Title	Modeling Argumentation Framework for Twitter Private Opinion Accounts
Author(s)	Zhong, Jiaqi
Citation	
Issue Date	2022-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/17637
Rights	
Description	Supervisor:東条　敏, 先端科学技術研究科, 修士(情報科学)

Master's Thesis

MODELING ARGUMENTATION FRAMEWORK FOR TWITTER PRIVATE OPINION ACCOUNTS

2010127 Zhong Jiaqi

Supervisor	Satoshi Tojo
Main Examiner	Nguyen Le Minh
Examiners	Kiyoaki Shirai
	Shinobu Hasegawa

Graduate School of Advanced Science and Technology
Japan Advanced Institute of Science and Technology
(Information Science)

February, 2022

Abstract

Nowadays, it is convenient for users of social medias to express their thoughts and opinions. Among many tools, Twitter is one of the most widely used and has open APIs which allow developers to extract large amounts of data. However, when users want to obtain some information to make decisions, there is no way to distinguish the reliability of what other users post.

In Japanese, there is a special expression - private opinion (本音) and public sound (建前). Private opinion means one's real opinion, public sound means one's public stance. Private opinion is a method that does not make the other person uncomfortable by returning it in bland words without saying something that is unpleasant in the heart, and it is regarded as common sense in Japanese society. It's hard to judge public sound as the lie, but it's still a persistent treatment for Japanese people whose virtue is to read between the lines. In order to really understand the real thoughts of Japanese, we can focus on accounts that only post private opinions. These special accounts do exist among Japanese netizens, which are called private opinion accounts (本音垢). At the same time, there are also accounts posting vicious words which are more straightforward than the private opinion accounts, which called the venom accounts (毒舌垢). Obviously, we can judge that private opinion has a higher degree of reliability than public sound, therefore we choose to crawl and analyze the private opinion accounts and the venom accounts in the research. Our research can be viewed as a development of XAI and reliable AI, which is in very high demand in recent years.

Our objective is to identify private opinion structure in Twitter data for modeling argumentation graphs with attack relations. Our research based on the Argumentation Framework (AF), which is introduced by Dung, are pairs consisting of a set $\mathcal{AR}$ of arguments and a binary relation between arguments, representing attacks. Formally, an AF is any $\langle \mathcal{AR}, attacks \rangle$ where $attacks \subseteq \mathcal{AR} \times \mathcal{AR}$. Bench-Capon defines the valued-based argumentation framework (VAF) by attaching to each argument the social values that it promotes. Formally, a VAF is a 5-tuple $\langle \mathcal{AR}, attacks, R, val, valpref \rangle$, where R is a non-empty set of values, val is a function which maps from elements of $\mathcal{AR}$ to elements of R, and $valpref$ is a preference relation on $R \times R$.

In our research, we extend the notion of VAF to introduce a Weighted Annotated Discussion Graph (WADisG). Let Γ be a non-empty set of tweets and $G = \langle V, E, A, R, W \rangle$ be a WADisG; for every tweet in Γ there is a node in V and if tweet a attacks tweet b, there is a directed edge (a, b) in E, A is an annotation function for edges $A : E \rightarrow S$, where S represents

attack relations for any directed edge (a,b), and the value's range of S is {attacks,none}. We defined the valued-based argumentation frameworks for as $F = \langle V, attacks, R, W, Valpref \rangle$, where the weighting function for arguments is $W : V \rightarrow R$ for tweets, and the preference relation $Valpref \subseteq R \times R$ is the ordering relation over R. We give weight to arrows rather than nodes because our data set is special compared with typical twitter conversation analysis. In a typical twitter conversation, a tweet usually has many replies from different users, which is suitable for the traditional VAF that gives weight to different nodes (arguments). In our dataset, users rarely receive a reply to a tweet, but they usually quote the arguments of others in a tweet before giving a counter-argument. Therefore, we divide such tweets into sub tweets and annotate the sub tweets with attack relation. The weight of the same user on a topic is calculated by the weighting function therefore the node weights at both ends of the attack are the same. We give weights to different arrows to represent the overall reliability of the tweet. In the proposed WADisG, the grounded extension $S \subseteq T$ of F is the accepted set of tweets based on the weighting scheme W and we refer to it as the solution of G, i.e. the set of tweets with high reliability.

We crawled the data from Twitter API and have done some processing work such as format conversion, solving the garbled code problem, cleaning our data, and annotate for the tweets. After we finish processing the tweets, we carry out some basic morphological analysis work, such as word segmentation to get word frequency. Then, we combine TF-IDF and Japanese grammar (refer to Japanese papers) to select the most useful features for training our model. We carried out two binary classifications, argument classification and attack classification. Later, we use some commonly used machine learning algorithms on our dataset and evaluate the performance of these algorithms based on the confusion matrix and F Score. We found that Multinomial Naive Bayes performs the best and Passive Aggressive the worst in our classification experiments. In the end, we use answer set programming to calculate the set of reliable tweets and visualized with argumentation graph.

Finally, we analyze the structure of private opinion from morphological analysis and syntactic analysis aspect. We look forward to some future work such as making a more complete information retrieval system, including but not limited to adding a more friendly interactive interface, adding indicators from other users such as Net Promoter Score, using the threshold setting in order to provide users in need with a reliable decision-making reference tool.

Keywords: Argumentation Framework, Value-based Argumentation, Attack Labelling, Private Opinion (本音), Reliable Artificial Intelligence.

Acknowledgement

I would like to express my gratitude to all those who helped me during the writing of this thesis. My deepest gratitude goes foremost to Assistant Professor Teeradaj Racharak, Professor Satoshi Tojo and all lab members. Without their help, my research could not have reached its present form.

Contents

List of Figures

List of Tables

Listings

Chapter 1

Introduction

1.1 Background and Research Questions

1.1.1 Twitter for Academic Research

Nowadays, in the information-oriented society, it is convenient for users of popular social medias to express their thoughts and opinions. Although many specialized and generalized social networks exist, today Twitter is one of the most widely used for sharing and critiquing relevant news, and citizens' reactions to news and events on Twitter are often seen as indicators of social interest in the topic.

However, the critical question is that it is not easy to judge whether they are reliable or not. When we users of social medias want to obtain some information, such as authoritative news, evaluation of a certain product, we may prefer to search for other users' comments, opinions and other information. We have chosen Twitter for our research mainly for the following reasons.

1. Twitter has the characteristic of easily expressing the users' thoughts and opinions because every single tweet has a post limit of 140 characters.

2. Twitter must be expressed in concise sentences which is convenient for later annotation research work.

3. Twitter is easy to crawl according to the topic or crawl according to the user which is applied in this research. Twitter has open, extensive developer APIs that allow developers to interact with Twitter servers and easily extract large amounts of data.

4. Twitter has a large number of users. In 2021, twitter had 192 million active users every day [1].

Among the above reasons, we especially want to emphasize the third point. Twitter provides many convenient APIs for academic research. At a high level, APIs are a conversation method that allows computer programs to request and deliver information to each other. This is possible by allowing software applications known as endpoints: an address that corresponds to a particular type of information provided on Twitter (endpoints are generally unique, like phone numbers). Twitter gives users access to crawl tweets through APIs. When we want to use Twitter's API, we are required to register an application, in our research we use Twitter Developer [2]. By default, the application can only use Twitter's public information. Twitter's API includes a wide range of endpoints, which fall into five main categories: accounts and users, tweets and replies, direct messages, advertisement, publisher tools and SDKs. Later, we will elaborate in more detail. In our research, the main information we obtained includes accounts and users, tweets and replies.

1.1.2 Private Opinion Accounts

In Japanese, there is a special expression - private opinion (本音) and public sound (建前). Private opinion means what one really feels, one's real opinion. Public sound means one's public stance, polite face. It is a method that does not make the other person uncomfortable by returning it in bland words without dare to say something that is unpleasant in the heart, and it is regarded as common sense in Japanese society. It's hard to judge public sound as lies, but it's still a persistent treatment for Japanese people whose virtue is to read between the lines.

Even if the Japanese do not confirm through language, they will judge according to the atmosphere and actions of the people around them. In their dialogue with others, they will expect to "understand what I mean", but have not conveyed their ideas. It is rare in the world that the Japanese are familiar with the "read between the lines". However, in European and American countries where many nationalities and religions are mixed together, it is natural for people with different cultures and languages in regions, schools and workplaces to live, and it is very difficult to live by the "read between the lines" with only one culture, they are more used to expressing their ideas directly.

Japanese society is called "village society", and there are many unique local rules of the communities to which they belong. This is actually an act

of observing the atmosphere and conditions and aligning words and deeds with local rules. Take an example of "read between the lines" in Japanese communities, when you see something, even if you think its answer is A, you won't answer immediately, but observe what people around you think. As a result, if you know that the people around you are B, don't adhere to the opinion of A, but turn your opinion into B.

In other words, in order to really understand the real thoughts of Japanese, we can focus on accounts that only post private opinions. These special accounts do exist among Japanese netizens, which called private opinion accounts (本音垢). At the same time, there are also accounts posting vicious words which are more straightforward than the private opinion accounts, which called the venom accounts (毒舌垢). Obviously, we can judge that private opinion has a higher degree of reliability than public sound, therefore we choose to crawl and analyze the private opinion accounts and the venom accounts in the research.

1.1.3 Need in Reliable AI and XAI

Today, we use applications loaded with artificial intelligence as natural as breathing. Many of these applications use machine learning algorithms. We can rely on their smart recommendation to listen to songs and watch videos that match our preferences, sometimes the accuracy of their recommendations even exceeds our imagination. However, most machine learning algorithms are black boxes. We can know the input and output, but it is difficult to figure out the specific internal operating mechanism. This leads us to have questions about the recommendation of the system: how does the AI system calculate the content recommended to me? Undoubtedly, this is an era when users have a demand for Reliable AI and XAI.

When the existing AI (Figure 1.1) users interact with the system, they can only get the output results given by the system, so the users will have many questions such as "Why not something else?", "Why should I trust you?", "How do I correct an error?".

In Figure 1.2, XAI will be equipped with explainable models, and the output results will also provide an interface for explanation. When its users interact with it, users can better understand the reasons for making certain decisions for they understand why not choose something else and know the reasons why errors occur.

Explainable AI (XAI) is a set of tools and frameworks for understanding how machine learning models make decisions. It does not break down every step in the AI model. It is nearly impossible to track the millions of parameters used in deep learning algorithms. Instead, XAI provides ana-

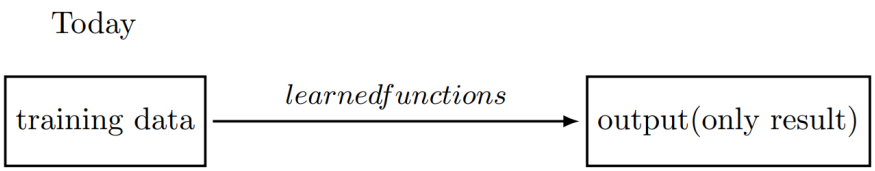


Figure 1.1: Today's AI

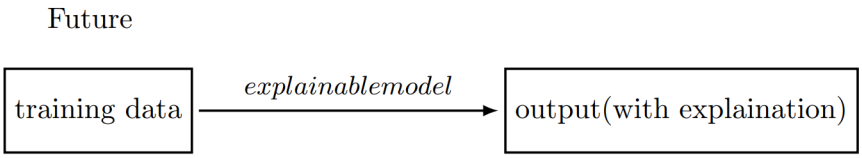


Figure 1.2: Future's XAI

lytical information about how the model works, which experts can use to understand the logic leading up to the decision.

Proper application of XAI can provide three major benefits:

1. Increased confidence in the ML model
   Confidence in AI-based systems will increase if decision makers and other stakeholders can understand how the ML model led to the final output. With the Explainable AI tool, we can find a clear and easy-to-understand explanation of how the model derived its output. For example, when analyzing a medical image such as X-rays using a deep learning model, an extensible map (heat map) can be generated using an Explainable AI to highlight the location in the image used for diagnosis. An ML model that classifies fractures could highlight points in the image that were used to determine a patient's fracture.

2. General troubleshooting improvements
   Once the AI is accountable, we can debug our model and troubleshoot whether it is working properly. For example, assume that we have a model that can identify the animals in an image and find that this model continues to classify dogs playing in the snow as foxes. We can

easily identify why these errors are occurring continuously by using the Explainable AI tool. If we look at the Explainable AI model we're using to see how the predictions are made, we'll see that the ML model distinguishes between dogs and foxes based on the background in the image. This model interpreted the background of the room as a dog and mistakenly learned that if there was snow in the image, there was a fox there.

3. Remove bias and potential AI bugs
   XAI can also be used to identify the cause of bias. For example, let's say you have a model that can identify a car that makes an illegal left turn. You will notice that this model was biased from the training data when asked to define where in the image the violation was identified. In this model, instead of focusing on the car turning left illegally, the question was whether there was a bug in it. Such an effect may be caused by a realistic data set that contains a large amount of images taken on undeveloped roads, or a realistic bias that can be easily truncated in areas with poor funding in the city.

Some people are reluctant to do ML projects because they don't understand what ML is doing. We wouldn't be able to give full control to a mysterious machine learning model, especially when it involves important decisions. AI systems make forecasts that have a huge impact. In industries such as medical and self-driving cars, this prediction can make a difference between human life and death.

Especially without an explanation of how the decision was made, it would be difficult to be confident that the model's decision would be credible, much less make the model better than humans. As users, we want to know how the AI model make predictions and decisions and how can we be sure that there is no bias in the algorithm. Meanwhile, we expect enough transparency and interpretability to trust the model's decisions.

Decision makers want to understand the rationale behind AI-based decisions and be confident that the decisions are appropriate. In fact, according to a PwC study [3], most (82%) CEOs need an explanation to trust their AI decisions.

