Detecting arguments and their positions in experimental communication data

Hendrik Hüning[*] *Lydia Mechtenberg*[*]
Stephanie W. Wang[†]

March 8, 2022

Abstract

This paper applies argument mining techniques to political discussions obtained from an online chat experiment prior to a ballot. We investigate the task of automatically detecting chat messages that give justification for an underlying claim. We use bag-of-words features as well as state of the art embedding models to train different classifiers on the given task. Moreover, we automatically detect each argument's position and compare our method to a method based on word scores proposed in the political science literature. Our results show that argument mining techniques can be successfully applied to experimental communication data and that it can replace labor-intensive manual annotations by human coders.

Keywords: Chat experiments, argument mining, communication, natural language processing, machine learning

JEL: C88, D83

[*]Department of Economics, Hamburg University, Von-Melle-Park 5, 20146 Hamburg, Germany, Email addresses: hendrik.huening@uni-hamburg.de and lydia.mechtenberg@uni-hamburg.de.

[†]Department of Economics, University of Pittsburgh, 230 South Bouquet Street, Pittsburgh, PA 15260, USA, Email: swwang@pitt.edu.

We are grateful to Sophia Schulze-Schleithoff, Miriam Hinternesch and Sofia Schnitzler for excellent research assistance and the WISO lab of Hamburg University for the outstanding technical assistance. This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 822590. Any dissemination of results here presented reflects only the authors' view. The Agency is not responsible for any use that may be made of the information it contains.

1 Introduction

The analysis of communication has gained interest and importance in the economic literature in recent years (Penczynski 2019). This is at least partly driven by today's role of online debates for the exchange of ideas, opinions, and information. Anonymous strangers and friends alike engage in discussions on political or societal issues in online forums, on social media platforms, or via messenger services such as WhatsApp. This raises a number of questions: How important are such debates in shaping individual opinions? What determines the persuasiveness of arguments? How, if at all, do online debates affect individual and collective decision-making such as voting? In order to be able to answer questions like these, large textual data sets have to be evaluated. The manual coding and analysis of such texts can be time- and labour-intensive and, for large data sets, highly uneconomical.

This paper investigates the usefulness of argument mining techniques to automatically detect argumentative content in experimental communication data. Our communication data stem from online chat discussions about the political topic rent control prior to a ballot. As these discussions are on a specific topic, we investigate context-specific argument detection. Analysing online chat discussions with argument mining techniques is a challenging task because of the brevity of messages, unusual usage of punctuation, fragment sentences and the influence of spoken and face-to-face communication for expressing sentiment (Schabus et al. 2016).

The emerging field of argument mining investigates the possibility of automatically detecting argumentative content in natural language text and therefore provides a promising approach for the analysis of online debates. Despite the fact that this is a young research field, a lot of different approaches and subtasks have already been evaluated. While some research investigates argument-component identification, i.e. detecting parts of text or sentences that are argumentative Mochales and Moens (2011), other research implements a more fine-grained approach by analyzing the argumentative structure (Cabrio and Villata 2012, Peldszus 2014) or the relationship between claims and premises. Argument mining techniques have been applied to a variety of different document types such as legal texts, Wikipedia or newspaper articles, user comments, online product reviews or social media texts. Lippi and Torroni (2016) as well as Cabrio and Villata (2018) provide comprehensive overviews of the literature in terms of methods used and fields of application.

Although there is some work from the computational linguistics field that applies argument mining to (online) discourse (Lawrence and Reed 2017, Lugini and Litman 2018), none of the existing contributions investigates communication within an experiment explicitly designed to analyze the exchange of arguments in online chats prior to a real (individual or collective) decision, e.g. a vote.[1] In other words, communication can be of direct consequence for a real-world political decision in our case. Our experimental chat data offer a fruitful opportunity to study argumentation in a controlled environment.

With regard to the experimental economics literature, automated approaches for the analysis of chat data are scarce. Penczynski (2019) evaluates the usefulness of machine learning and natural language processing techniques for experimental chat data. He studies human versus algorithmic classification of intra-team communication in various game-theoretical settings. He finds that out-of-sample predictions from an algorithm trained with bag-of-words features can replicate human classification of reasoning in chat messages fairly well. More recently, Tebbe and Wegener (2022) perform a horse race between different machine learning algorithms to classify chat messages into informative and non-informative messages. They also investigate the performance of the classification method with regard to training set size and find that for small data sets, training sets of about 30% of the overall data sets can yield good results already.

We contribute to this literature by investigating the usefulness of machine learning and language models for argument detection in experimental chat data. None of the previous studies uses embedding models that are frequently used in the computational linguistics literature to classify sophisticated concepts such as arguments in natural language text. Our findings suggest that structural features, i.e. punctuation, are poor predictors of argumentative reasoning in chat data. Moreover, features obtained from language models not generally outperform a simple bag-of-words approach. Rather unstructured textual data such as chats not necessarily benefit from the strength of language models that take into account contextual knowledge of words.

The remainder of the paper is structured as follows: Section 2 presents our data. Section 3 provides details of the methods used. Our results are summarized in Section 4. Section 5 concludes.

2 Data

We collected our data through an online survey experiment that was conducted in two waves around the Local Rent Control Initiative ballot on the

[1]Other research applying argument mining to (online) discourse include Abbott et al. (2011), Biran and Rambow (2011), Yin et al. (2012), Ghosh et al. (2014), Swanson et al. (2015), Oraby et al. (2015), Addawood and Bashir (2016) and Habernal and Gurevych (2017).

