostan, Barbaros. “Player Motivations: A Psychological Perspective.” *Computers in Entertainment* 7 (January 1, 2009).

Conversation AI. “Toxic Comment Classification Challenge.” Accessed June 10, 2021. https://kaggle.com/c/jigsaw-toxic-comment-classification-challenge.

Danilak, Michal Mimino. “Language Detection Library Ported from Google’s Language-Detection.,” n.d. https://github.com/Mimino666/langdetect.

“Dota 2 Matches.” Accessed June 9, 2021. https://kaggle.com/devinanzelmo/dota-2-matches.

Herbrich, Ralf, Tom Minka, and Thore Graepel. “TrueSkill™: A Bayesian Skill Rating System.” In *Advances in Neural Information Processing Systems*, Vol. 19. MIT Press, 2006. https://proceedings.neurips.cc/paper/2006/hash/f44ee263952e65b3610b8ba51229d1f9-Abstract.html.

Honnibal, M, I Montani, S Van Landeghem, and A Boyd. “SpaCy: Industrial-Strength Natural Language Processing in Python,” 2020. https://doi.org/10.5281/zenodo.1212303.

Ingersoll, Christina. “Free to Play? Hate, Harassment, and Positive Social Experiences in Online Games.” ADL, July 2019. https://www.adl.org/media/13139/download.

Marlow, Shannon L., Christina N. Lacerenza, and Eduardo Salas. “Communication in Virtual Teams: A Conceptual Framework and Research Agenda.” *Human Resource Management Review* 27, no. 4 (December 2017): 575–89. https://doi.org/10.1016/j.hrmr.2016.12.005.

Märtens, Marcus, Siqi Shen, Alexandru Iosup, and Fernando Kuipers. “Toxicity Detection in Multiplayer Online Games,” 7. Zagreb, Croatia, 2015. https://doi.org/10.1109/NetGames.2015.7382991.

Mercer LLC. “US Flexible Working Policies & Practices Survey,” 2021. https://www.imercer.com/products/flexible-working-policies-practices-survey?WT.ac=US-20210315-WHN-Flexpp.

Monge, C. K., and T. C. O’Brien. “Effects of Individual Toxic Behavior on Team Performance in *League of Legends*.” *Media Psychology* 25, no. 1 (January 2, 2022): 82–105. https://doi.org/10.1080/15213269.2020.1868322.

Nordlander, Emil. *The Different Emotional Effects of Voice and Text Communication in a Game Environment*, 2018. http://urn.kb.se/resolve?urn=urn:nbn:se:bth-17225.

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. “GloVe: Global Vectors for Word Representation,” n.d. https://nlp.stanford.edu/projects/glove/.

Shockley, Kristen M., Tammy D. Allen, Hope Dodd, and Aashna M. Waiwood. “Remote Worker Communication during COVID-19: The Role of Quantity, Quality, and Supervisor Expectation-Setting.” *Journal of Applied Psychology* 106, no. 10 (October 2021): 1466–82. https://doi.org/10.1037/apl0000970.

Traas, Arjen. “The Impact of Toxic Behavior on Match Outcomes in DotA.” Tilburg University, 2017. https://arno.uvt.nl/show.cgi?fid=145375#:~:text=We%20identified%20predictive%20variables%20of,chances%20of%20winning%20the%20game.