Answers to the question of explainability is related to the responsibility, validity (e.g., robustness), privacy-preserving and more broadly trust of AI systems (Figure 1.3) which are reliable will be intrinsically connected to the adoption of AI in industry at scale [4]. Explanation in AI aims to create a suite of techniques that produce more explainable models, while maintaining a high level of searching, learning, planning, reasoning performance: optimization, accuracy, precision; and enable human users to understand,

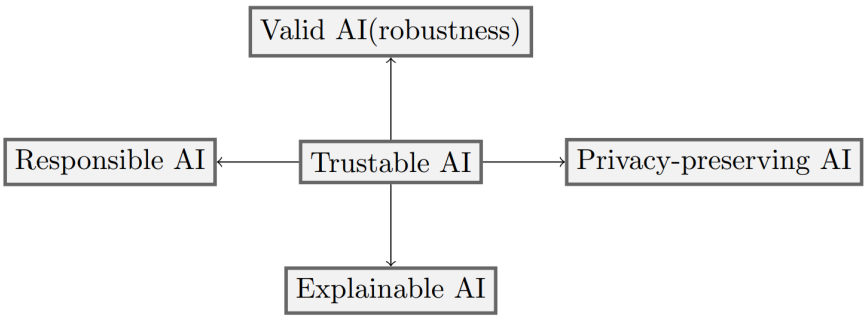


Figure 1.3: Combination of Existing AI towards Trustable AI.

appropriately trust, and effectively manage the emerging generation of AI systems.

1.2 Objectives and Contributions

We aim to clarify the structure of the private opinion in Japanese. The thorough objectives are as follows.

1. To study machine learning techniques that are appropriate for argument mining tasks in Value-Based Argumentation Frameworks [10].

2. To identify private opinion structure in Twitter data for modeling argumentation graphs with attack relations.

Previously, Cocarascu and Toni [17] have shown that using Argumentation Frameworks [7] can improve the performance of machine learning techniques in detecting deceptive reviews. Relatedly, in our research, to a certain extent, tweets can also be regarded as reviews, and whether their reliability are high or low matters because the social consequences of the butterfly effect (such as defamation) can also be triggered for users who see these tweets.

In the previous research by Mihai Dusmanu et al. [5], when performing classification tasks on argument detection namely to classify texts into non-argumentative texts and arguments, the classification is often wrong when tweets contain irony. As a result, the value of precision and recall are not so good. We consider when an argument is private opinion, its reliability is higher than when it is public sound. Considering the influence of private

opinion and public sound, the performance of argument mining is expected to improve. There exist plenty of private opinion accounts in Twitter and some hashtags with venoms are more likely to be private opinion for when people behave maliciously, it's basically their true thoughts. Analyzing this kind of accounts and hashtags is possibly to identify private opinion and public sound in Tweet data for modeling argumentation graphs.

For societal benefits, if this research is achieved, making decisions in purchasing or marketing will be easier than before because argument mining will provide users with suggestions with high reliability. On the other hand, we can always see vicious incidents caused by defamation, including suicide. The victims of these incidents include celebrities and ordinary people. If we can figure out the structure of private opinions and then distinguish highly reliable arguments and statements, these tragedies may also be avoided.

1.3 Thesis Structures

In this thesis, Chapter 2 lists the related works such as Argumentation Frameworks [7], Valued-Based Argumentation Frameworks [10] and some grammar knowledge in Japanese. Chapter 3 describes the process of modeling Twitter Data using Valued-Based Argumentation Frameworks including data crawling, data processing, and the final application. In Chapter 4 the classification experimentation is conducted. And in Chapter 5 the evaluation is given. Finally we draw the conclusion in Chapter 6.

Chapter 2

Preliminaries

2.1 Overview of Argumentation

Argumentation is a "verbal", "social", and "rational" activity aimed at convincing a reasonable critic of the acceptability of a standpoint by putting forward a constellation of propositions justifying or refuting the proposition expressed in the standpoint [6].

Verbal means people can and do gesture and frown at each other, and occasionally this might be away of resolving some disagreement. In our conception here, argumentation is an inherently linguistic activity—either in spoken or written mode. This is to be distinguished from, for instance, a fistfight, which can be a different means of sorting out a conflict.

Social means argumentation is a matter of interaction among a number of people, with a minimum of two. Granted, many of us munch on a difficult decision for some time by mentally walking through the consequences of the alternatives, but genuine arguing requires a person to argue with.

Rational means argumentation targets specifically the dimension of reason. When one person reminds the other "Be reasonable!", the point is to call for a style of dialogue that is not driven by emotional outbreak, power struggle, personal offense, etc., but by the sober exchange of—reasonable arguments.

In fact, argumentation intersects with philosophy and logic (especially in computational argumentation field). For philosophy, we will show an example, and for logic, we will elaborate in 2.3 and later parts.

In philosophy, in the terminology of Aristotle, 3 central factors determining the success of an argumentative exchange are labeled as follows.

1. Logos: Speakers employ rules of sound reasoning.

2. Ethos: Speakers signal their authority or credibility (or that of their source) which is analogous with expert system.

3. Pathos: Speakers seek to communicate their standpoint in a manner that seeks to evoke an emotional response.

Here we give a simple example about a building needs to be demolished and show the arguments from the above three dimensions.

Logos: That building needs to be demolished, because it is full of asbestos, which is known to be hazardous, and there is no way to stop its diffusion from the different parts of the building.

Ethos: That building needs to be demolished, because it is full of asbestos, as the report by the university engineers has shown.

Pathos: That building needs to be demolished, because it is an irresponsible source of danger to the health and indeed the life of our children who spend so many hours in those poisonous rooms every day!

We can notice that the three dimensions do not usually occur in isolation; most arguments will mix them with each other, to different degrees. But the logos dimension is arguably the most important one, and for argumentation mining the most relevant.

2.2 Overview of Argument Mining

We will briefly introduce argument mining from three aspects.

1. Field: it is a research area within the natural language processing field.

2. Work: in this field we automatically extract human arguments and their relations from a variety of textual corpora.

3. Goal: we aim to provide machine-processable structured data for computational models of arguments and reasoning engines.

As we mentioned above, argument mining is automatic extraction and identification of argumentative structures from natural language text with the aid of computer programs. Argumentation structure is also called as "constellation of propositions". The shape of a justification might be quite simple, for instance a single convincing sentence. But quite often it will be more complex and involve a web of interrelated points that the speaker carefully assembles in a non-arbitrary way. Here is an example of argument mining (Figure 2.1).

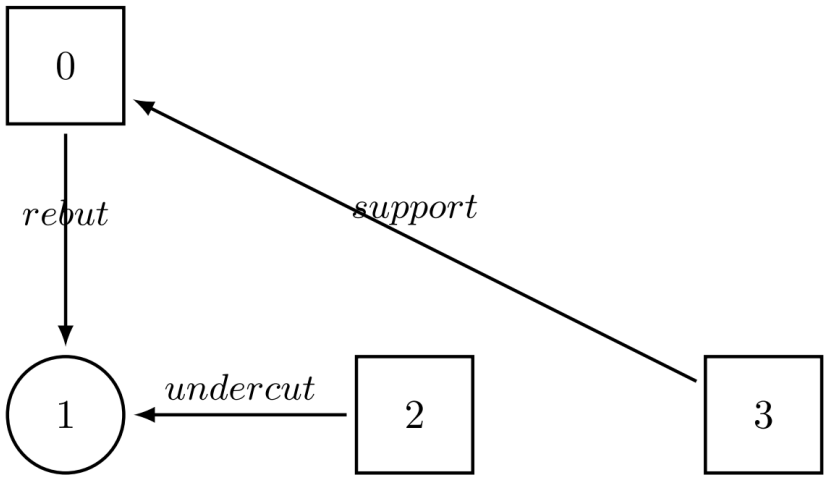


Figure 2.1: An Example of Argument Mining

[0] We really need to turn down that building.
[1] Granted, it will be expensive.
[2] But the degree of asbestos contamination is not tolerable anymore.
[3] Also, it is one of the ugliest buildings in town.

In this example, nodes represent for arguments and arrows represent for attack or support relations. Both of them are conceptions from abstract argumentation frameworks [7] and we will introduce later in more details. Here, node 0 is the central claim of the text. Node 1 places an "attack" of type "rebut", which gives a reason why node 0 should not hold (possible counter-considerations). Node 2 is a counterargument, an "undercut" type of attack on the "rebut" relation, indicating that node 1 is not regarded as a good or sufficient argument against node 0. Node 3 is a "support" relation, provides an argument directly in favor of node 0.

In this example, we treat the arguments as nodes and the relations as arrows. In fact, this is modeling. In Chapter 3, we will perform modeling combined with Twitter data. Before that, we will focus on the modeling of the argument.

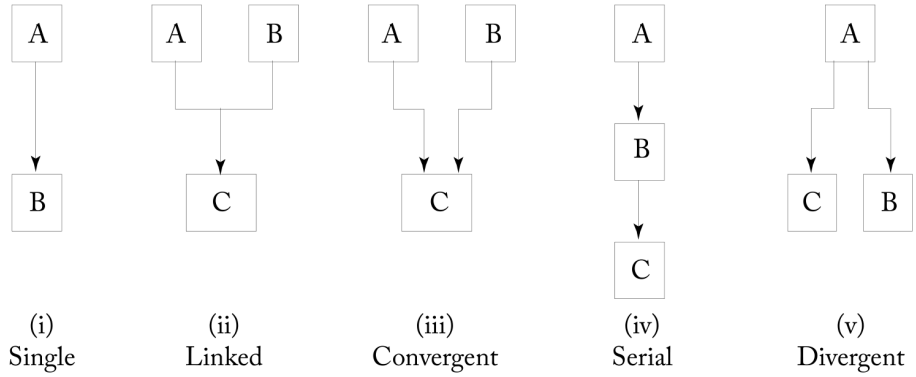


Figure 2.2: Argument Structures

2.2.1 Argument Structure

An argument structure is comprised of a constellation of propositions related to a claim, which is the proposition that the argument seeks to establish. Other propositions are used as premises, which are brought to bear, in support of the conclusion, and sometimes also to attack it. Any claim may play the role of final conclusion of some argument, and any conclusion can be made into a premise supporting further reasoning.

Proposed common structure of arguments [8], including the relation of argument components to one another are shown in Figure 2.2.

The essential distinctions are as follows.

- Linked: two reasons together strengthen each other.
- Convergent: there are multiple distinct premises that each stand alone to support a claim.
- Serial: a chain of reasoning is required where one premise supports an interim conclusion, which is the premise of the next argument (e.g., if p, then q, if q then r).
- Divergent: one premise that supports two or more conclusions.

2.2.2 Types of Argument Models

Argumentation models are classified into rhetorical models, dialogical models, and monological models by Bentahar, Bélanger, and Moulin [9].These types of models are complementary, and each model comes from a different

Table 2.1: Types of Argument Models

Model Type	Argument Evaluation Based On	What is Linked	How They Are Linked	Structure
Monological	Tentative proofs	Premises, claims	Internal inference structure	Micro-structure
Rhetorical	Audience's perception	Whole arguments	Persuasion structure	Rhetorical structure
Dialogical	Defeasible reasoning	Whole arguments	Dialogical structure	Macro-structure

perspective. The monological model treats arguments as tentative proofs and connects premises and claims to describe the microstructure: the internal reasoning structure of each argument. The rhetorical model looks at the argument based on the audience's perception and links the whole argument to a rhetorical structure that emphasizes the persuasive structure of the argument. The dialogical model looks at the argument through the perspective of defeasible reasoning and links the whole argument to a macro structure: the dialogical structure. The distinctions are listed in Table 2.1.

Famous example of monological models is Toulmin model, which was first proposed by British philosopher Stephen Toulmin in 1958. Toulmin Model has six elements: claim (arguments), grounds (facts and evidence), warrant (implicit assumption) and qualifier, rebuttal, backing (support for warrant). An example is followed and visualized in Figure2.3.

The example of rhetorical models we want to introduce here is the New Rhetoric. It takes persuasion and audience as central concerns arguments seek to persuade a given audience, or to convince the universal audience Argumentation is thus a persuasive act involving two or more interlocutors, and it is "relative to the audience to be influenced" "The goal of all argumentation...is to create or increase the adherence of minds to the theses presented for their assent."

And in our research, we apply Dung's argumentation frameworks, which is a representative example of dialogical models.

Existing approaches in natural language processing primarily focus on micro-level (monological) rather than macro-level (dialogical) perspective. The former focuses on internal structure of arguments while the latter focuses on connections between arguments. It is also for this reason that we treat each tweet as an independent argument and analyze whether there is an

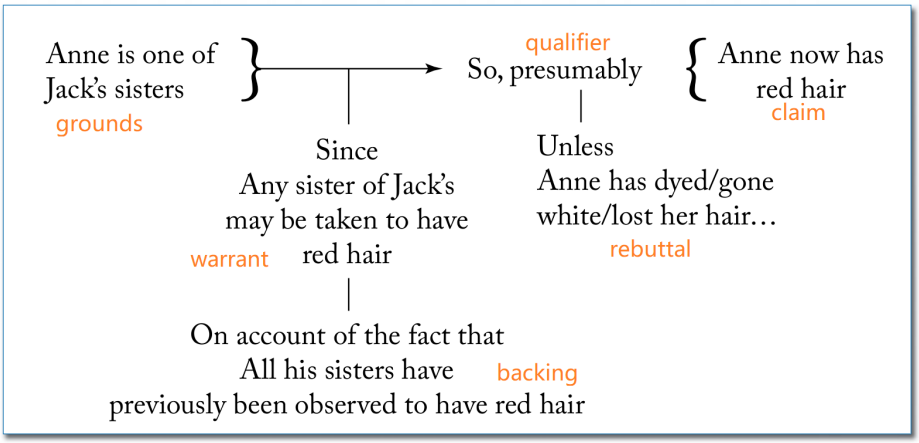


Figure 2.3: Toulmin Model

attack relation between the tweets.

2.3 Abstract Argumentation Frameworks

Argumentation Frameworks (AFs), also call Abstract Argumentation Frameworks, is introduced by Dung [7], are pairs consisting of a set $\mathcal{AR}$ of arguments and a binary relation between arguments, representing attacks. Formally, an AF is any $\langle \mathcal{AR}, attacks \rangle$ where $attacks \subseteq \mathcal{AR} \times \mathcal{AR}$.

Before introducing the example of Argumentation Frameworks, we need to give some relevant definitions.