6th of November 2018 in California. On that day, citizens of California could vote in favour or against a proposition that expands local governments' authority to enact rent control in their communities. In the online survey, 1560 participants answered questions related to rent control. At the end of the survey, approximately half of participants had the chance to exchange opinions and arguments in a chat. Two of the survey questions asked subjects to formulate an argument (a) in favour of and (b) against rent control (free text). Answers to these free-text box questions allow us to collect a large amount of possible arguments on the topic. This is our first type of textual data input. To ensure data independence, we used only text-box messages of those participants who subsequently did not participate in the chat. This leaves us with 817 participants and 1634 (potential) arguments.

Our second type of textual data are the chat-messages themselves. At the end of the survey, 743 participants were randomly assigned to chat groups of five in which they could discuss the pros and cons of rent control. This resulted in 264 chats. These chats lasted on average 10.7 minutes and created 6415 messages. The chat environment was similar in design to WhatsApp, a chat platform supposedly familiar to most of our subjects.

For the classification task of detecting arguments in the chat discussions, textual data from both textbox messages (1634) and chat-messages (6445), are used. In total, our text corpus comprises 8079 messages. Inspecting the corpus, we find that, as expected, the data set is less structured compared to other forms of natural language text such as newspaper or scientific articles. We therefore expect structural features such as punctuation to play a lesser role in detecting argumentative content in the chat messages. On the other hand, the chat environment in this online survey experiment was clearly designed for a vital debate about rent control. Results from an analysis of this data allow a better understanding of the effects of debates in chat environments such as on WhatsApp, Facebook Messenger or others.

3 Methods

In this section, we first introduce our labeling scheme for the textual data. Second, we detail the text mining and natural language processing (NLP) techniques that are used to arrive at numerical representations of the textual data. Third, we outline the classification procedures that use the labels and numerical representations as inputs to learn arguments on rent control.

3.1 Labeling Scheme

Our unit of analysis is a message. We choose the message level as our unit of interest and not sentences as punctuation is not used by all chat participants in the same frequency making it hard to split the text into sentences. We manually labeled 2299 chat messages and all 1634 text-box messages as either containing an argument or not. We use the so-called claim-premise model as the underlying argumentation theory (Toulmin 1958, Walton 2009). The claim constitutes a statement or position of a person on a certain topic. The premise supports the claim by providing evidence or justification for the claim. As Rinott et al. (2015) point out, the existence of a premise is crucial for an argument being persuasive.[2] We label each message as containing a premise that supports or attacks the underlying claim. The following examples illustrate our labeling scheme (Compare Table 1):

"Rent control is good because it will lead to more affordable housing."

The first part *"Rent control is good"* constitutes the claim, while the second part *"it will lead to more affordable housing"* establishes the premise. The word *because* functions as a discourse connector between the claim and the premise Lawrence and Reed (2015). In this case, both claim and premise are present and we label this message as containing an argument. In many instances, however, the claim is not explicitly stated. This happens quite frequently in both, the text-box and chat data.[3] Consider the following example:

"It would lead to higher rental prices in the long run."

In this case, the claim is implicit (*Rent control is bad*). This message is labeled as containing an argument although the claim-part of the argument is missing. In cases, however, where only the claim is stated (e.g. *"Rent control*

[2]Manually labeling the text-box messages was necessary because they sometimes did not contain an argument. Some participants stated that they did not recall any argument or wrote something else besides an argument (this occurred in 17 percent of the cases). In contrast to Rinott et al. (2015), we do not distinguish between different types of justifications (evidence such as: study, expert, anecdotal).

[3]In fact, it turns out that the majority of the arguments formulated in the chat are implicit, i.e. not containing the underlying claim. This highlights the special character of the chat environment, where a participant might have stated a claim at the beginning of a chat discussion and justification thereof appear later during the discussion. Wojatzki and Zesch (2016) discuss the problem of implicit argumentation especially in informal settings and propose a possible solution. Since claims are not as frequently used, we do not use them as a separate class in the classification task.

is not a good idea"), the message is labeled as not containing an argument. Moreover, all messages that are off-topic, e.g. contain self-introductions to other members of the chat group, are labeled as not containing arguments. It is important to note that we do not distinguish between arguments for-

Table 1: Examples of labeling scheme

Example	**Type**	**Labeling**
"Hi there, how are you?"	None	No Argument
"Rent control is not a good idea."	Claim only	No Argument
"Rent control is good because it will lead to more affordable housing."	Claim plus premise	Argument
"It would lead to higher rental prices in the long run."	Premise with implicit claim	Argument

mulated on rent control in general and those that specifically address the ballot's proposition. Interestingly, the majority of participants discuss rent control in general rather than arguing about the (in)appropriateness of the proposition itself. As the topic rent control is highly polarized in the USA, we have opinionated text Indurkhya and Damerau (2010), where participants express strong opinions in favour of or against rent control.

Three trained coders annotated the data set independently. In total, 3933 textbox- and chat-messages were labeled. We used only those 3193 messages, where all three coders agreed and discarded the rest. 1415 (44%) of these were labeled as containing an argument and 1778 (56%) as not containing an argument. Unweighted Cohen's kappa and Krippendorff's alpha for the labeling procedure are 0.75 and 0.75 respectively, indicating substantial agreement among coders.

