# Detecting argumentative discourse in online chat experiments

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

March 3, 2021

**Abstract**

This paper applies argument mining techniques to chat data obtained from an online survey experiment with deliberative content. 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 word- and sentence-embedding models to train different classifiers on the given task. In contrast to previous studies, our results indicate that structural features are less important to predict argumentative reasoning in the chat. Moreover, the random forest classifier with features extracted from BERT has the best overall performance in the classification task. Our results show that argument mining techniques can be successfully applied to chat data obtained from economic experiments. It offers the chance to answer empirical research questions such as the effect of deliberation on economic, social, or political behaviour.

**Keywords:** Chat experiments, argument mining, natural language processing

**JEL: C63, 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, Pittsburgh University, 230 South Bouquet Street, Pittsburgh, PA 15260, USA, Email: swwang@pitt.edu.

We are grateful to Sophia Schulze-Schleithoff 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

Nowadays, online debates are an important forum for the exchange of ideas, opinions, and information. Their prevalence raises a number of questions: How important are such debates in shaping individual opinions? What, if anything, determines the persuasiveness of arguments and turns individuals into opinion leaders? How, if at all, do online debates affect individual and collective decision-making? In order to be able to answer questions like these, large textual datasets have to be evaluated. Together with the trend towards larger datasets in experimental economics, this calls for a combination of an experimental and an automated approach to generate and analyze chat messages of a large number of subjects.

This paper investigates the use of argument mining techniques and state-of-the-art language models to detect premises, i.e. the argumentative content, in online chat discussions. As these discussions are on a specific topic, we investigate context-specific premise 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).

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.

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 on argument mining applied to (online) dis-

course (Lawrence and Reed (2017), Lugini and Litman (2018)), none of the existing contributions investigates deliberation 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] While Twitter is arguably more appropriate for information dissemination rather than debating (Smith et al. 2013, Addawood and Bashir 2016), chats allow an immediate response to others' opinions and arguments, and are hence a suitable forum for deliberation and debate. From the perspective of the data set being used, Lugini and Litman (2018) are closest to our approach. They apply argument mining to written transcripts of classroom discussions on text or literature pieces among pupils. Face-to-face discussions among pupils that know each other are, however, quite different from online chat discussions (e.g. on Facebook) or chats on private messenger services such as WhatsApp. Our experimental chat data offer a fruitful opportunity to study online debates on a specific topic in a controlled environment.

With regard to economic experiments, automated approaches for the analysis of chat data are scarce. Only recently 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. Our study is similar in the sense that we also evaluate the task of detecting reasoning, i.e. arguments, in chat data obtained from an online survey experiment. Moreover, we compare the performance of bag-of-words features with features generated through state-of-the-art language models using four frequently used classification algorithms.

Our findings suggest that structural features 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 chat data not necessarily benefits from the strength of language models that take into account contextual knowledge of words and messages.

The remainder of the paper is structured as follows: Section 2 presents our data. Section 3 introduces our labeling scheme, while Section 4 provides details of the feature selection and classification task. Classification results are summarized and discussed in Section 5. Section 6 concludes.

[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), Habernal and Gurevych (2017).

## 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 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 premises 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 Labeling process

Our unit of analysis is a message.[2] We manually labeled 2299 chat messages and all 1634 text-box messages as either containing a premise or not. Man-

[2]Since punctuation is not used by all chat participants in the same frequency, we choose the message level instead of the sentence level for our analysis.

ually labeling the text-box messages was necessary because they sometimes did not contain a premise. Some participants stated that they did not recall any argument or wrote something else besides a premise (this occurred in 17 percent of the cases). In contrast to Rinott et al. (2015), we do not distinguish between different types of premises (evidence such as: study, expert, anecdotal). 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 a premise. 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 a premise although the claim-part of the argument is missing. In cases, however, where only the claim is stated (e.g. *"Rent control is not a good idea"*), the message is labeled as not containing a premise. Moreover, all messages that are off-topic, e.g. contain self-introductions to other members of the chat group, are labeled as not containing a premise. It is important to note that we do not distinguish between arguments formulated 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.

[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. Because claims are not as frequently used, we do not use them as a separate class in the classification task.