For two arguments $\mathcal{X}$ and $\mathcal{Y}$, the meaning of **attacks**$(\mathcal{X}, \mathcal{Y})$ is that there exists an attack of $\mathcal{X}$ onto $\mathcal{Y}$.

Let $\mathcal{CS}$ be the complete set, any subset of it including the set $\mathcal{S} \subseteq \mathcal{CS}$.

A set $\mathcal{S}$ of arguments **attacks** an argument $\mathcal{Y}$ if there exists $\mathcal{X} \in \mathcal{S}$ such that $\mathcal{X}$ attacks $\mathcal{Y}$.

A set $\mathcal{S}$ **defends** an argument $\mathcal{X}$ if for any argument $\mathcal{Y}$ which attacks $\mathcal{X}$, $\mathcal{S}$ contains an argument that attacks $\mathcal{Y}$.

A set $\mathcal{S}$ is **conflict-free** if for any $\mathcal{X}$, $\mathcal{Y} \in \mathcal{S}$, there is no $(\mathcal{X}, \mathcal{Y}) \in attacks$.

An argument $\mathcal{X}$ is **acceptable** with respect to the set that defends it, i.e. **acceptability** means finding a subset which is conflict-free and collectively defends itself.

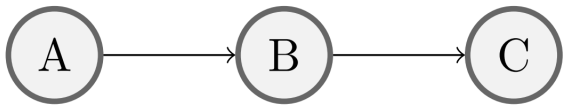


Figure 2.4: An Example of Argumentation Frameworks

In the example shown in Figure 2.4, we want to find the acceptable sets from the relations $\mathcal{A}$ attacks $\mathcal{B}$ and $\mathcal{B}$ attacks $\mathcal{C}$. Combining the above definitions, we can calculate the acceptable sets are: $\varnothing$,$\{\mathcal{A}\}$,$\{\mathcal{B}\}$,$\{\mathcal{C}\}$,$\{\mathcal{A},\mathcal{C}\}$.

From this example, we can find that Argumentation Frameworks can bring the following contributions.

1. encapsulates arguments as nodes in a digraph.
2. connects them through a relation of attack.
3. defines a calculus of opposition for determining what is acceptable.
4. allows a range of different semantics including what we will introduce.

Basic argumentation semantics are listed below:

- A set $\mathcal{S}$ is **admissible** if $\mathcal{S}$ is conflict-free and $\mathcal{S}$ defends all arguments attacking $\mathcal{S}$.
- A set $\mathcal{S}$ is **preferred** if $\mathcal{S}$ is the maximal (w.r.t. set inclusion) admissible.
- A set $\mathcal{S}$ is **complete** if $\mathcal{S}$ is admissible and $\mathcal{S}$ contains all arguments it defends.
- A set $\mathcal{S}$ is **grounded** if $\mathcal{S}$ is the minimal (w.r.t. set inclusion) complete.
- A set $\mathcal{S}$ is **stable** if $\mathcal{S}$ is conflict-free and $\mathcal{S}$ attacks each argument $\mathcal{X} \in \mathcal{CS} \backslash \mathcal{S}$.

2.4 Valued-Based Argumentation Frameworks

Bench-Capon [10] defines valued-based argumentation frameworks by attaching to each argument the social values that it promotes, and making the semantics dependent on a particular preference order over values, representing a particular audience. Formally, a VAF is a 5-tuple:

$$VAF = \langle AR, attacks, V, val, valpref \rangle$$

Where $\mathcal{AR}$, and *attacks* are as for a standard argumentation framework, $\mathcal{V}$ is a non-empty set of values, *val* is a function which maps from elements of $\mathcal{AR}$ to elements of $\mathcal{V}$, and $valpref$ is a preference relation (transitive, irreflexive and asymmetric) on $\mathcal{V} \times \mathcal{V}$. We say that an argument $\mathcal{A}$ relates to value v if accepting $\mathcal{A}$ promotes or defends v: the value in question is given by $val(\mathcal{A})$. For every $\mathcal{A} \in \mathcal{AF}$, $val(\mathcal{A}) \in \mathcal{V}$.

The purpose of extending the AF to VAF was to distinguish between one argument attacking another, and that attack succeeding, so that the attacked argument is defeated. Therefore the notion of **defeat** is defined as follows:

An argument $\mathcal{A} \in \mathcal{AF} defeats$ an argument $\mathcal{B} \in \mathcal{AF}$ if and only if both $attacks(\mathcal{A}, \mathcal{B})$ and not $valpref(val(\mathcal{B}), val(\mathcal{A}))$.

Basic argumentation semantics are also defined in VAF.

An argument $\mathcal{A} \in \mathcal{AR}$ is **acceptable** with respect to set of arguments $\mathcal{S}$, $acceptable(\mathcal{A}, \mathcal{S})$ if:

$(\forall x)((x \in \mathcal{AR} \& defeats(x, \mathcal{A})) \rightarrow (\exists y)((y \in \mathcal{S}) \& defeats(y, x)))$.

An set $\mathcal{S}$ of arguments is **conflict-free** if

$(\forall x)(\forall y)((x \in \mathcal{S} \& y \in \mathcal{S}) \rightarrow (\neg attacks(x, y) \vee valpref(val(y), val(x))))$.

A conflict-free set of arguments $\mathcal{S}$ is **admissible** if

$(\forall x)((x \in \mathcal{S}) \rightarrow acceptable(x, \mathcal{S}))$.

A set of arguments $\mathcal{S}$ in an argumentation framework $\mathcal{AF}$ is a **preferred** extension if it is a maximal (with respect to set inclusion) admissible set of $\mathcal{AR}$.

A conflict-free set of arguments $\mathcal{S}$ is a **stable** extension if and only if $\mathcal{S}$ attacks each argument in $\mathcal{AR}$ which does not belong to $\mathcal{S}$.

2.5 Argument Detection

As we stated in Chapter 1, the goal of this research is to clarify the structure of the private opinion in Japanese, therefore we need to have some basic knowledge about Japanese grammar. In Chapter 3, we will introduce the data set and data analysis in detail. It is important to highlight here that our research will distinguish between argumentative data and non-argumentative data. And for argumentative data, we distinguish between the presence or absence of an attack relation.

2.5.1 Overview of Argument Detection

Argument detection is a subtask of argument mining mentioned in 2.2. Argument detection can be divided into three steps:

1. Understand the context: Has anyone tried to convince you of something?
2. Determine the conclusion: What are they trying to convince you?
3. Find out why: Why do they think you should believe them?

Understand the Context

When we want to judge whether a piece of text is argumentative or not, we usually judge it based on the context. This also applies to tweets. Under normal circumstances, we can understand the background or topic of the argumentation through the context - debate, class, media, political discussion between friends, etc. We can also judge whether this is an argument or not by some words and phrases that mark the argument, such as "argument", "my point of view", "my opinion", "what should you think". In the following scenarios, argumentation often occurs.

Quarrels: unstructured events triggered by emotions can lead to verbal abuse, shouting, and violence; although quarrels always involve contentious activities, it is not uncommon for the lack of arguments to be understood as reasons.

Advertisement: attempting to influence or shape opinions that benefit the product. Advertisers want to sell you something. They want you to conclude, "Well, I'll buy that." The reason is usually shown in the ad-because buying it will make you stronger, sexier, smarter, more popular, and so on. Many of the arguments here are implicit, that is, they are unstated in the advertisement. But every advertisement is almost always an argument.

Debate: it is a highly structured event, victory depends on presenting the reason that most impresses the judge; it is rare to find a debate without arguments, although it is not rare to find a debate have very few good arguments.

Persuasive discussions: usually unstructured conversations in which participants try to persuade each other to accept certain beliefs (for example, discussions about politics, religion, morality, etc.); since these discussions are personalized because of their persuasiveness, such discussions usually do not lack some kind of arguments.

Opinioneering: putting forward opinions, such as in letters to editors, columns, or Op/Ed articles or sermons. They are not an immediate dialogue in nature; these may lack arguments. There are different ways over and over again.

Negotiation: in the context of different interests, driven by the continuous desire to reach an agreement, this is more a compromise than a persuasive discussion; successful negotiations usually involve arguments supporting the initial position, followed by a series of compromises.

Action planning: characterized by the need for action, whether individual or group; This process includes proposing and then evaluating various feasible schemes, and then comparing and evaluating the feasible schemes; this may not be argumentative in nature, for example, when the option is "forced through".

Learning environment: classrooms, offices and administrative lawns provide space for teacher-student interaction; in this environment, teachers often raise arguments during lectures, and often raise arguments during discussions and group work; however, these may be non-arguments Sexual, especially when they involve the simple transmission of information from one brain to another.

Determine the Conclusion

"You are worth buying this product because its quality is above average and its price is below average." This is an argument designed to get you to buy this product. Usually, the main purpose of an argument is to emphasize a point of view. In other words, argument is a tool designed to persuade or force people to believe something. This "thing" is what we call a conclusion. The validity of the argument depends on whether it provides a convincing reason to believe the conclusion, but the first consideration is that if the conclusion of the argument is not determined first, the validity of the argument cannot be evaluated.

In some argumentation, the conclusion is placed in a prominent position

at the beginning or end of the argument, for example, at the beginning or end of the paragraph containing the argument. Other situations are that to repeat the conclusion many times in the argumentation to remind people that it is the most important point. There are some words and phrases whose main purpose is to introduce the conclusion - called "conclusion marker". Conclusion markers include the following: "as a result", "therefore", "thus", "as a consequence", "hence", "so", "in that case", "then", "accordingly", "the bottom line" and "for this reason". But not all of these terms mark a conclusion; for example, "then" usually represents the next event in a series of events.

Find out Why

The reason is another important part of the argument that must be determined. These are claims that support the conclusion - as their name implies, they give you reason to believe it. Without them, there would be no debate - just a claim. Therefore, it is wrong to respond to the argument request with "he will back tomorrow". This may be a conclusion, but without reason, there is no argument.

The most effective way to find a reason is to find the "reason markers". These terms include the following: "since", "because", "reason", "according to", "cause", "considering", "by", "assume", "if", "for", "in fact", "in light of" etc.

2.5.2 Related Japanese Grammar

Similar to English, in Japanese, judging whether the text is argumentative or not can also be based on some markers. And for argumentative data, the detection of attack relation are related to counter-argument because it is used when we want to deny others' arguments.

In Ijuin's work [11], she found that the typical form of ' counter-argument' is "確かに〜。しかし，〜。 " and "確かに〜が，〜。 ". Including the pattern in which a sentence is inserted before "counter-argument", it accounts for 89% of the total.

On the other hand, the use of "確かに" at the end of the opinion and the use of the colloquial expression ("でも" or "けど") at the end of the opinion are often seen in Japanese learners.

She found that native Japanese speakers and Japanese learners had different tendencies when it came to counter-arguments. In any case, we are analyzing data from all users who tweeted their private opinion in Japanese regardless of their nationality, and both cases should be taken into account.

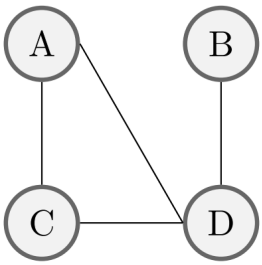


Figure 2.5: An Example of Answer Set Programming

In our follow-up experiments, we will combine Japanese grammar and statistical information for further analysis. Specifically, in 3.2.4, we will refer to other Japanese grammar documents that are referenced in data annotation.

2.6 Answer Set Programming

In 3.3, we will build weighted argumentation frameworks. After that we will use the WAF (weighted argumentation frameworks) as input using answer set programming to get the solution for the set of tweets with high reliability.

ASP (Answer Set Programming) is a declarative programming paradigm based on logical programs and their answer sets, which provides a simple and powerful modeling language to solve combinatorial problems, while Potsdam Answer Set Solving Collection (Potassco) [21] includes a variety of ASP tools such as gringo and clasp are combined to become an integrated ASP system clingo.

A simple ASP program may contain three parts: facts, rules, and output. The facts and rules section is used to describe the problem. The output section is used to view the results.

Here in Figure 2.5 is an example of ASP. The program of this example is as follows.

Listing 2.1: Answer Set Programming Example Sample Code

```
% facts
v(1..4).

e(1, (3;4)).
```

```
e(2, 4).
e(3, (1;4)).
e(4, (1;2;3)).

% rules
c(X, Y) :- v(X), v(Y), not e(X, Y), X != Y.

% Display
#show c/2.
```

The goal of this example is to find points in the graph that are not directly connected by edges.

The facts section describes the situation in Figure 2.5, where v represents a vertex and e represents an edge. Writing "v(1⋯4)" is equivalent to "v(1).v(2).v(3).v(4).", here "." is the terminator of a code. "e(1, (3;4))." is equivalent to "e(1, 3). e(1, 4)", indicating that 1 to 3, 4 have edges. "1...4" and ";" are syntactic sugar. In all, the fact section states that there are 4 points, and some of them have edges and some don't.

The ":-" in the rule section can be understood as if the condition behind it is true, then the front is true. And "," means "and". "not", as the name suggests, means negation. So the condition for c(X, Y) to hold is: there are two points X and Y, there is no edge between them, and X, Y are not the same point.

In this example, c(X, Y) is actually the result. So, c is output on the last line. Here "#show" means output, "c" in "c/2" means c(X, Y), and "2" means c has two elements.

We use clingo to run this ASP program and the results are as follows.

Listing 2.2: Answer Set Programming Example Result

```
Answer: 1
c(2,1) c(1,2) c(3,2) c(2,3)
SATISFIABLE

Models : 1
```

c(2,1) c(1,2) c(3,2) c(2,3) is the solution of this example.

In our later implementation, basic argumentation semantics mentioned in 2.4 are also included.

Chapter 3

Modeling Twitter Data Using AF and VAF

3.1 Tweets Crawling

In our research, we use data crawled from Twitter API to create a data set, which also reflects the originality of our research. During the data crawling process, we use the Twitter Developer API [2] mentioned in 1.1.1. And in 1.1.2, we stated the reason we chose private opinion accounts and venom accounts from Twitter.

To apply for Twitter API (register as a developer), we should first apply for a developer account and write some primary reasons for using Twitter developer tools and briefly state our research plan.

After we finish registering, we can enter the Developer Portal and create our research application.