3.2 Numerical Representations of Text Data

The manual labels serve as the first input for our classification procedure. As a second input, we need numerical presentations of the textual data that efficiently carry the information of each particular message. We implement four approaches. First, we construct bag-of-words features that represent each textbox- and chat-message by the frequency and type of words that appear in it. We define this "conventional" approach of representing natural language text as our benchmark case. Second, we implement pre-trained context-free word embeddings for each word of the corpus. These word embeddings are aggregated on the message level and used as features. Third, we train an own embedding model to obtain word embeddings that are specific to our data set. Fourth, we use the state-of-the-art language model architecture of BERT to calculate contextual embeddings for each message. The concept of embeddings is explained below.

3.2.1 Bag of Words (BOW)

Before we construct the bag-of-words features, we pre-processed our text corpus by removing special signs such as #,*,>, removing numbers and changing all text to lower case. Stopwords and punctuations are not removed as they can be highly informative about whether a message contains an argument or not.

Lexical features: As common in the literature, we use n-gram features: We calculate all unigrams and bigrams that occur in our text corpus at least ten times.[4]

Statistical features: As statistical features for each message we consider the length of the message (in characters and words) and the average word length measured by the average number of syllables per word. These features capture the complexity of the message. We hypothesize that in our text corpus of rather short messages, the longer or more complex a given chat message, the more likely it contains complex reasoning, e.g. a formulated argument. This is particularly true in our context of the experimental chat environment. Participants mostly write short messages to introduce themselves to each other, to state their opinion, or to state (dis)agreement with other chat participants. More elaborate messages are more likely to contain arguments on the topic rent control or with regard to the ballot's proposition.

Structural features: This feature-set comprises the number of dots, question marks, exclamation points and commas in each sentence. Although these features performed well on textual data from persuasive essays and Wikipedia articles (Aker et al., 2017), we expect them to play at most a minor role in the context of chat messages because punctuation is less frequently and formally used in this environment than in other forms of natural language text.

Syntactical features: Based on part-of-speech tagging (POS), the number of nouns, verbs, pronouns, adjectives, adverbs etc. of each message are

[4]We also experimented with specific keyword lists as input features such as argumentative connectors like *because*, *since* etc. (Lawrence and Reed 2015). Since some of these connectors are already covered by the n-gram features and others are rarely used, we abstain from including additional keywords as features. Moreover, since the majority of messages contain premises where the claim is implicit, argumentative connectors are not as frequently used in our data set as in more structured data sets.

used as features.

Morphological features: Finally, we calculate the frequency of morphological features in each message of the corpus.[5] The morphological features considered include abbreviations, grammatical case, definiteness, degree (positive, comparative, superlative), gender (neutral, fem., masc.), mood (indicative, imperative), number (singular or plural), numeral type (cardinal, ordinal, multiplicative), person (first person etc.), personal or possessive personal pronouns (my, mine, yours etc.), reflexive (does the word refer to the subject of the message or not), tense (past or present), verbform (finite, non-finite etc.), voice (active or passive), foreign (word from other language) and typo (misspelling detected).

Since our corpus is only loosely structured, we use many features on the token-level. In total, we arrive at 1296 bag-of-word features. In order to reduce the number of features used for classification, the information-gain criterion is used as a classifier-independent feature selection technique.[6] All features with an information gain of zero in the training set are not considered for the classification task.

3.2.2 Embedding Models

A breakthrough in natural language modeling is the concept of word embedding models. The idea is that each word gets represented by a numerical vector of the same length that is estimated from the word's embedding, i.e. the words surrounding it. In other words, the semantic meaning of a word is estimated by the context it is usually used in. Semantic similarities and dissimilarities of words can be analysed by the relative position of vectors in the vector space. We implement three state-of-the-art embedding techniques.

Google news word vectors: As a first approach, we use word embeddings that were trained on the large Google News text corpus (about 100 billion words). These embeddings are freely available online for reuse.[7] These word vectors were trained with Word2vec that uses a skip-gram neural network to predict a word from its context (Mikolov et al. 2013). In

[5]POS-tagging and morphological features are obtained with the udpipe implementation in R, see https://cran.r-project.org/web/packages/udpipe/vignettes/udpipe-annotation.html.

[6]The R-package FSelector Romanski and Kotthoff (2018) is used. This feature selection follows Addawood and Bashir (2016) among others. This feature selection technique has the advantage of reducing the number of features used for classification substantially. The disadvantage is that it might remove features that are only useful in combination with other features. Since our labeled data set is quite small, we prefer a parsimonious model with not too many features.

[7]See https://code.google.com/archive/p/word2vec/.

other words, the algorithm predicts for each input word the probability of the words surrounding it, i.e. the probability of the "nearby words".

The Google news data set contains embeddings of 3 million words. For our purpose, we extract those word vectors that are part of the vocabulary of our text corpus, i.e. 1492 word vectors. In order to obtain a vector representing a message in our corpus, we average the word vectors of all words of a particular message. In order to account for the importance of a word for a message, each word vector is weighted by *tf/idf*. The *tf/idf* weight is the frequency of the word in the particular message (term frequency=tf) divided by the frequency of the word in the overall corpus (inverse document frequency=idf). Words that appear regularly across all messages are downgraded and words that appear rarely are upgraded because they are particularly informative if they appear in a message. Moreover, the weight regularizes the length of messages ranging form one word to 150 words. Since the Google News word vectors have 300 dimensions, we obtain 300 features that are used in the subsequent classification task.