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

Table 1: Examples for coding scheme

Two trained coders annotated the data set independently. In total, 3933 textbox- and chat-messages were labeled. 1614 (41%) of these were labeled as containing a premise and 2319 (59%) as not containing a premise. Unweighted Cohen's kappa and Krippendorff's alpha for the labeling procedure are 0.69 and 0.68 respectively, indicating substantial agreement among coders.

# 4 Feature Selection and Classification

For the classification task of detecting argumentative reasoning in the chat messages, we implement four approaches. First, we construct bag-of-words features that represent each textbox- and chat-message. We define this "traditional" approach of representing natural language text as our benchmark case. Second, we implement pre-trained context-free vector representations for each word of the corpus. These vector representations are aggregated on the message level and used as features. Third, we train own vector representations to obtain word embeddings that are specific to our dataset. Fourth, we use the state-of-the-art language model architecture of BERT to calculate contextual vector representations for each message.

## 4.1 Bag-of-words (BOW)

Before we constructed the 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 a premise 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]

[4]We also experimented with specific keyword lists as input features such as argumen-

**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, exclamations 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 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, nonfinite 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.

tative 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 dataset as in more structured datasets.

[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.

In total, this amounts to 1296 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. The remaining 141 features are scaled to a range between 0 and 1.

## 4.2 Embeddings - Vector representations

A breakthrough in natural language modeling is the concept of word and sentence embedding models. The idea is that each word or message 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 vector representation techniques.

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

[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 dataset is quite small, we prefer a parsimonious model with not too many features.

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

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 vector representations that are explicitly trained on our corpus. We do so since many pretrained word vectors that are available online, such as the Google news vectors, are trained on datasets 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 vector representation 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 vector representations 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 vector representations 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 vector representation depending on the context they are used in Devlin et al. (2018). For this reason, BERT belongs to the family of contextual models. It 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

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

contains vector representations for each message of our text corpus. These vector representations have 768 dimensions, i.e. we obtain 768 features to use in the subsequent classification task.

## 4.3 Classification methods

All four feature sets, namely bag-of-words, Google News vectors, GloVe vectors and BERT vectors are fed separately into four different classifiers to predict messages containing a premise or not. For the classification task, we split our data set randomly into a training set (80%) and test set (20%). Four classifiers are trained on the training set to distinguish argumentative and non-argumentative messages. We use 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), Naïve Bayes (NB) 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 on training of other messages. Stratification ensures that the share of classes in the original data set, i.e. the share of premises versus non-premises, is represented in each of the $k = 10$ folds.

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 premises and non-premises, of all messages in the test set. Precision is defined as the number of premises identified by the classifier divided by the total number of actual premises in the test set. Recall is defined as the number of actual premises identified by the classifier divided by the total number of predicted premises. 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, 41% premises and 59% of non-premises in the original data set, F1 might be more indicative of the classifiers' performance.

# 5 Results

In the following, we report results of the classification task of identifying premises in textbox- and chat-messages as defined above. Results are summarized in Table 2.

The bag-of-word approach performs reasonably well. In combination with the SVM the bag-of-words approach achieves a F1 value of 0.81. Vector representation used from the Google News database, however, performs slightly worse. Across all classifiers the F1-value is between 0.71 and 0.78. Word vectors that are specifically trained on our data (GloVe) perform even worse in predicting premises 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 dataset. This indicates that for small data sets such as ours obtained from an online experiment, the use of vector representations 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.

Vector representations obtained from BERT in combination with a random forest classifier perform almost equally well compared to the bag-of-words features in combination with SVM, reaching a F-value of 0.8. It has to be noted, however, that only the BERT Base model was used and no specific fine-tuning on the dataset at hand was performed. Further performance gains could be expected with fine-tuning or if BERT Large was used.