3.1.1 Obtaining the Bearer Token

There are OAuth 1.0 and OAuth 2.0 [12] for authentication of Twitter API.

OAuth 1.0: we can use the API to perform (Tweet) a specific function on behalf of the user with the consent of the user. We could also get the Tweet of private Twitter account.

OAuth 2.0: Bearer Token authenticates requests on behalf of Twitter developer Application. As this method is specific to the developer Application, it does not involve any users. This method is typically for developers that need read-only access to public information.

Since our research only gets tweets, we will use the method using Bearer Token of OAtuth 2.0. OAuth 2.0 authentication method requires for us to

```
C:¥Users¥かき>curl -X POST -H "Content-Type: application/x-www-form-urlencoded;charset=UTF-8" -H "Author
ization: Basic ZWJWcGlLb2FPaWJOTmlyMVB5ekVYUVVxWjpxNkVCdElnb3plRlI2QTlNSjVuYmpEdFUzcjE2TUN2cldtcHRjclljW
UU5b0NVVmpjaw==" -d "grant_type=client_credentials" https://api.twitter.com/oauth2/token
{"token_type":"bearer","access_token":"AAAAAAAAAAAAAAAAAAAAAFtoNAEAAAAATy6AOuQ3pCV1feS%2FxH4hQPAsTC0%3Du
bFy4JL5rFvqt4JhNRXC2HAG6pSEalCxIb2PZwidnj9T5APtBT"}
```

Figure 3.1: Using POST Method to Obtain the Bearer Token

pass a Bearer Token with our request, which we can generate within the Keys and tokens section of our developer Apps.

Since we have created our research application, we then can generate the API key and API key secret from it. By using the POST method from cURL [13] command to the Consumer API keys(API key and API key secret) just obtained, we can obtain the Bearer Token. Specifically, we need to use the Base64 [14] encoder to encode the API key and API key secret, i.e., using ":" to connect and the API key and API key secret then encode it.

In our case, the API key is "ebVpiKoaOibNNir1PyzEXQUqZ", and the API key secret is "q6EBtIgozeFR6A9MJ5nbjDtU3r16MCvrWmptcrYcYE9oCUVjck", using the Base64 encoder we get the Bearer Token which is "ZWJWcGlLb2FPaWJOTmlyMVB5ekVYUVVxWjpxNkVCdElnb3plRlI2QTlNSjVuYmpEdFUzcjE2TUN2cldtcHRjclljWUU5b0NVVmpjaw==".

Refer to the official POST method document, we can write the following curl command.

$$curl - XPOST - H"Content - Type : application/x - www - form - urlencoded; charset = UTF - 8" - H"Authorization : BasicZWJWcGlLb2FPaWJOTmlyMVB5ekVYUVVxWjpxNkVCdElnb3plRlI2QTlNSjVuYmpEdFUzcjE2TUN2cldtcHRjclljWUU5b0NVVmpjaw == " - d" grant\_type = client_credentials" https : //api.twitter.com/oauth2/token$$

The following Bearer Token in the form of JSON will be returned as the Figure 3.1 shows.

$$\{"token\_type" : "bearer", "access\_token" : "AAAAAAAAAAAAAAAAAAAAAFtoNAEAAAAATy6AOuQ3pCV1feS\%2FxH4hQPAsTC0\%3DubFy4JL5rFvqt4JhNRXC2HAG6pSEalCxIb2PZwidnj9T5APtBT"\}$$

3.1.2 Using the Bearer Token to Obtain Tweets

In order to obtain the tweets, we should use the GET method from cURL command. In the command, we should specify the number of tweets and the id of the user that needs to be obtained. In our example case, we specified

```
C:¥Users¥かき>curl -H "Authorization: Bearer AAAAAAAAAAAAAAAAAAAAAFtoNAEAAAAATy6AOuQ3pCV1feS%2FxH4hQPAsT
C0%3DubFy4JL5rFvqt4JhNRXC2HAG6pSEalCxIb2PZwidnj9T5APtBT" "https://api.twitter.com/1.1/statuses/user_time
line.json?screen_name=thankyou_lucky&count=10"
[{"created_at":"Tue Sep 08 11:26:39 +0000 2015","id":641211039275573250,"id_str":"641211039275573250","t
ext":"¥u4eca¥u3046¥u3061¥u306f¥u3061¥u3083¥u3093¥u3068¥u7b11¥u3048¥u3066¥u3044¥u308b¥u306e¥u304b¥u306a",
"truncated":false,"entities":{"hashtags":[],"symbols":[],"user_mentions":[],"urls":[]},"source":"¥u003ca
href=¥"http:¥/¥/twitter.com¥/download¥/iphone¥" rel=¥"nofollow¥"¥u003eTwitter for iPhone¥u003c¥/a¥u003e
","in_reply_to_status_id":null,"in_reply_to_status_id_str":null,"in_reply_to_user_id":null,"in_reply_to_
user_id_str":null,"in_reply_to_screen_name":null,"user":{"id":3235118730,"id_str":"3235118730","name":"¥
u672c¥u97f3¥u57a2","screen_name":"thankyou_lucky","location":"","description":"","url":null,"entities":{
"description":{"urls":[]}},"protected":false,"followers_count":6,"friends_count":15,"listed_count":0,"cr
eated_at":"Wed Jun 03 14:48:14 +0000 2015","favourites_count":1,"utc_offset":null,"time_zone":null,"geo_
enabled":true,"verified":false,"statuses_count":27,"lang":null,"contributors_enabled":false,"is_translat
or":false,"is_translation_enabled":false,"profile_background_color":"C0DEED","profile_background_image_u
rl":"http:¥/¥/abs.twimg.com¥/images¥/themes¥/theme1¥/bg.png","profile_background_image_url_https":"https
:¥/¥/abs.twimg.com¥/images¥/themes¥/theme1¥/bg.png","profile_background_tile":false,"profile_image_url":
"http:¥/¥/pbs.twimg.com¥/profile_images¥/640788562972053504¥/Fe7eY4OK_normal.jpg","profile_image_url_htt
ps":"https:¥/¥/pbs.twimg.com¥/profile_images¥/640788562972053504¥/Fe7eY4OK_normal.jpg","profile_link_col
or":"1DA1F2","profile_sidebar_border_color":"C0DEED","profile_sidebar_fill_color":"DDEEF6","profile_text
_color":"333333","profile_use_background_image":true,"has_extended_profile":false,"default_profile":true
,"default_profile_image":false,"following":null,"follow_request_sent":null,"notifications":null,"transla
tor_type":"none","withheld_in_countries":[]},"geo":null,"coordinates":null,"place":null,"contributors":n
```


Figure 3.2: Using GET Method to Obtain the Private Opinion Tweets

the number of tweets as 10 and the id of the user as "thankyou_lucky"(the name of this user is "本音垢").

Refer to the official GET method document, we can write the following curl command.

$$curl - H"Authorization : BearerAAAAAAAAAAAAAAAAAAAAAAFtoNAEAAAAATy6AOuQ3pCV1feS\%2FxH4hQPAsTC0\%3DubFy4JL5rFvqt4JhNRXC2HAG6pSEalCxIb2PZwidnj9T5APtBT""https://api.twitter.com/1.1/statuses/user\_timeline.json?screen\_name = thankyou\_lucky\&count = 10"$$

The obtained tweets in the form of JSON will be returned as the Figure 3.2 shows.

So far, we have introduced the process of crawling the data of the private opinion accounts in Twitter. As for the method to determine that the crawled data is private opinion, we default all tweets posted on the private opinion accounts and venom accounts are the private opinion. However, due to the weak visibility of JSON data, we can use some online visualization tools to display it in the form of a tree. Alternatively, in our research, we will perform other more complex data processing, which we will introduce in the next section.

3.2 Tweets Processing

In the process of data processing, we mainly carried out the following work, and we will explain each work in detail.

1. Tweets format conversion
2. Solving the garbled code problem
3. Tweets Cleaning
4. Tweets Annotation

3.2.1 Tweets Format Conversion

There are many different ways to describe data. CSV and JSON are different text-based data formats. More specifically, it's like a rule for writing data. These formats are used to communicate between computers and, in particular, to transfer data over the Internet.

In the previous part, the data we got from Twitter is in JSON format. JSON is an abbreviation for JavaScript Object Notation. This data uses a description method called a dictionary, which has a one-to-one correspondence. The characteristic of this format is that it has a high affinity with JavaScript, as the name suggests. Since JSON is composed of JavaScript objects, it can be handled by a program written in JavaScript as it is.

CSV stands for "Comma-Separated Values". As the name implies, they are separated by commas (","). This is not complicated, so it is easy to intuitively grasp the contents of the data. Also, this file format has a good affinity with Microsoft's Excel and Google Spread Sheet.

In short, JSON is used to create the list in the list, and CSV is used for the table data. In our research, the dataset needs to be manually annotated, and CSV files can be used to directly read and annotate the readable text. Moreover, Pandas, a data analysis library in Python language used in this study, can easily process CSV files. Based on the above reasons, we have carried out the work of file format conversion from JSON to CSV.

3.2.2 Solving the Garbled Code Problem

If we open the converted CSV file directly using Microsoft's Excel in Windows system, the crawled Japanese texts will become garbled characters as the Figure 3.3 shows.

In order to solve the garbled problem, we need to mention Unicode. Unicode is an industry standard in computer science, including character

	created_at	id	id_str	text	truncated	source	in_reply_to	in_reply_to	in_reply_
0	Wed Apr 1	8.55E+17	8.55E+17	浣褲倧銈撱奄銈夔壇銇嬤沙銇熈軌銆併乢銇ｃ仭銉曨偐銉垏竟銇娵　銇勦煓?@05ma_ma_i	FALSE	<a href="http://twitter.com/download/iph			
1	Tue Dec 1	8.09E+17	8.09E+17	蹇樸倢銈嬫€偈猶銈併惟銆?鎏ｃ伴銇　啊鍗樸伝瑗€銇堛伏銈傘					

Figure 3.3: Garbled Data

sets, encoding schemes, etc. Unicode was created to solve the limitations of traditional character encoding schemes. It sets a unified and unique binary encoding for each character in each language to meet the requirements of cross-language and cross-platform text conversion and processing. If various character encodings are described as local dialects, then Unicode is a language developed by countries around the world. In this language environment, there will be no more language coding conflicts, and content in any language can be displayed on the same screen, which is the greatest benefit of Unicode. It is to encode all the text in the world with 2 bytes uniformly. With uniform encoding like this, 2 bytes is enough to hold most of the text in all the languages in the world. The scientific name of Unicode is "Universal Multiple-Octet Coded Character Set", a universal multi-octet coded character set, referred to as UCS. UCS2 is now used, that is, 2-byte encoding, and UCS4 was developed to prevent 2 bytes from being insufficient in the future.

There is one called "Zero Width No-Break Space" in UCS encoding, and its encoding is FEFF. And FEFF is a non-existent character in UCS, so it should not appear in the actual transmission. The UCS specification recommends that we transmit the characters "Zero Width No-Break Space" before transmitting the byte stream. In this way, if the receiver receives FEFF, it indicates that the byte stream is Big-Endian; if it receives FFFE, it indicates that the byte stream is Little-Endian. Therefore, the character "Zero Width No-Break Space" is also called BOM (byte-order mark).

UTF-8 (Universal Character Set/Unicode Transformation Format) is a variable-length character encoding for Unicode. It can be used to represent any character in the Unicode standard, and the first byte in its encoding is still compatible with ASCII, so that the original software that handles ASCII characters can continue to use it without or with only a few modifications. As a result, it has gradually become the preferred encoding for emails, web

Table 3.1: Common patterns for arguments and attack

arguments	attacks
と思う/思わない	だけど
なぜなら	とはいえ
それで	ところで
つまり	しかし
一方	けれども
あるいは	でも
したがって	なのに
そのため	ところが

pages, and other applications that store or transmit text.

In our research, we saved the encoding of the CSV file as UTF-8 with BOM which solved the garbled problem and displayed Japanese normally. UTF-8 does not need BOM to indicate byte order, but can use BOM to indicate encoding. The UTF-8 encoding for the characters "Zero Width No-Break Space" is EF BB BF. So if the receiver receives a byte stream starting with EF BB BF, it knows that this is UTF-8 encoding. The encoding format of text documents in Windows systems is specified by BOM, and characters should be written at the beginning of the file: EF BB BF (UTF-8 BOM encoding), so that UTF-8 encoded files can be correctly recognized.

3.2.3 Tweets Cleaning

As the Figure 3.3 shows, there are many parts of the data that are not used in analyzing the private opinion structure. At this stage, we mostly removed unwanted fields and mainly keep the serial number and text. In the follow-up research, we need to count the word frequency, and we will also perform operations such as merging duplicate data.

3.2.4 Tweets Annotation

In our research, binary classification by machine learning is mainly performed twice. The first binary classification is to distinguish whether a text is argumentative or non-argumentative. The second binary classification is to distinguish whether there is an attack relation between argumentative texts.

During the annotation of the first binary classification, argumentative ones are labeled as 1, non-argumentative ones are labeled as 0. We divide argumentative tweets into sub tweets and annotate the sub tweets with attack relation. During the annotation of the second binary classification, if there is

attack relation exists, pairs of sub tweets are labeled as 1, otherwise labeled as 0. We annotated based on the studies of Tashiro [15] and Ijuin [11, 16], which focused on common patterns for arguments and attack shown in Table 3.1.

3.3 Modeling with VAF

In our study, we model weighted argumentation frameworks for the acquired Twitter data based on valued-based argumentation frameworks mentioned in 2.4.

First we give some definitions.

Definition 3.3.1 (Discussion Graph)**.** Let Γ be a non-empty set of tweets. The Discussion Graph (DisG) for Γ is the directed graph (V,E) which

- for every tweet in Γ there is a node in V and
- if tweet a attacks tweet b there is a directed edge (a,b) in E.

Only the nodes and edges obtained by applying this process belong to V and E, respectively.

Definition 3.3.2 (Annotated Discussion Graph)**.** A annotated discussion graph (ADisG) is a tuple (V,E,A) , where (V,E) is a discussion graph and A is a annotation function for edges $A : E \to S$, where S represents a set of possible semantic relations for any directed edge (a,b).