Global vectors (GloVe): As a second approach, we use word embeddings that are explicitly trained on our corpus. We do so since many pre-trained word vectors that are available online, such as the Google news vectors, are trained on data sets that are not comparable with the chat data we are analysing. Language may be used differently in "short-message-contexts" such as social media and chats (Liu et al. 2017). We use GloVe for this approach (Pennington et al. 2014). GloVe also produces embeddings of words but follows a different optimization approach than Word2vec to obtain word embeddings. GloVe uses single value decomposition on the full word co-occurrence matrix that is built from the corpus to arrive at low-dimensional word vectors.

We use the GloVe implementation in R to learn word embeddings that are specific to our text corpus. Our chosen window size is five, i.e. five words before and after the word in question are considered for calculating its embedding. Moreover, the chosen vector-size is 300. As suggested in the GloVe model, we sum up the main and context component. As before with the Google News word vectors, we average the *tf/idf*-weighted word vectors of each message. Since we choose vectors to have 300 dimensions, we obtain 300 features that can be used in the subsequent classification task.

BERT: The word embeddings obtained with Word2vec and GloVe share the disadvantage that each word in our corpus is represented by a fixed vector. This is problematic for words that have a different meaning depending on the context they are used in. The most obvious example are polysemous words. Words such as "train" have a different meaning depending on their context. The Bidirectional Encoder Representations from Transformers model, shortly BERT, overcomes this problem and allows words to have a different embedding depending on the context they are used in (Devlin et al. 2018). Moreover, BERT allows to calculate embeddings directly for entire sentences or messages as in our case that represent the semantic meaning of that message. BERT outperforms many other language models on a variety of classification tasks such as those defined in the General Language Understanding Evaluation (GLUE) benchmark (Wang et al. 2018).

We use the pretrained-BERT model (base-uncased-model) and apply it to the vocabulary of our text corpus.[8] We access its 12th output layer that contains the embeddings for each message of our text corpus. These embeddings have 768 dimensions, i.e. we obtain 768 features to use in the subsequent classification task.

3.3 Classification Procedure

All four feature sets, namely bag-of-words, Google News embeddings, GloVe embeddings and BERT embeddings are fed separately into four different classifiers to predict messages containing an argument or not. For the classification task, we split our data set randomly into a training set (80%) and test set (20%). After splitting the data into training and test set, features in the bag-of-words feature set are scaled to the range between 0 and 1 because some classification algorithms are sensitive to features that are defined on different scales. Four classifiers are trained on the training set to distinguish argumentative and non-argumentative messages. We use four classifiers that were frequently used in previous research on argument mining and proven to be suitable for the task (Lippi and Torroni 2016). These include Logistic Regression (LR), Support Vector Machine with linear Kernel (SVM), Neural Network with a single hidden layer (NN) and Random Forests (RF). As a benchmark model, we train these classifiers on bag-of-word features. Subsequently, we train the classifiers with the sentence embeddings obtained from the three language model classes and compare the performance with that of the bag-of-words approach.

All results are obtained by performing stratified k-fold cross-validation. In cross-validation, the training set is randomly split in k equally sized folds. In our case k equals 10. For each fold, the classifier is trained on all other $(k-1)$ folds and evaluated on fold k. This is repeated for all k folds. Cross validation tests the generalization ability of the model within the training phase and ensures that a prediction for a particular message is solely based

[8]We use the implementation of BERT in R provided by Johnathan Bratt, see: https://github.com/jonathanbratt/RBERT.

on training of other messages. Stratification ensures that the share of classes in the original data set, i.e. the share of arguments versus non-arguments, is represented in each of the $k = 10$ folds. Moreover, for the Neural Network and Random Forest classifiers, 10 randomly drawn hyper-parameter were tested in each fold. The hyper-parameter that yielded the best overall accuracy was chosen for the final model.

As performance measures of our classification task, we report the accuracy, precision, recall and F1 values for all classifier and feature-set combinations that are estimated. Accuracy is defined as the share of correctly identified arguments and non-arguments of all messages in the test set. Precision is defined as the number of arguments identified by the classifier divided by the total number of actual arguments in the test set. Recall is defined as the number of actual arguments identified by the classifier divided by the total number of predicted arguments. F1 is the harmonic mean of precision and recall. F1 is frequently reported in case of imbalanced class distributions. Although we have a rather moderate imbalance between classes, 44% arguments and 56% of non-arguments in the original data set, F1 might be more indicative of the classifiers' performance.

4 Results

4.1 Arguments versus non-Arguments

In the following, we report results of the classification task of identifying arguments in textbox- and chat-messages as defined above. Results are summarized in Table 2. We evaluate the performances using the F1 value defined above.

The bag-of-word approach performs reasonably well. In combination with the SVM the bag-of-words approach achieves a F1 value of 0.91. Word Embeddings used from the Google News database, however, performs slightly worse. Across all classifiers the F1-value is between 0.82 and 0.88. Word vectors that are specifically trained on our data (GloVe) perform even worse in predicting arguments in messages. This result indicates that word vectors trained specifically for our chat-data context do not lead to an improvement in prediction accuracy. In fact, these word vectors cannot sufficiently capture the semantic meaning of words in this small data set. This indicates that for small data sets such as ours obtained from an online experiment, the use of embeddings that are pre-trained on large data sets should be preferred. The benefit of very generic word vectors obtained from large data sets is larger than the benefit from word vectors that are specifically trained for the chat-context.