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). As expected, the length and average number of syllables are very informative with regard to a premise 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 (ug) *will* indicate for most participants that they 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

| | LR | SVM | NB | RF |
|---|---|---|---|---|
| **Bag-of-words** | | | | |
| Accuracy | 0.82 | 0.83 | 0.67 | 0.8 |
| Precision | 0.75 | 0.75 | 0.72 | 0.69 |
| Recall | 0.84 | 0.87 | 0.33 | 0.91 |
| F1 | 0.79 | 0.81 | 0.45 | 0.79 |
| **Google Word Vectors** | | | | |
| Accuracy | 0.7 | 0.8 | 0.68 | 0.82 |
| Precision | 0.59 | 0.71 | 0.59 | 0.77 |
| Recall | 0.9 | 0.84 | 0.74 | 0.8 |
| F1 | 0.71 | 0.77 | 0.66 | 0.78 |
| **GloVe** | | | | |
| Accuracy | 0.73 | 0.77 | 0.67 | 0.8 |
| Precision | 0.66 | 0.68 | 0.58 | 0.78 |
| Recall | 0.71 | 0.81 | 0.74 | 0.71 |
| F1 | 0.69 | 0.74 | 0.65 | 0.75 |
| **BERT (base)** | | | | |
| Accuracy | 0.77 | 0.78 | 0.81 | 0.84 |
| Precision | 0.72 | 0.75 | 0.74 | 0.81 |
| Recall | 0.73 | 0.71 | 0.8 | 0.79 |
| F1 | 0.73 | 0.73 | 0.77 | 0.8 |

Table 2: Prediction results by feature-set and classifier. Classifiers used: Logistic Regression (LR), Support-Vector Machine with linear Kernel (SVM), Naïve Bayes (NB) and Random Forest (RF).

sentences.

Finally, unigrams (ug) 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 positions on rent control or take into account the values behind arguments.

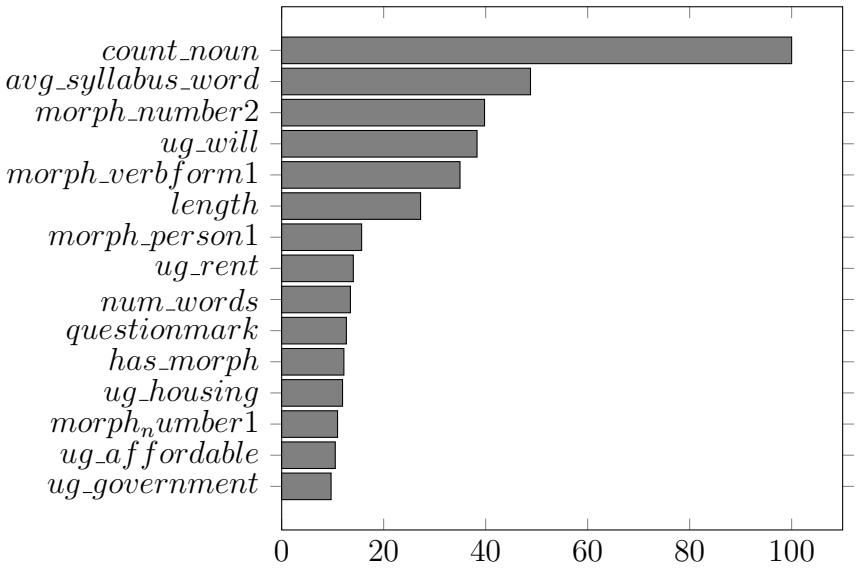


Figure 1: Feature Importance in RF with BOW-features.

# 6 Discussion

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 also investigates the usefulness of machine learning techniques for 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.

It is important to note that the methods we use for classification are very generic in nature and can be easily applied to other concepts that are useful to study in economic experiments such as policy positions or expressed

social preferences. In each case, one only needs a sufficiently large manually coded subset of the textual data to train a classification algorithm a given concept. The size of the training set strongly depends on the sophistication of the concept being analyzed but should generally be at least a few hundred examples per class, e.g. arguments versus non-arguments.

In order to automatically detect policy positions in natural language text, unsupervised scaling techniques such as Wordscores (Benoit and Laver 2003) or Wordfish (Slapin and Proksch 2008) have been developed. Despite their striking success in applications to party manifestos, it is more difficult to apply these methods to rather short text, i.e. the contribution of one individual to a chat discussion. Since these unsupervised methods rely on distributional knowledge based on word frequencies, sufficiently large amount of text is needed to efficiently estimate differences (in policy positions) across individuals. The supervised machine learning approach proposed in this paper has the advantage that an efficient estimate can be obtained for rather short pieces of text making it very much suitable for (experimental) chat data.