In our research, the set of semantic relations S for a directed edge (a,b) we have considered is {attacks,none}, attacks is the same as the standard argumentation frameworks, meaning that tweet a does not agree with the claim expressed in tweet b and otherwise none.

Definition 3.3.3 (Weighted Annotated Discussion Graph)**.** A weighted annotated discussion graph (WADisG) is a tuple $\langle V, E, A, R, W \rangle$, where (V,E,A) is a ADisG, R is a nonempty set of ordered values and W is a weighting function $W : V \to R$ that assigns a weight value in R to each tweet in V, representing the reliability of the tweet.

When we design the weighting function, we consider factors such as the Twitter user's profile, icon, header, age etc. Later we will explain the weight function in detail in 5.2.

Definition 3.3.4 (Valued-Based Argumentation Frameworks)**.** A valued-based argumentation framework (VAF) is a tuple $\langle \mathcal{AR}, attacks, R, Val, Valp$

$ref\rangle$ where $\mathcal{AR}$ is a set of arguments, $attacks$ is an irreflexive binary relation on $\mathcal{AR}$, R is a nonempty set of values, Val is a valuation function $Val : \mathcal{AR} \rightarrow R$ that assigns to each argument in $\mathcal{AR}$ a weight value in R, and $Valpref \subseteq R \times R$ is a preference relation on R (transitive, irreflexive and asymmetric), reflecting the value preferences of arguments.

Definition 3.3.5 (Defeat)**.** Given a VAF $\langle \mathcal{AR}, attacks, R, Val, Valpref\rangle$ and arguments a and b in $\mathcal{AR}$, we say that a defeats b (written defeats (a,b)) iff $(a, b) \in attacks$ and $(Val(b), Val(a)) \notin Valpref$. We also say that a effectively attacks b.

Definition 3.3.6 (Conflict-free)**.** Given a VAF $\langle \mathcal{AR}, attacks, R, Val, Valpre f\rangle$ and a set $S \subseteq \mathcal{AR}$ of arguments we say that S is conflict-free iff $\forall a, b \in S, (a, b) \notin attacks$ or $(Val(b), Val(a)) \in Valpref$; i.e. $\neg defeats(a, b)$.
There might be an attack between two arguments in a conflict-free set, if this relation of attack is not effective; i.e. given a conflict-free set S and arguments $a, b \in S$, it can be the case that a attacks b whenever b is preferred than a according to the preference relation $Valpref$. For instance, if the set of ordered values R are instantiated to the natural numbers N, the valuation function Val to a mapping from arguments to N and the preference relation $Valpref$ to the total order relation on N , then it could be that a attacks b whenever $Val(b) > Val(a)$.

Definition 3.3.7 (Acceptable Argument)**.** Given a VAF $\langle \mathcal{AR}, attacks, R, Va l, Valpref\rangle$, a set $S \subseteq \mathcal{AR}$ of arguments and an argument $a \in \mathcal{AR}$ we say that a is acceptable with respect to S iff $\forall b \in \mathcal{AR}$, defeats (b,a) implies that$\exists c \in S$, defeats (c,b).

Definition 3.3.8 (Admissible Extension)**.** Given a VAF $\langle \mathcal{AR}, attacks, R, Va l, Valpref\rangle$ and a set $S \subseteq \mathcal{AR}$ of conflict-free arguments we say that S is an admissible extension if for any $a \in S$, a is acceptable with respect to S.

Definition 3.3.9 (Defend)**.** A set S defends an argument a if for any argument b which attacks a, S contains an argument that attacks b.

Definition 3.3.10 (Complete Extension)**.** Given a VAF $\langle \mathcal{AR}, attacks, R, Va l, Valpref\rangle$ and a set $S \subseteq \mathcal{AR}$ we say a set S is a complete extension if S is admissible and contains all arguments it defends.

Definition 3.3.11 (Grounded Extension)**.** Given a VAF $\langle \mathcal{AR}, attacks, R, Va l, Valpref\rangle$ and a set $S \subseteq \mathcal{AR}$ we say a set S is grounded if S is the minimal (w.r.t. set inclusion) complete.

Definition 3.3.12 (VAF for a Weighted Annotated Discussion Graph)**.** Let $G = \langle V, E, A, R, W \rangle$ be a WADisG. We defined the valued-based argumentation frameworks for G is $F = \langle V, attacks, R, W, Valpref \rangle$, where

- the set of arguments is the set of nodes (or tweets) V,
- attacks is the same as the standard argumentation frameworks and are defined as follows:

$$attacks = \{(a, b)|(a, b) \in E \quad and \quad A(a, b) = attacks\}$$

- R is the non-empty set of ordered values that models the reliability or weight of tweets,
- the valuation function for arguments is the weighting function $W : V \rightarrow R$ for tweets. and
- the preference relation $Valpref \subseteq R \times R$ is the ordering relation over R.

In our framework, we give weight to arrows rather than nodes because our data set is special compared with typical twitter conversation analysis. In a typical twitter conversation, a tweet usually has many replies from different users, which is suitable for the traditional VAF that gives weight to different nodes (arguments). In our dataset, users rarely receive a reply to a tweet, but they usually quote the arguments of others in a tweet before giving a counter-argument. Therefore, we divide such tweets into sub tweets and annotate the sub tweets with attack relation. The weight of the same user on a topic is calculated by the weighting function therefore the node weights at both ends of the attack are the same. We give weights to different arrows to represent the overall reliability of the tweet. In the VAF, the grounded extension $S \subseteq T$ of F is the accepted set of tweets based on the weighting scheme W and we refer to it as the solution of G, i.e. the set of tweets with high reliability.

3.4 Implementing AF Using ASP

To compute in a stratified program the required predicate for being defended, we have to use the order $<$ over the domain elements (such an order is provided by all ASP-solvers) and derive corresponding predicates for infimum, supremum, and successor w.r.t. $<$. In our ground extension, the $\pi_<$ ordering is defined as follows.

Listing 3.1: AF Answer Set Programming Ordering Sample Code

```
lt(X,Y) :- arg(X), arg(Y), X<Y.
nsucc(X,Z) :- lt(X,Y), lt(Y,Z).
succ(X,Y) :- lt(X,Y), not nsucc(X,Y).
ninf(Y) :- lt(X,Y).
inf(X) :- arg(X), not ninf(X).
nsup(X) :- lt(X,Y).
sup(X) :- arg(X), not nsup(X).
```

Here $\pi_<$ is indeed stratified and constraint-free. Hence, $\pi_<(\hat{F})$ yields exactly one answer set for each AF F. To illustrate the purpose of $\pi_<$, recall our example framework F, and assume the arguments are ordered as follows: $a < b < c < d < e$. For this particular order, the single answer set S_0 of $\pi_<(\hat{F})$ contains

$$\{inf(a), succ(a, b), succ(b, c), succ(c, d), succ(d, e), sup(e)\}.$$

We define the required predicate defended(X) which itself is obtained via a predicate defended_upto(X, Y) with the intended meaning that argument X is defended by the current assignment with respect to all arguments $U \leq Y$. In other words, we perform a loop starting with the infimum Y and then use the successor predicate to derive defended_upto(X, Y) for all further Y. If we arrive at the supremum element in this way, i.e. defended_upto(X, Y) is derived for the supremum Y, we finally obtain defended(X). The $\pi_{defended}$ is defined as follows.

Listing 3.2: AF Answer Set Programming Defended Sample Code

```
defended_upto(X,Y) :- inf(Y), arg(X), not defeat(Y,X).
defended_upto(X,Y) :- inf(Y), in(Z), defeat(Z,Y), defeat(Y,X
    ).
defended_upto(X,Y) :- succ(Z,Y), defended_upto(X,Z), not
    defeat(Y,X).
defended_upto(X,Y) :- succ(Z,Y), defended_upto(X,Z), in(V),
    defeat(V,Y), defeat(Y,X).

defended(X) :- sup(Y), defended_upto(X,Y).
```

And $\pi_{grounded} = \pi_< \cup \pi_{defended} \cup in(X) : -defended(X)$.

Here $\pi_{grounded}$ is also stratified. The relevant predicate in(a) is derived for some argument a if defended_upto(a, b) holds for each b. However, if there is an unattacked argument c which attacks a, defended_upto(a, c) is not derived. It is thus not relevant in which order we derive the predicates defended_upto(a, b). Consequently, the particular definition of the order ¡, from which we obtained the inf(·), succ(·, ·), and sup(·) predicates used in π defended, plays no role. Any total order over the constants can be used.

3.5 Implementing VAF Using ASP

In VAF implementation, the $\pi_{<}$ ordering and the $\pi_{defended}$ is the same as those implemented in AF. And the π_{vaf} is defined as follows.

Listing 3.3: VAF Answer Set Programming Sample Code

```
%% valpref preference relation; transitivity
valpref(X,Y) :- valpref(X,Z), valpref(Z,Y).

%% pref computes preference of arguments depending on
%% the preference relation valpref
pref(X,Y) :- valpref(U,V), val(X,U), val(Y,V).

%% transitivity of pref
pref(X,Y) :- pref(X,Z), pref(Z,Y).

%% argument x defeats argument y, iff
%% x attacks y, and y is not preferred to x
defeat(X,Y) :- att(X,Y), not pref(Y,X).
```

Chapter 4

Experimentation

In 3.2.4, we mentioned that in our research, we carried out two binary classifications. In this chapter, we will describe our classification experimentation. In addition, we have selected some popular machine learning algorithms. We also hope to compare and find the best algorithm on our data set through this experimentation.

4.1 Tweets segmentation

After we finish processing the tweet data, the next step is to carry out some basic morphological analysis work, such as word segmentation because in the subsequent analysis, we will need to use some information like word frequency.

Morphological analysis refers to the work of dividing the text data of natural language without syntax information annotation into morpheme (the smallest unit with meaning in the language) according to the syntax of the object language and the dictionary containing word part of speech and other information, so as to distinguish the part of speech of each morpheme. In our study, we used MeCab [18] for morphological analysis. Source code for morphological analysis is shown in source code Listing 4.1.

Listing 4.1: Morphological Analysis Sample Code

```
import pandas as pd
import MeCab
import pandas as pd
import re

sample_csv = pd.read_csv(r'C:\Users\zjq\dataset\csv\annotated
    \data.csv')

```

```
#Filter out the data with argumentative value of 1.
df = sample_csv.loc[sample_csv['argumentative']==1]

#Create a MeCab instance.
mecab = MeCab.Tagger('')

#Create a data frame to output the final result.
df1 = pd.DataFrame( columns=['word0','word1'] )

#Read the CSV file stored in df line by line.
for row, item in df.iterrows():
    #Variables that store the results of morphological analysis. In order to process line by line, clear it once before processing.
    result = ''
    #Morphological analysis is performed line by line and divided into words.
    result = mecab.parse(item.text)
    #Set variables that are split into separate lines.
    lines = result.split("\n")
    #Read the variables of the morphological analysis result separated by one line from above.
    for words in lines:
        #Because it is separated by tabs and commas, reset the new variable.
        word = re.split('\t|,',words)
        #Store the results in the data frame.
        df1 = df1.append({'word0':word[0], 'word1':word[1]}, ignore_index=True)


#Display all.
pd.set_option('display.max_rows', None)
#Display Data Frame.
df1
```

The result of morphological analysis displayed as the Figure 4.1 shows. After we finish morphological analysis, we count the word frequency information. And the source code is shown in Listing 4.2.

Listing 4.2: Store Word Frequency Information into Dictionary Sample Code

```
#Calculate word frequency through value_counts.
data_counts = df1['word0'].value_counts()

#Convert word frequency results into DataFrame format.
# The index of the converted DataFrame is the word that needs to be counted, and the column is the number of times the word appears
df_data_counts = pd.DataFrame(data_counts)
```

	word0	word1
0	自分	名詞
1	の	助詞
2	結果	名詞
3	論	名詞
4	や	助詞
5	世間体	名詞
6	や	助詞
7	身近	名詞
8	な	助動詞
9	人	名詞

Figure 4.1: Result of Morphological Analysis

```

#Turn words into lists.
names = df_data_counts.index.values.tolist()

#Convert word frequency into a list.
counts =df_data_counts['word0'].tolist()

results = []
#Convert to dictionary.
for name, count in zip(names, counts):
    results.append({"name": name, "count": count})
dic1 = results
print(results)

#Write to a file.
import csv
labels = ['name', 'count']
with open('dic.csv', 'w') as f:
    writer = csv.DictWriter(f, fieldnames=labels)
    writer.writeheader()
    for elem in results:
        writer.writerow(elem)
```

4.2 Feature Selection

Now that we have the word frequency information, we will select the most useful features for training our model. The feature selection of text quantifies the feature words extracted from the text to represent the text information. Converting them from an unstructured raw text to a structured information that can be recognized and processed by a computer, that is, scientific abstraction of the text and establishment of its mathematical model to describe and replace the text. It enables the computer to realize the recognition of text through the calculation and operation of this model. Since text is unstructured data, in order to mine useful information from a large amount of text, it must first convert the text into a processable structured form.

In our research, we use the feature items obtained by word frequency statistics to represent each dimension in the text vector, and combine TF-IDF to select more important dimensions.

TF is Term Frequency and IDF is Inverse Document Frequency. TF-IDF is a statistical method used to evaluate the importance of a word to a document set or a document in a corpus. The importance of a word increases proportionally to the number of times it appears in the document, but decreases inversely to the frequency it appears in the corpus.

In a given document, term frequency (TF) refers to the frequency with which a given word appears in the document. This number is normalized to the term count to prevent it from skewing towards long files. (The same word may have a higher number of words in a long file than a short file, regardless of whether the word is important or not.) For a word t_i in a particular file, its importance can be expressed as:

$$tf_{i,j} = \frac{n_{i,j}}{\Sigma_k n_{k,j}}$$

$n_{i,j}$ is the number of occurrences of the word t_i in file d_j. The denominator is the sum of the occurrences of all words in the file d_j.