Table 2: Prediction results - Argument vs. non-Argument

	LR	SVM	NN	RF
Bag-of-words				
Accuracy	0.88	0.92	0.91	0.9
Precision	0.83	0.9	0.87	0.85
Recall	0.93	0.93	0.93	0.94
F1	0.87	0.91	0.9	0.89
Pre-trained (Google) Embeddings				
Accuracy	0.81	0.88	0.89	0.88
Precision	0.72	0.83	0.86	0.87
Recall	0.95	0.92	0.89	0.87
F1	0.82	0.87	0.88	0.87
Trained Embeddings				
Accuracy	0.77	0.84	0.87	0.87
Precision	0.83	0.77	0.83	0.85
Recall	0.6	0.92	0.89	0.85
F1	0.69	0.84	0.86	0.85
Embeddings from BERT (base)				
Accuracy	0.87	0.91	0.92	0.91
Precision	0.87	0.9	0.91	0.89
Recall	0.83	0.89	0.91	0.91
F1	0.85	0.9	0.91	0.9

Notes: We used Logistic Regression (LR), Support-Vector Machine with linear Kernel (SVM), Neural Network (NN) and Random Forest (RF) as the classification algorithms.

Embeddings obtained from BERT in combination with a neural network classifier perform equally well compared to the bag-of-words features in combination with SVM, reaching a F-value of 0.91. It has to be noted, however, that only the BERT Base model was used and no specific fine-tuning on the data set at hand was performed. Further performance gains could be expected with fine-tuning or if BERT Large was used. Overall, the results in Table 2 show two main patterns: First, across the classification algorithms, the bag-of-words feature set performs similarly well compared to the more sophisticated approach using BERT features. This highlights that the classification algorithms might not benefit that much from the contextual knowledge that the BERT features produce. This could be the case because of the unstructured and fragmented chat messages, where contextual knowledge is of less value. Second, there is a slight tendency that more sophisticated classifiers (NN and RF) outperform the simpler ones (LR and SVM). The advantage of a Neural Network and Random Forest is that they are able to also detect and exploit non-linear relationships between the input features that helps to detect the rather sophisticated linguistic concept of arguments in the data.

In order to get an idea of what features drive the performance result, we report feature importance of the bag-of-words approach combined with the Random Forest classifier (Compare Figure 1). Feature importance in a decision tree is defined as how well a feature splits the data in two subsets at a decision node, also called average decrease in impurity. This measure is averaged over all decision trees that are estimated in the Random Forest).

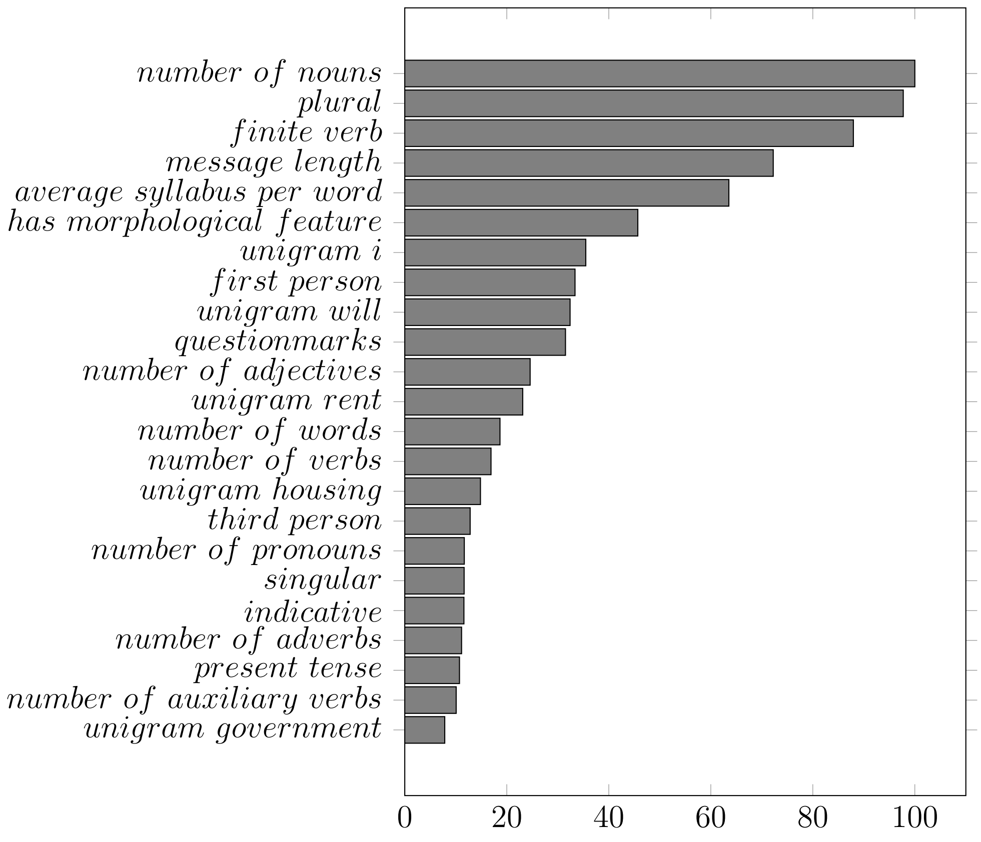
As expected, the length and average number of syllables per word are very informative with regard to an argument being present in the message. Many messages such as self-introductions or stated agreement with other participants are rather short. Thus, the length and complexity of a message are highly indicative of a participant elaborating on rent control and using premises to support or attack an underlying claim. Moreover, messages containing the unigram *will* indicate that participants elaborate on rent control by putting forward a consequence of it being implemented (e.g. *"If rent control is implemented, it will lead to ..."*). As we expected from the inspection of the data, structural features such as dots and commas only play a minor role in all of our estimated models. The exceptions are question marks that mostly occur when participants clarify something during the chat. The communication in the chat is rather informal and is characterised by unstructured and fragment sentences.