Inverse document frequency (IDF) is a measure of the general importance of words. The IDF for a particular word can be calculated by dividing the total number of documents by the number of documents containing the word, and then taking the logarithm of the quotient to get:

$$idf_i = log\frac{|D|}{\{j : t_i \in d_j\}}$$

$|D|$ represents the total number of documents in the corpus and $\{j : t_i \in d_j\}$ represents the number of files containing the term t_i.

```
In [25]:  tfidfs.toarray().shape
Out[25]:  (10, 2701)
```

Figure 4.2: Dimension of Feature Words

Multiply the above two items to get the formula of TF-IDF.

$$tfidf_{i,j} = tf_{i,j \times idf_i}$$

A high word frequency within a particular document, and a low document frequency of that word in the entire document set, can result in a highly weighted TF-IDF. Therefore, TF-IDF tends to filter out common words and keep important words.

At the beginning, as shown in the Figure 4.2, we got 2701 feature words. If all words are used as feature items, the dimension of the feature vector will be too huge, resulting in too much calculation.

Listing 4.3: Feature Selection USing TF-IDF Sample Code

```
from sklearn.feature_extraction.text import TfidfVectorizer
dic0 = open(r'C:\Users\zjq\dic\dic0.csv','r', encoding='UTF-8
    ').read()
dic1 = open(r'C:\Users\zjq\dic\dic1.csv','r', encoding='UTF-8
    ').read()
dic2 = open(r'C:\Users\zjq\dic\dic2.csv','r', encoding='UTF-8
    ').read()

docs = [
    dic0, dic1, dic2
]
vectorizer = TfidfVectorizer(use_idf = True, max_df=0.9)
# Ignore words appear in more than 90% of the entire document

tfidfs = vectorizer.fit_transform(docs)
print('feature_names:', vectorizer.get_feature_names())

words = vectorizer.get_feature_names()
for doc_id, vec in zip(range(len(docs)), tfidfs.toarray()):
    print('doc_id:', doc_id)
    for w_id, tfidf in sorted(enumerate(vec), key=lambda x:
        x[1], reverse=True):
        lemma = words[w_id]
        print('\t{0:s}{1:f}'.format(lemma, tfidf))
```

First we try to use PCA for dimension reduction. Principal Component Analysis is a common data analysis method, which is often used for dimension

reduction of high-dimensional data and can be used to extract the main feature components of the data.

The goal of PCA is to find r (r<n) new variables from the original n variables, so that they reflect the main characteristics of things, compress the scale of the original data matrix, reduce the dimension of the feature vector, and select the least number of dimensions to generalize the most important features. Each new variable is a linear combination of the original variables, which reflects the comprehensive effect of the original variables and has a certain practical meaning. These r new variables are called "principal components", they can largely reflect the influence of the original n variables, and these new variables are uncorrelated and orthogonal. Through principal component analysis, the data space is compressed, and the characteristics of multivariate data are visually expressed in the low-dimensional space.

Listing 4.4: Dimension Reduction USing PCA Sample Code

```
from sklearn.decomposition import PCA

#n_components specifies how many principal components should be calculated
#If we specify a real number between 0 and 1, the principal component is calculated until the cumulative contribution ratio(the sum of the contribution ratios of the principal components, contribution ratio represents the importance of each principal component) reaches that value.
pca = PCA(n_components = 0.9, whiten = True)

#Perform principal component analysis
pca.fit(tfidfs.toarray())

#Checking the number of principal components
pca.n_components_

#Display words with high factor loading(effect of each variable on each principal component which can estimate the meaning of each principal component) for each main component

for i in range(pca.n_components_):
    tc = list(zip(pca.components_[i],words))
    tc.sort()
    tc.reverse()
    print("[PCA%d]" % (i+1))
    for v,t in tc[:10]:
        print("%f,%s" % (v,t))
```

Figure 4.3 shows part of the results of PCA dimension reduction. The first principal component and the second principal component are the two

[PCA1]
0.133250,道具
0.133250,衣食住
0.133250,聞き
0.133250,知り
0.133250,産ん
0.133250,法律
0.133250,本人
0.133250,日々
0.133250,敬意
0.133250,成人
[PCA2]
0.102598,良かっ
0.083407,飲も
0.083407,運命
0.083407,遅い
0.083407,迎え
0.083407,諦める
0.083407,誘い
0.083407,覚める
0.083407,落ち込ん
0.083407,苦労

Figure 4.3: Results of PCA Dimension Reduction

most important principal components, which best summarize the most important features of the original dataset. But we can also find that the results are not satisfactory. If it is an ideal situation, the feature words of each dimension will be able to be smoothly summed up to be similar. One of the reasons for this result is that our dataset is not very large, which is also related to the fact that our dataset is not very large. Since the tweets posted by these accounts represent the real thoughts of users, a considerable number of users set their accounts as private accounts for their own reasons.

Anyway, we also tried to use PCA for visualization analysis, as shown in the Figure 4.4. We plot with the first principal component as the x-axis and the second principal component as the y-axis. Different points represent different users. The closer the users are, the greater the similarity of their tweets under this dimensionality reduction.

4.3 Model Training

After completing the feature selection using TF-IDF and PCA, combined with Japanese grammar (refer to Japanese papers), we determined the feature words and performed binary classification.

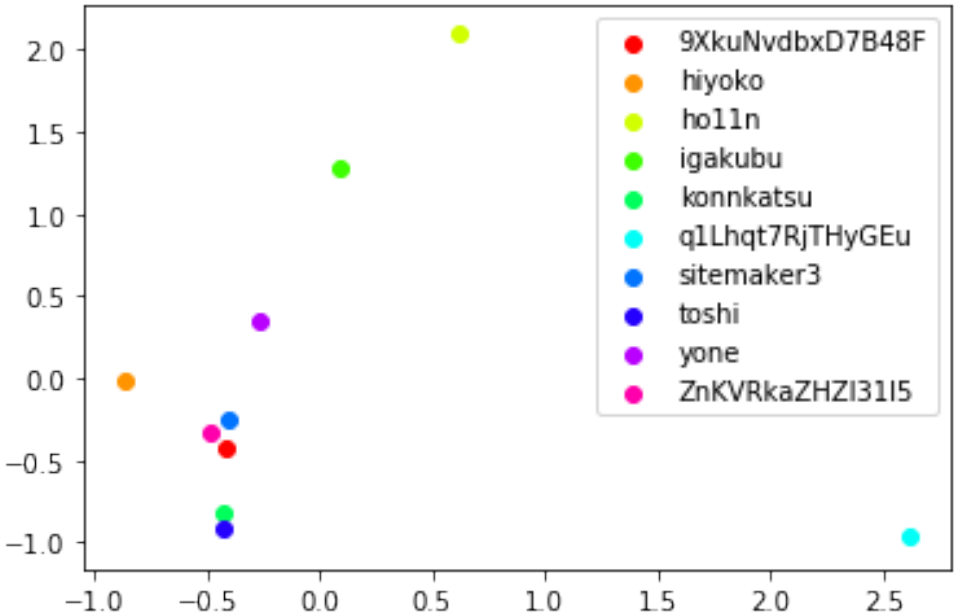


Figure 4.4: Visualization Analysis for PCA Dimension Reduction

4.3.1 Argument Classification

For argument classification, we selected the features as follows. “思”,“違う”,“なら”,“意思”,“べき”,“だけ”,“こと”,“に対して”,“ので”,“それで”,“つまり”,“一方”,“あるいは”,“したがって”,“そのため”.

The “argumentative” column represents the true label of the text, 1 means the text is argumentative, and 0 means the text is non-argumentative.

We divided 80% of the data into training data and 20% of the data into test data.

Listing 4.5: Argument Classification Sample Code

```
import pandas as pd
df = pd.read_csv('args.csv',header = 0)
from sklearn.model_selection import train_test_split

feature = df.loc[:, ['思','違う','なら','意思','べき','だけ
    ','こと','に対して','ので','それで','つまり','一方','ある
    いは','したがって','そのため']]
target = df.loc[:, ['argumentative']]

train_feature ,test_feature, train_target, test_target =
    train_test_split(feature, target, train_size=0.8,
    random_state=1)
```

4.3.2 Attack Classification

For argument classification, we selected the features as follows. “けど”,“なら”,“でも”,“なく”,“ない”,“違う”,“別に”,“おかしい”,“のに”,“とはいえ”,“とことで”,“しかし”,“ところが”,“だけど”,“なのに”.

The “attack” column represents the true label of the relation between the subtexts of the text, 1 means that there is an attack relation between the subtexts of the text, and 0 means that it does not exist.

Same as argument classification experiment, we divided 80% of the data into training data and 20% of the data into test data.

Listing 4.6: Attack Classification Sample Code

```
import pandas as pd
df = pd.read_csv('atts.csv',header = 0)
from sklearn.model_selection import train_test_split

feature = df.loc[:, ['けど','なら','でも','なく','ない','違う','別に','おかしい','のに','とはいえ','ところで','しかし','ところが','だけど','なのに']]
target = df.loc[:, ['attack']]

train_feature ,test_feature, train_target, test_target = train_test_split(feature, target, train_size=0.8, random_state=1)
```

Here random_state refers to the random number seed. The random number seed is actually the number of the group of random numbers. When repeated experiments are required, it is guaranteed to get the same set of random numbers. In the next chapter, we will experiment with different random number seeds in order to evaluate the performance of different machine learning algorithms. Here we set it to 1.

Chapter 5

Evaluation

5.1 Evaluation on F Score

We will use some commonly used machine learning algorithms on our dataset and evaluate the performance of these algorithms by F Score.

5.1.1 Decision tree

First we use decision tree. Decision tree learning adopts a top-down recursive method, and its basic idea is to construct a tree with the fastest entropy drop based on information entropy. The entropy value to the leaf node is zero, and the instances in each leaf node belong to the same class.

In information theory and probability statistics, entropy is a measure of the uncertainty of a random variable. Let X be a discrete random variable with a finite number of values, and its probability distribution is as follows.

$$P(X = x_i) = p_i, i = 1, 2, ..., n$$

Then the entropy of the random variable X is defined as:

$$H(X) = -\sum_{i=1}^{n} p_i log p_i$$

In the above formula, if p_i=0, then define 0log0=0. Usually, the logarithm in the formula is base 2 or base e (natural logarithm). At this time, the unit of entropy is called bit or nat. It can be seen from the definition that the entropy only depends on the distribution of X, and has nothing to do with the value of X, so the entropy of X can also be recorded as H(p), that is:

$$H(p) = -\sum_{i=1}^{n} p_i log p_i$$

The greater the entropy, the greater the uncertainty of the random variable. It can be verified from the definition:

$$0 \leq H(p) \leq logn$$

When the random variable is determined, the minimum value of the entropy is 0, and when the entropy value is the largest, the uncertainty of the random variable is the largest.

Given a random variable (X,Y), its joint probability distribution is

$$P(X = x_i, Y = y_j) = p_{ij}, i = 1, 2, ..., n, j = 1, 2, ..., m$$

Conditional entropy H(Y| X) represents the uncertainty of random variable Y under the condition of known random variable X, and the conditional entropy H(Y| X) of random variable Y under the given condition of random variable X is defined as X Mathematical expectation on X of the entropy of the conditional probability distribution of Y given the conditions:

$$H(Y|X) = \sum_{i=1}^{n} p_i H(Y|X = x_i)$$

Here, $p_i = P(X = x_i)$, i=1,2,...,n. When the probability in entropy and conditional entropy is obtained by data estimation (maximum likelihood estimation), the corresponding entropy and conditional entropy are called empirical entropy and conditional empirical entropy, respectively. At this time, if there is 0 probability, then let 0log0=0.

The information gain represents the degree to which the uncertainty of the information of class Y is reduced by knowing the information of feature X. The information gain g(D,A) of feature A to dataset D is defined as the difference between the empirical entropy H(D) of set D and the empirical conditional entropy H(D|A) of feature A, namely:

$$g(D, A) = H(D) - H(D|A)$$

Generally, the difference between the entropy H(Y) and the conditional entropy H(Y|X) is called mutual information. The information gain in decision tree learning is equivalent to the mutual information between classes and features in the training dataset. We set criterion as entropy, which means information entropy. Then we load the decision tree classification model, create decision tree using training data and predict test data using the created decision tree.

Listing 5.1: Decision Tree Sample Code

```
from sklearn import tree

clf = tree.DecisionTreeClassifier(criterion = 'entropy')
clf = clf.fit(train_feature,train_target)
predicted = clf.predict(test_feature)
```

We will evaluate all the algorithms with F score after introducing them. In addition to this, we will also show the visualization of decision trees.

Listing 5.2: Decision Tree Visualization Sample Code

```
import pydotplus
from IPython.display import Image
from six import StringIO

dot_data = StringIO()
tree.export_graphviz(clf,out_file = dot_data)
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
Image(graph.create_png())
```

The visualization result is shown in Figure 5.1. It is necessary to refer to the same feature many times whether in argument classification or attack classification. It is hard to say that it is the most appropriate method for data with a large amount of features.

5.1.2 SVM

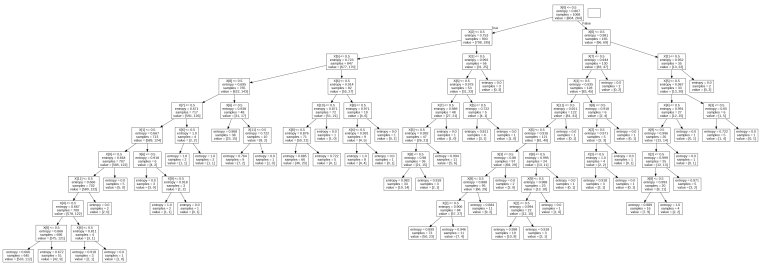
SVM (support vector machine) is a two class classification model. Its basic model is defined as the linear classifier with the largest interval in the feature space. Its learning strategy is to maximize the margin, which can finally be transformed into the solution of a convex quadratic programming problem.