Finally, unigrams such as *housing*, *affordable* and *government* illustrate the two positions on the topic rent control. While some argue that it should be implemented because it leads to more affordable housing, others reject rent control because they dislike any intervention of the government in markets (liberty-based arguments). In future work, it would be interesting to investigate if a system is able to detect these different values behind arguments.

4.2 Pro versus Con Arguments

In a second step, we train algorithms to classify chat messages into arguments in favor and against rent control. Only those textbox and chat messages are used for training that were classified as argumentative by the first step (see above). Messages in favor of rent control are labeled as one and those against rent control as zero. Thus, we again perform a binary classification task. We

Figure 1: Feature Importance in RF with BOW-features.



use the same classification techniques from before. For convenience, we only report results using BERT features. Results are depicted in Table 3. The results indicate that the logistic regression classifier performs considerably worse compared to the other classifiers. The Support Vector Machine and the Random Forest perform best with F1-values of 0.76.

Table 3: Prediction results - Pro versus Con

	LR	SVM	NN	RF
BERT (base)				
Accuracy	0.66	0.76	0.75	0.77
Precision	0.65	0.75	0.76	0.79
Recall	0.65	0.76	0.73	0.72
F1	0.65	0.76	0.74	0.76

Notes: We used Logistic Regression (LR), Support-Vector Machine with linear Kernel (SVM), Neural Network (NN) and Random Forest (RF) as the classification algorithms.

Overall, classification of arguments into their positions, i.e. in favor of and against rent control, performs significantly worse than our first step distinguishing argumentative from non-argumentative chat messages. This has two main reasons: First, while the first step was based on 1415 messages labeled as argumentative and 1778 as non-argumentative, this second step only relies on the 1415 messages labeled as argumentative, i.e. the training set size is considerably smaller. Second, while distinguishing chat messages that contain small talk such as *Hi, how are you?* and agreement such as *I agree with participant 4* from argumentative reasoning is relatively easy, it is a lot harder to distinguish arguments in favor from those against rent control. For instance, take the arguments *Rent control will lead to fixed and projectable prices for renters* and *Rent control will lead to fixed prices that cannot fluctuate anymore*. The same argument is used on both sides and only small linguistic differences indicate if it is meant to be an argument in favor or against rent control making it harder for a classification algorithm to detect these positions.

4.3 Comparison to Wordscores model

The previous results indicate that we can distinguish arguments in favor of and against rent control reasonably well. A participant's argumentative position with regard to rent control can be calculated straightforwardly by aggregating her chat messages that contain arguments with the following steps.

First, we use the raw probabilities for each argumentative message being in favor of or against rent control from our second algorithm and multiply those against rent control by -1 (Compare Table A1 for actual chat messages and their probabilities). The sum of these modified raw probabilities measure a participant's average argumentative position on rent control. For instance, a participant formulates three arguments during the chat discussion, two in favor of rent control and one against. Modified raw probabilities of these arguments are 0.6, 0.8 and -0.7. Then, her average argumentative position is 0.7 and positive, i.e. the participant argues more in favor of than against rent control. We denote this as the participant's ArgScore. These ArgScores summarize participants' argumentative policy positions with regard to rent control. A distribution of ArgScores, displayed in the left panel of Figure 2, indicates that while many do not use arguments or have a balanced use of arguments, there are also many participants that have strong argumentative positions in one direction. These scores can be used to investigate participants' persuasiveness in the chat discussions.

Another popular method that extracts policy positions from natural language text is *Wordscores* (Laver et al. 2003). The *Wordscores* model is a supervised scaling method that relies on reference texts where policy positions are known a priori. The model learns relative word frequencies from those reference texts and predicts policy positions for texts where policy positions are unknown.[9] In the following, we compare participants' argument scores we obtained from the argument mining exercise to their policy positions obtained from the Wordscores model.

More specifically, we use our textbox-messages as reference texts and label them as 1 (-1) if they contain an argument in favor of (against) rent control. Before we train the Wordscores model on these labeled textbox data, we lemmatize words in the textboxes and chat-messages, i.e. words are transformed into their lemmas. For instance, the words *works*, *working* and *worked* all become *work*. This ensures that the model learns word frequencies regardless of their inflections which leads to better predictions for the chat-messages. Finally, the trained Wordscores model is used to predict policy positions for the sum of a participant's chat-messages, i.e. everything she

[9]A related method that builds on Wordscores is Wordfish (Slapin and Proksch 2008). While Wordfish does not rely on reference texts, it asks the researcher to identify two texts (from the textual corpus) with opposite policy positions that are used for identification. In our case, however, this method is more difficult to apply as our chat-messages are rather short and results would strongly depend on the choice of the two texts needed for identification.