We normalize with standard deviation for SVM. Here we use StandardScaler [19] to standardize features by removing the mean and scaling to unit variance. The standard score of a sample x is calculated as:

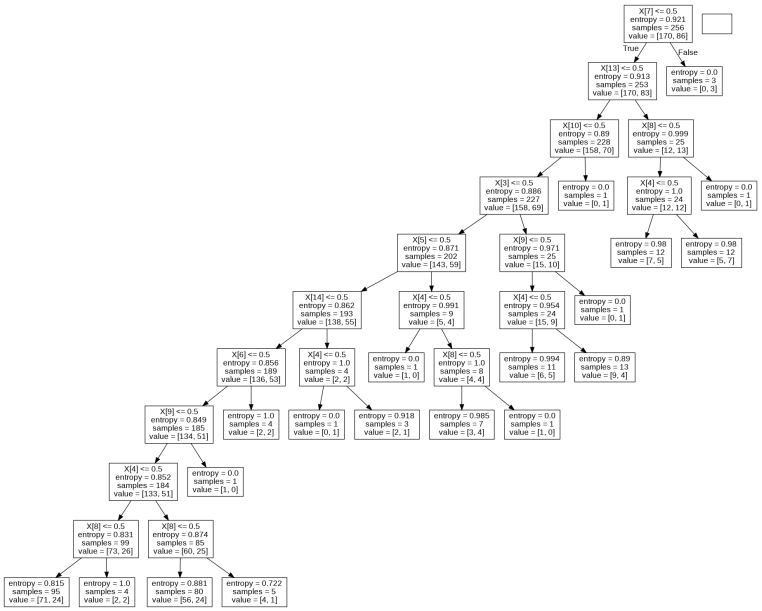
$$z = \frac{(x - u)}{s}$$

where u is the mean of the training samples or zero if with_mean=False, and s is the standard deviation of the training samples or one if with_std=False.

Centering and scaling happen independently on each feature by computing the relevant statistics on the samples in the training set. Mean and standard deviation are then stored to be used on later data using transform.



(a) Argument Classification



(b) Attack Classification

Figure 5.1: Visualization for Decision Tree

Empirically, normalization can make the features between different dimensions have a certain numerical comparison, which can greatly improve the accuracy of the classifier. The default values of both with_mean and with_std are True, i.e., u is not 0 and s is not 1. The fit function is to compute the mean and std to be used for later scaling. And the transform function is to perform standardization by centering and scaling.

Listing 5.3: Normalization Sample Code

```
from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

sc.fit(train_feature)
train_feature_std = sc.transform(train_feature)
test_feature_std = sc.transform(test_feature)
print(train_feature_std)
```

Linear separability means that two types of samples can be separated by a linear function, such as a straight line in two-dimensional space, a plane in three-dimensional space and a linear function in high-dimensional space.

In most cases, the data are not linearly separable. At this time, the hyperplane satisfying such conditions does not exist at all. For the nonlinear case, the processing method of SVM is to select a kernel function κ to solve the problem of linear indivisibility in the original space by mapping the data to a high-dimensional space. Specifically, when the data are nonlinearly separable, the support vector machine first completes the calculation in the low-dimensional space, then maps the input space to the high-dimensional feature space through the kernel function, and finally constructs the optimal separation hyperplane in the high-dimensional feature space, so as to separate the non-linear data that is not easy to be divided on the plane.

We set kernel as linear. Linear kernel function is

$$\kappa(x_1, x_2) = \langle x_1, x_2 \rangle$$

i.e., inner product in primitive space. The meaning of kernel function is that although it also converts the features from low dimension to high dimension, the kernel function calculates in the low dimension in advance, and displays the substantive classification effect in the high dimension, which avoids the complex calculation directly in the high-dimensional space.

Listing 5.4: SVM Sample Code

```
from sklearn import svm

clf_s = svm.SVC(kernel = 'linear')
```

```
clf_s.fit(train_feature_std,train_target)
predicted = clf_s.predict(test_feature_std)
```

5.1.3 MultinomialNB

NB is the abbreviation for Naive Bayes. There is a sample data set $D = d_1, d_2, ..., d_n$, the feature attribute set corresponding to the sample data is $X = x_1, x_2, ..., x_d$, and the class variable is $Y = y_1, y_2, ...y_m$, that is, D can be divided into y_m categories.

Where $x_1, x_2, ..., x_d$ are independent and random, then the a prior probability of Y is $P_{prior} = P(Y)$, and the a posterior probability of Y is $P_{post} = P(Y|X)$. It can be obtained by Naive Bayes algorithm, and the posterior probability can be calculated by a prior probability $P_{prior} = P(Y)$, evidence $P(X)$ and class conditional probability $P(X|Y)$:

$$P(Y|X) = \frac{P(Y)P(X|Y)}{P(X)}$$

Naive Bayes is based on the independence of each feature. In the case of a given category of , the above formula can be further expressed as the following formula:

$$P(X|Y = y) = \prod_{i=1}^{d} P(x_i|Y = y)$$

From the above two equations, the posterior probability can be calculated as:

$$P_{post} = P(Y|X) = \frac{P(Y)\prod_{i=1}^{d} P(x_i|Y)}{P(X)}$$

Since the size of $P(X)$ is fixed, only the numerator part of the above formula can be compared when comparing the posterior probability. So we can get a Naive Bayes calculation where the sample data belongs to the category:

$$P(y_i|x_1, x_2, ...x_d) = \frac{P(y_i)\prod_{i=1}^{d} P(x_i|y_i)}{\prod_{i=1}^{d} P(x_j)}$$

For our classification experiment, it can also be written as:

$$P(class|feature) = \frac{P(class)P(feature|class)}{P(feature)}$$

The features of our dataset are discrete and suitable for the use of multinomial models. MultinomialNB assumes that the prior probability of the feature is a multinomial distribution, that is, the following formula:

$$P(X_j = x_{jl}|Y = C_k) = \frac{x_{jl} + \lambda}{m_k + n\lambda}$$

Among them, $P(X_j = x_{jl}|Y = C_k)$ is the conditional probability of the l-th value of the j-th dimension feature of the k-th category. m_k is the number of samples of the kth class output in the training set. λ is a constant greater than 0 and takes a value of 1, that is, Laplace smoothing, and can also take other values.

In text classification, the prior and conditional probabilities of polynomials are as follows. Suppose a document $d = (t_1, t_2, ..., t_k)$, t_k is the word that has appeared in the document, and repetition is allowed, then the prior probability P(class c) is calculated as:

$$P(class \quad c) = \frac{total \quad number \quad of \quad words \quad in \quad class \quad c}{total \quad number \quad of \quad words \quad in \quad training \quad sample}$$

Class conditional probability $P(t_k|c)$ is calculated as:

$$P(t_k|c) = \frac{total \quad times \quad t_k \quad in \quad class \quad c \quad appears \quad every \quad document + 1}{(total \quad number \quad of \quad words \quad in \quad class \quad c + |V|}$$

V is the word list of the training sample (i.e., the word is extracted, and the word appears many times, only one is counted), and $\mid V \mid$ indicates how many kinds of words the training sample contains. $P(t_k|c)$ can be seen as how much evidence the word tk provides in proving that d belongs to class c, while $P(class \quad c)$ can be thought of as how much (how likely) is the class c as a whole.

Listing 5.5: MultinomialNB Sample Code

```
from sklearn.naive_bayes import MultinomialNB
clf = MultinomialNB()
clf = clf.fit(train_feature, train_target)
predicted = clf.predict(test_feature)
```

5.1.4 Passive Aggressive

Passive Aggressive is a linear classifier for online learning(learning sequentially each time data is given). We use Passive Aggressive processing

binary classification problem. In the binary classification, the import is x, the binary prediction is $y \in \{-1, +1\}$.

The linear classifier takes the model parameters W as features, inputs x , and predicts $sign(W^T x)$. Here sign(x) is a function that returns 1 if x is non-negative and -1 if it is negative.

In online learning, every time data is given, the parameters are updated using a pre-designed update formula. In Passive Aggressive, when the t-th data ($x^{(t)}$, $y^{(t)}$) is given, the parameter $W^{(t)}$ is updated to $W^{(t+1)}$ using the following formula.

$$W^{(t+1)} = W^{(t)} + \frac{l_{hinge}(x^{(t)}, y^{(t)}, W^{(t)})}{\| x^{(t)} \|^2} y^{(t)} x^{(t)}$$

$$l_{hinge}(x, y, W) = max(0, 1 - yW^T x)$$

If the current model can be classified with sufficient margin ($yW^T x > 1$), then $l_{hinge}(x, y, W) = 0$, the update is not performed. Otherwise, the update is performed by changing the update magnitude according to the wrong proportion.

Listing 5.6: Passive Aggressive Sample Code

```
from sklearn.linear_model import PassiveAggressiveClassifier
clf = PassiveAggressiveClassifier()
clf.fit(train_feature,train_target)
predicted = clf.predict(test_feature)
```

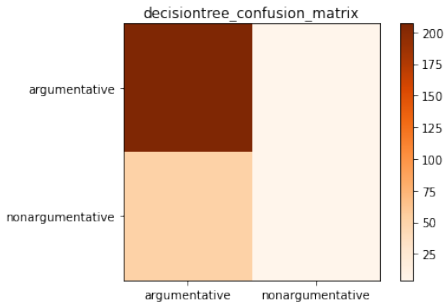
First we use the confusion matrix to visualize the performance of different algorithms (random_state is set to 1). For argument classification, we found that Passive Aggressive is not performing as well as the other three algorithms through the confusion matrix as the Figure 5.2 shows. As for attack classification, as the Figure 5.3 shows, generally, Passive Aggressive performs even worse than on the argument classification.

We use Table 5.1 and Table 5.2 to compare the F score of the above algorithms and experimented with different random_state in argument classification.

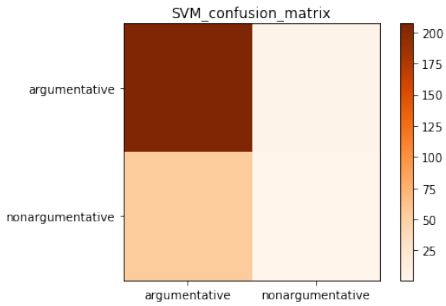
Table 5.3 and Table 5.4 are used to compare the F score of the above algorithms and experimented with different random_state in attack classification.

In Table 5.1, Table 5.2, Table 5.3, the accuracy and precision of MultinomialNB are equal, this is because True Negative and False Negative in the classification are 0.

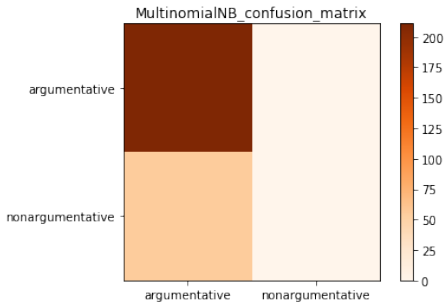
Combining F score from multiple tables, we find that Multinomial Naive Bayes performs the best and Passive Aggressive the worst in our classification experiments.



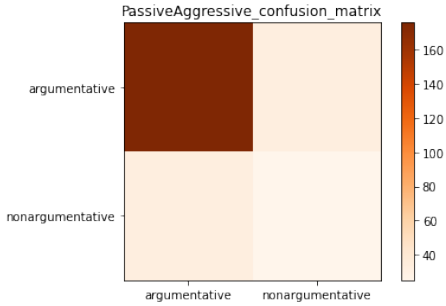
(a) Decision tree



(b) SVM



(c) MultinomialNB



(d) Passive Aggressive

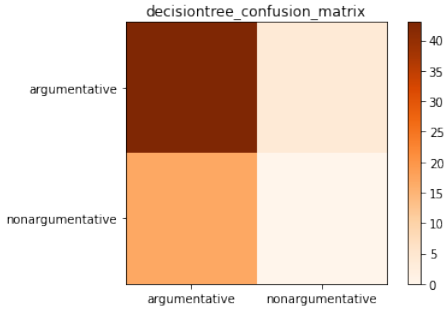
Figure 5.2: Confusion Matrix Visualization for Argument Classification

Table 5.1: Result of Argument Classification(random_state=1)

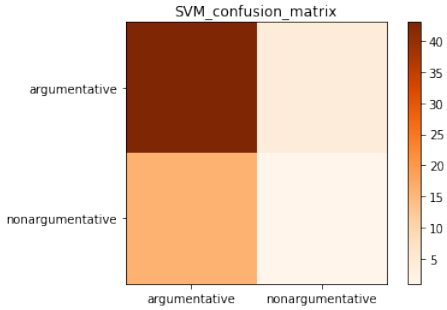
	decision-tree	SVM	MultinomialNB	PassiveAggressive
accuracy	0.7902	0.7790	0.7903	0.7903
precision	0.7992	0.7901	0.7903	0.8172
recall	0.9810	0.9810	1	0.7204
f-score	0.8809	0.8758	0.8829	0.7658

Table 5.2: Result of Argument Classification(random_state=500)

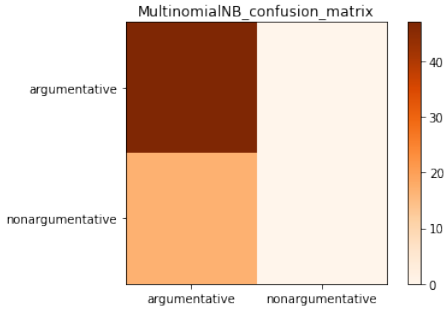
	decision-tree	SVM	MultinomialNB	PassiveAggressive
accuracy	0.7603	0.7460	0.7753	0.7341
precision	0.7761	0.7769	0.7753	0.7720
recall	0.9710	0.9758	1	0.9324
f-score	0.8627	0.8651	0.8734	0.8447



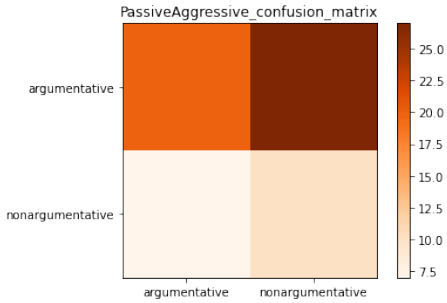
(a) Decision tree



(b) SVM



(c) MultinomialNB



(d) Passive Aggressive

Figure 5.3: Confusion Matrix Visualization for Attack Classification

Table 5.3: Result of Attack Classification(random_state=1)

	decision-tree	SVM	MultinomialNB	PassiveAggressive
accuracy	0.6719	0.6875	0.7344	0.4688
precision	0.7167	0.7288	0.7344	0.7407
recall	0.9149	0.9149	1	0.4255
f-score	0.8038	0.8113	0.8469	0.5405

Table 5.4: Result of Attack Classification(random_state=500)

	decision-tree	SVM	MultinomialNB	PassiveAggressive
accuracy	0.6094	0.6250	0.6563	0.5313
precision	0.6429	0.6491	0.6508	0.6897
recall	0.8780	0.9024	1	0.4878
f-score	0.7423	0.7551	0.7885	0.5714

5.2 Evaluation on Argumentation Graph

As we mentioned in 2.6, we use ASP queries for reasoning problems within valued-based argumentation frameworks, and the implementation approach is proposed by by Uwe Egly, Sarah Gaggl, and Stefan Woltran [20].

In the weighted argumentation frameworks we proposed, we have defined a weighting function $W : V \rightarrow R$ which maps the reliability of a tweet t, denoted as r_t, to a logarithmic scale which is $W(t) : log(3r_t + 2)$. When we calculate weights for different topics, the value of r_t is calculated in cumulative form, i.e., when the Twitter user's profile, icon, and header are related to the topic being calculated, r_t is incremented by one. In addition, when calculating topics do not require professional knowledge, we consider the age factor because older people have more life experience. Also, according to our dataset, order people tend to look at problems from a realistic perspective, i.e. they use more evidence to support their arguments. When the age is older, r_t is increased by one, and young people have less experience therefore the r_t is 0.

We refer to the report classification of some news websites and divide our tweets dataset into the following 6 topics: entertainment, life, politics, love, economy, and ASD (Autistic Spectrum Disorder). Among them, life and love topics do not require professional knowledge, and the age factor is considered to calculate the weight.

We use clingo to run ASP program for love topic and the results are as follows.

Listing 5.7: Answer Set Programming Evaluation Result

```
Answer: 1
in(b14) in(b36) in(b48) in(b49) in(b50) in(b53) in(b54) in(
    b55) in(b56) in(b59) in(b60) in(b62) in(b63) in(b64) in(
    b70) in(b71) in(b74) in(b180)
SATISFIABLE

Models : 1
```

The answer is the solution of for love topic, i.e., the set of acceptable tweets. In addition, we also visualized the framework we modeled in 3.3, shown in Figure 5.4. Tweets with heavier weights have thicker arrows, representing higher reliability.

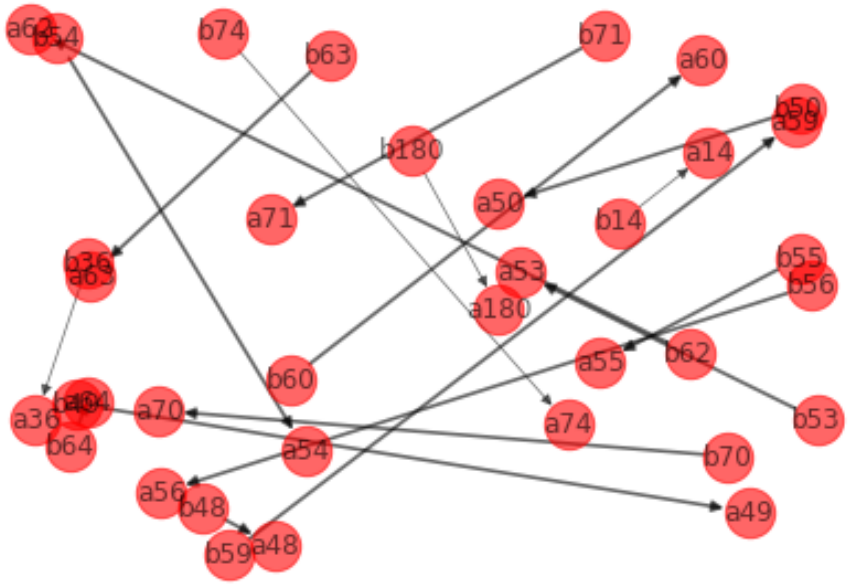


Figure 5.4: Visualization Analysis for Argumentation Graph

Chapter 6

Conclusion

6.1 Concluding Remarks

1. The structure of private opinion.

   - Morphological Analysis Aspect
     In 4.2 we use TF-IDF to calculate the select the features which represent arguments or attacks from private opinion. And we list them in Table 6.1 and Table 6.2 respectively.
   - Syntactic Analysis Aspect
     We use GiNZA [22] for syntactic analysis. Syntactic analysis is the process of analyzing a string of symbols, in our case is Japanese language, conforming to the rules of a formal grammar. We use displaCy from spaCy [23] to visualize one example of syntactic analysis, shown in Figure 6.1.

     Dependencies in both front and back directions are handled in

Table 6.1: Common Phrases in the Arguments of the Private Opinion

arguments	TF-IDF
思っ	0.228171
違う	0.228171
なら	0.192242
意思	0.186046
べき	0.181209
だけ	0.178155
に対して	0.172563
ので	0.159449

Table 6.2: Common Phrases in the Attacks of the Private Opinion

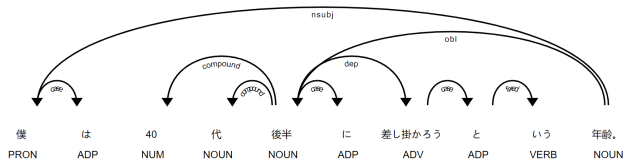
attacks	TF-IDF
けど	0.332797
でも	0.302313
なく	0.299042
ない	0.280712
違う	0.263279
別に	0.24712
おかしい	0.245563
のに	0.18015

units of tokens corresponding to morphemes, and by assigning labels to the dependencies, grammatical relationships such as the subject and object are output. A dependency is a binary asymmetric relationship between a head word and its subordinates. The head word of a sentence is usually verb, and all other words either depend on the head word or are associated with it through dependency paths. The dependency structure is a labeled directed graph. The arrow points from the central word to the subordinates. Specifically, the arrow points from the head to the child. As can be seen from Figure 6.1, each token has only one head. In natural language processing, a word, a punctuation mark, a space, etc. are called a token. In our case, token refers to a morpheme.

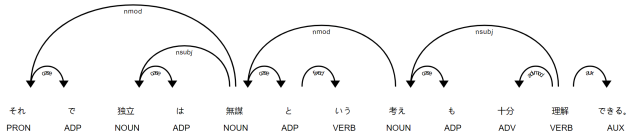
Relation tokens are used to represent subordinate grammatical functions. Commonly used tokens are: root (center word, usually verb), nsubj (nominal subject), prep (preposition), nmod (nominal modifier), advmod (advmod (adverbial modifier)), det (qualifier), amod (adjective modifier), case (case marking), obl (oblique noun), fixed (fixed multiword expression), dep (dependent), aux (auxiliary), acl (clausal modifier of noun), nummod (numeric modifier), advcl (adverbial clause modifier), ccomp (clausal complement).

In this example, We divided a tweet into 5 parts for better visualization analysis for private opinion.

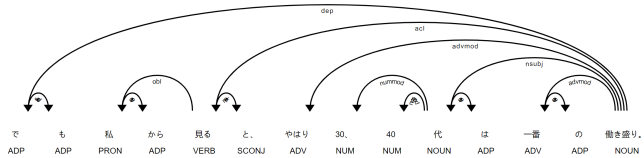
The Twitter user first stated his background, and affirmed that he, who is also in an unoptimistic situation, is not a rational behavior to make a certain action. Then from a negative point of view, it gives a negative opinion and provides evidence. In this expression, there are 3 phrases that we selected as features, “思う”, “こと”,



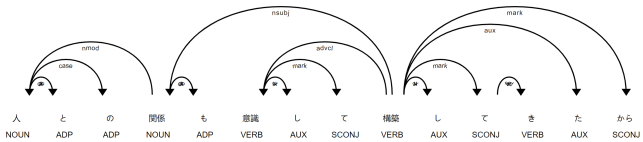
(a)



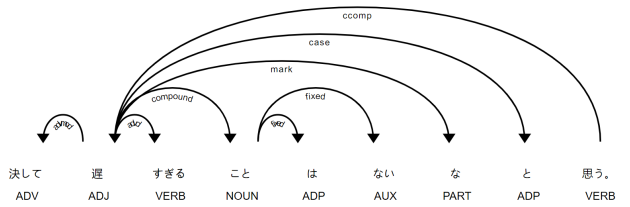
(b)



(c)



(d)



(e)

Figure 6.1: Syntactic Analysis for Private Opinion Arguments

"それで". At the same time, other lexical expressions appearing in the private opinion classified as argumentative may also serve as references for further experiments, which we will discuss in 6.2.

2. Algorithm performs best in classification.
   According to 5.1, we can conclude that the best performing algorithm is Multinomial Naive Bayes.

3. The contribution of argumentation graph done in recognizing reliable tweets. Specifically, we can further normalize the reliability so that it is in the [0,1] interval, the user can set the threshold of the reliability, and the search results will show the arguments higher than the set threshold. When users want to make a decision, argumentation graph can be referred to. Namely, argumentation graph can make a great contribution to information retrieval.

6.2 Future Work

In the experiment of this study, the overall performance of attack classification is worse than that of argument classification. Part of the reason is that the attack data set is smaller than the argument data set (because the attack relation only exists between arguments). As we stated in 6.1, in addition to the features we choose, there are also other symbols that may be worthy of being used as judgment markers for the private opinion arguments. Similarly, in addition to the features we choose, the private opinions classified as attack also appear other markers that may be worth judging as private opinion attacks. Through such repeated experiments, we can expect the improvement of F score and other indicators. However, the above experiments have not been carried out in this study. To trace back to the source, our criteria for selecting features are according to the value of TF-IDF and referring to some relevant Japanese grammar papers. If the above repeated experiments are carried out, whether the True Positive in estimated results of classification are really the True Positive in actual results remains to be discussed. This problem needs the support of more authoritative Japanese experts.

Secondly, feasible research in the future includes making a more complete information retrieval system, including but not limited to adding a more beautiful and easy-to-operate interactive interface, adding evaluation indicators from other users such as Net Promoter Score, using the threshold setting we mentioned in 6.1, in order to provide users in need with a reliable decision-making reference.

Finally, one of the objectives of this study is to visualize the structure of private opinion, as we summarized above. When people communicate in private opinion, because they speak the truth, they think each other's words are reliable compared with those public sound. We hope that future research can be extended beyond private opinion and public sound. When introducing professional knowledge in a certain field to people in other research fields or without background knowledge, what kind of expression and conversation structure can be used to better communicate effectively. If we can figure it out, we can expect that it will make a great contribution to the field of education.

Bibliography

[1] "10 Twitter Statistics Every Marketer Should Know in 2021", [Online]. Available:https://web.archive.org/web/20210608183202/https://www.oberlo.co.uk/blog/twitter-statistics (Accessed by 2021.1.25)

[2] Twitter Developer Platform Tutorials Document, [Online]. Available: https://developer.twitter.com/en/docs/tutorials (Accessed by 2022.1.30)

[3] "Bosses Want to See Explainable AI",[Online].Available:https://www.computerweekly.com/news/252462403/Bosses-want-to-see-explainable-AI (Accessed by 2019.4.30)

[4] Freddy Lecue. *On The Role of Knowledge Graphs in Explainable AI*, SWJ (2019)

[5] Mihai Dusmanu, Elena Cabrio, and Serena Villata. Argument Mining on Twitter: Arguments, Facts and Sources, *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing*, (2017)

[6] Frans H. van Eemeren and Rob Grootendorst. A Systematic Theory of Argumentation: The Pragma-dialectical Approach, *Cambridge University Press*, pp. 1–4 (2004)

[7] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, *Artificial intelligence*, vol.77, no.2, pp.321–357 (1995)

[8] Iyad Rahwan. Mass argumentation and the semantic web, *Web Semantics: Science, Services and Agents on the WorldWide Web*, pp. 29–37 (2008)

[9] Jamal Bentahar, Bernard Moulin, and Micheline Bélanger. A taxonomy of argumentation models used for knowledge representation, *Artificial Intelligence Review*, pp. 211–259 (2010)

[10] Trevor J.M. Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks, *J. Log. Comput*, vol.13, no.3, pp.429–448 (2003)

[11] 伊集院郁子. 意見文における譲歩構造の機能と位置―『確かに』を手がかりに―, アカデミック・ジャパニーズ・ジャーナル, pp. 101–110 (2010)

[12] Twitter OAuth 2.0 Documentation Homepage,[Online].Available:https://developer.twitter.com/en/docs/authentication/oauth-2-0 (Accessed by 2022.1.30)

[13] cURL Documentation Homepage,[Online].Available:https://curl.se/docs/faq.html#What_is_cURL (Accessed by 2022.1.30)

[14] Base64 Wikipedia Homepage, [Online].Available:https://en.wikipedia.org/wiki/Base64 (Accessed by 2022.1.18)

[15] 田代ひとみ. 中級日本語学習者の意見文における論理的表現, 横浜国立大学留学生センター教育研究論集, pp. 131–144 (2007)

[16] 伊集院郁子. 日本人大学生の意見文における「譲歩」の論理性, 東京外国語大学留学生日本語教育センター論集, pp. 35–51 (2014)

[17] Oana Cocarascu and Francesca Toni. Detecting deceptive reviews using Argumentation, *Proceedings of the 1st International Workshop on AI for Privacy and Security*, pp. 1–8 (2016)

[18] MeCab Homepage,[Online].Available:https://taku910.github.io/mecab/ (Accessed by 2022.1.30)

[19] StandardScaler Document Homepage, [Online].Available:https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html#sklearn.preprocessing.StandardScaler (Accessed by 2022.1.30)

[20] Uwe Egly, Sarah Gaggl, and Stefan Woltran. Answer-Set Programming Encodings for Argumentation Frameworks, *Argument and Computation*, pp. 147–177 (2010)

[21] Potassco Homepage,[Online].Available:https://potassco.org/(2022.1.30)

[22] GiNZA Homepage,[Online].Available:https://github.com/megagonlabs/ginza (Accessed by 2022.1.30)

[23] displaCy Document Homepage,[Online].Available:https://spacy.io/usage/visualizers (Accessed by 2022.1.30)