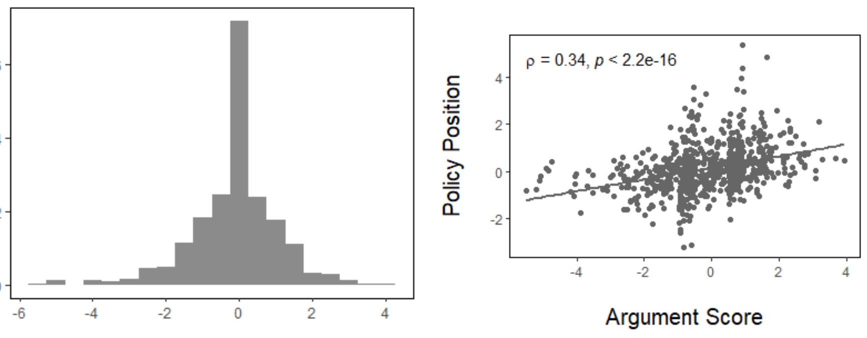
contributed to the chat discussion.

Finally, we compare the policy position scores we obtained from these procedure with the argument scores that we arrived at with the argument mining technique. As the right panel of Figure 2 displays, there is a strong association between argument scores and policy position scores. Pearson's correlation coefficient is 0.34 (p-value: 2.2e-16). Since both methods rely on the same set of information, i.e. textbox data containing arguments in favor of and against rent control and chat messages, it is intuitive to expect some correlation. It is, however, important to stress the difference of these techniques. While the argument mining method precisely concentrates on argumentative structures from contextual knowledge, Wordscores is a bag-of-words (BOW) model that relies on relative word frequencies. For instance, an individual might use many words during the chat discussion that appear more often in the textbox messages with arguments in favor of rent control. Hence, her policy position score is highly positive. This does not mean, however, that she uses arguments. It is totally possible that she repeats frequently her position on rent control without offering supporting arguments.

Although the argument mining technique can provide fine-grained scores for each participant's argument strength, it is remarkable how well the policy positions based on word scores can capture argumentation. Thus, when the right set of reference texts are available, in our case textbox-messages that contain pro and contra arguments of rent control, the Wordscores model can serve as a good proxy for measuring argumentation.

Figure 2: Comparison with policy positions from Wordscores model

(a) Distribution of ArgScores (b) Correlation with policy positions



5 Conclusion

Automated methods for the analysis of experimental communication data are scarce. In this paper, we studied the automated detection of arguments in chat communications collected through an online survey experiment. The results of our classification exercise are encouraging because they highlight that the sophisticated concept of argumentation can be automatically detected in experimental text data using NLP and ML techniques. This is in line with Penczynski (2019), who investigates the usefulness of machine learning techniques for detecting sophisticated reasoning in experimental text data. Especially experimental chat data is challenging for automated coding through machines because of the brevity of messages as well as unstructured and fragment sentences.

Despite this challenge, we can detect arguments in our data reasonably well. Structural features such as the use of dots and commas play a lesser role in identifying messages containing argumentative reasoning. This contrasts with previous findings such as in Aker et al. (2017). All in all, a simple bag-of-word feature approach performs similarly well compared to embeddings obtained from the contextual language model BERT.

Our results highlight that argument mining techniques can successfully be applied to communication data from economic experiments and open up a promising future avenue of empirical research such as on how communication affects economic or voting decisions. The authors use the results of this work in research on the empirical question of how arguments in online discourse affect voting behaviour.

References

Abbott, R., M. Walker, P. Anand, J. E. Fox Tree, R. Bowmani, and J. Kind (2011). Recognizing disagreement in informal political argument. In *Proceedings of the Workshop on Language in Social Media (LSM 2011)*, Portland, Oregon, pp. 2–11.

Addawood, A. and M. Bashir (2016). What is your evidence? a study of controversial topics on social media. In *Proceedings of the 3rd Workshop on Argument Mining*, Berlin, Germany, pp. 1–11.

Aker, A., A. Sliwa, Y. Ma, R. Lui, N. Borad, S. Ziyaei, and M. Ghobadi (2017). What works and what does not: Classifier and feature analysis for argument mining. In *Proceedings of the 4th Workshop on Argument Mining, Copenhagen*, Copenhagen, Denmark, pp. 91–96.

Biran, O. and O. Rambow (2011). Identifying justifications in written dialogs by classifying text as argumentative. *International Journal of Semantic Computing 5*(4), 363–381.

Cabrio, E. and S. Villata (2012). Natural language arguments: A combined approach. In *ECAI*.

Cabrio, E. and S. Villata (2018). Five years of argument mining: a data-driven analysis. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, pp. 5427–5433.

Devlin, J., M. Chang, K. Lee, and K. Toutanova (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *Computer Science abs/1810.04805*.

Ghosh, D., S. Muresan, N. Wacholder, M. Aakhus, and M. Mitsui (2014). Analyzing argumentative discourse units in online interactions. In *Proceedings of the 1st Workshop on Argumentation Mining*, Baltimore, Maryland, pp. 39–48.

Habernal, I. and I. Gurevych (2017). Argumentation mining in user-generated web discourse. *Computational Linguistics 43*(1), 125–179.

Indurkhya, N. and F. J. Damerau (2010). *Handbook of Natural Language Processing*, Volume 2. Boca Raton, Florida: Chapman and Hall/CRC.

Laver, M., K. R. Benoit, and J. Garry (2003). Extracting policy positions from political texts using words as data. *American Political Science Review 97*(2), 311–331.

Lawrence, J. and C. Reed (2015). Combining argument mining techniques. In *Proceedings of the 2nd Workshop on Argumentation Mining*, Denver, Colorado, pp. 127–136.

Lawrence, J. and C. Reed (2017). Using complex argumentative interactions to reconstruct the argumentative structure of large-scale debates. In *Proceedings of the 4th Workshop on Argumentation Mining*, Copenhagen, Denmark, pp. 108–117.

Lippi, M. and P. Torroni (2016). Argumentation mining: State of the art and emerging trends. *ACM Transactions on Internet Technology 16*(2), 10:1–10:25.

Liu, H., Y. Gao, P. Lv, M. Li, S. Geng, M. Li, and H. Wang (2017). Using argument-based features to predict an analyse review helpfulness. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, pp. 1358–1363.

Lugini, L. and D. Litman (2018). Argument component classification for classroom discussions. In *Proceedings of the 5th Workshop on Argument Mining*, Brussels, Belgium, pp. 57–67.

Mikolov, T., K. Chen, G. Corrado, and J. Dean (2013). Efficient estimation of word representations in vector space. *Computer Science abs/1301.3781.*

Mochales, R. and M.-F. Moens (2011). Argumentation mining. *Artificial Intelligence and Law 19*(1), 1–22.

Oraby, S., L. Reed, R. Compton, E. Riloff, M. Walker, and S. Whittaker (2015). And that's a fact: Distinguising factual and emotional argumentation in online dialogue. In *Proceedings of the 2nd Workshop on Argumentation Mining*, Denver, Colorado, pp. 116–126.

Peldszus, A. (2014). Towards segment-based recognition of argumentation structure in short texts. In *Proceedings of the 1st Workshop on Argumentation Mining*, Baltimore, Maryland, pp. 88–97.

Penczynski, S. (2019). Using machine learning for communication classifcation. *Experimental Economics 22*, 1002–1029.

Pennington, J., R. Socher, and C. D. Manning (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, pp. 1532–1543.

Rinott, R., L. Dankin, C. Alzate Perez, M. M. Khapra, E. Aharoni, and N. Slonim (2015). Show me your evidence - an automatic method for context dependent evidence detection. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, pp. 440–450. Association for Computational Linguistics.

Romanski, P. and L. Kotthoff (2018). Fselector: Selecting attributes. `https://cran.r-project.org/package=FSelector`. Accessed: 2020-04-17.

Schabus, D., B. Krenn, and F. Neubarth (2016). Data-driven identification of dialogue acts in chat messages. In *Proceedings of the 13th Conference on Natural Language Processing (KONVENS 2016)*, Bochum, Germany, pp. 236–241.

Slapin, J. B. and S.-O. Proksch (2008). A scaling model for estimating time-series party positions from texts. *American Journal of Political Science 52*, 705–722.

Swanson, R., B. Ecker, and M. Walker (2015). Argument mining: Extracting arguments from online dialogue. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, Prague, Czech Republic, pp. 217–226.

Tebbe, E. and B. Wegener (2022). Is natural language processing the cheap charlie of analyzing cheap talk? a horse race between classifiers on experimental communication data. *Journal of Behavioral and Experimental Economics 96*, 101808.

Toulmin, S. E. (1958). *The Use of Argument.* Cambridge University Press.

Walton, D. (2009). Argumentation theory: A very short introduction. In G. Simari and I. Rahwan (Eds.), *Argumentation in Artificial Intelligence*, pp. 1–22. Boston, Massachusetts: Springer.

Wang, A., A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman (2018). Glue: A multi-task benchmark and analysis platform for natural language understanding. In *2019 International Conference on Learning Representations (ICLR)*, New Orleans, Louisiana.

Wojatzki, M. M. and T. Zesch (2016). Stance-based argument mining - modeling implicit argumentation using stance. In *Proceedings of the 13th Conference on Natural Language Processing (KONVENS 2016)*, Bochum, Germany, pp. 313–322.

Yin, J., N. Narang, P. Thomas, and C. Paris (2012). Unifying local and global agreement and disagreement classification in online debates. In *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis*, Jeju, Korea, pp. 61–69.

Appendix A

Table A1: Chat messages and their classifications

Chat messages (cleaned)	Prob(Arg)	Prob(InFavor)	Prob(Against)	Modified Prob
"what are you guys voting?"	0.04	-	-	-
"go to the polls"	0.07	-	-	-
"voting yes for sure, its a small step"	0.06	-	-	-
"i am voting against this."	0.01	-	-	-
"too much government control is not good for anyone"	0.71	0.28	0.72	-0.72
"landlords will not be able to increase rent prices and set prices at random to who the favor"	0.96	0.75	0.25	0.75
"if there is to much rent control, landlords would not be able to afford to keep maintenance up on the properties."	0.96	0.07	0.93	-0.93
"housing should be a human right, not a privilege."	0.85	0.90	0.10	0.90
"keeps less people from becoming homeless."	0.96	0.51	0.49	0.51

Notes: The table shows cleaned chat messages (column 1) together with the predicted raw probability of algorithm one in column 2 (argument probability) as well as the raw probabilities of algorithm two in columns 3 and 4 (probabilities of argument position, i.e. in favor or against). In both cases, the results of the Random Forest classifier are shown. The modified probability (column 5) multiplies the probability of arguments against rent control by -1.