# 7 Conclusion

In this paper, we studied the performance of state-of-the-art language models on the task of premise detection in textbox- and chat-messages collected through an online survey experiment. Despite the challenge of having relatively short and formally unstructured messages, we can detect premises 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 vector representations obtained from the contextual language model BERT.

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

# References


Abbott, Rob, Marylin Walker, Pranav Anand, Jean. E. Fox Tree, Robeson Bowmani, and Joseph Kind. Recognizing disagreement in informal political argument. In *Proceedings of the Workshop on Language in Social Media (LSM 2011)*, pages 2–11, Portland, Oregon, 2011.

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

Aker, Ahmet, Alfred Sliwa, Yuan Ma, Ruishen Lui, Niravkumar Borad, Seyedeh Ziyaei, and Mina Ghobadi. What works and what does not: Classifier and feature analysis for argument mining. In *Proceedings of the 4th Workshop on Argument Mining, Copenhagen*, pages 91–96, Copenhagen, Denmark, 2017.

Benoit, Kenneth and Michael Laver. Estimating irish party policy positions using computer wordscoring: The 2002 election - a research note. *Irish Political Studies*, 18:97–107, 2003.

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

Cabrio, Elena and Serena Villata. Natural language arguments: A combined approach. In *ECAI*, 2012.

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

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *Computer Science*, abs/1810.04805, 2018. URL `http://arxiv.org/abs/1810.04805`.

Ghosh, Debanjan, Smaranda Muresan, Nina Wacholder, Mark Aakhus, and Matthew Mitsui. Analyzing argumentative discourse units in online interactions. In *Proceedings of the 1st Workshop on Argumentation Mining*, pages 39–48, Baltimore, Maryland, 2014.

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

Indurkhya, Nitin and Fred J. Damerau. *Handbook of Natural Language Processing*, volume 2. Chapman and Hall/CRC, Boca Raton, Florida, 2010.

Lawrence, John and Chris Reed. Combining argument mining techniques. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 127–136, Denver, Colorado, 2015.

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

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

Liu, Haijing, Yang Gao, Pin Lv, Mengxue Li, Shiqiang Geng, Minglan Li, and Hao Wang. Using argument-based features to predict an analyse review helpfulness. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1358–1363, Copenhagen, Denmark, 2017.

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

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *Computer Science*, abs/1301.3781, 2013. URL `https://arxiv.org/abs/1301.3781`.

Mochales, Raquel and Marie-Francine Moens. Argumentation mining. *Artificial Intelligence and Law*, 19(1):1–22, 2011.

Oraby, Shereen, Lena Reed, Ryan Compton, Ellen Riloff, Marilyn Walker, and Steve Whittaker. And that's a fact: Distinguising factual and emotional argumentation in online dialogue. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 116–126, Denver, Colorado, 2015.

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

Penczynski, Stefan. Using machine learning for communication classifcation. *Experimental Economics*, 22:1002–1029, 2019.

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

Rinott, Ruty, Lena Dankin, Carlos Alzate Perez, Mitesh M. Khapra, Ehud Aharoni, and Noam Slonim. 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*, pages 440–450, Lisbon, Portugal, 2015. Association for Computational Linguistics.

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

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

Slapin, Jonathan B. and Sven-Oliver Proksch. A scaling model for estimating time-series party positions from texts. *American Journal of Political Science*, 52:705–722, 2008.

Smith, Laura M., Linhong Zhu, Kristina Lerman, and Zornitsa Kozareva. The role of social media in the discussion of controversial topics. In *2013 International Conference on Social Computing*, pages 236–243, Alexandria, Virginia, 2013.

Swanson, Reid, Brian Ecker, and Marilyn Walker. Argument mining: Extracting arguments from online dialogue. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 217–226, Prague, Czech Republic, 2015.

Toulmin, Stephen E. *The Use of Argument*. Cambridge University Press, 1958.

Walton, Douglas. Argumentation theory: A very short introduction. In Simari, Guillermo and Iyad Rahwan, editors, *Argumentation in Artificial Intelligence*, pages 1–22. Springer, Boston, Massachusetts, 2009.

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

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

Yin, Jie, Nalin Narang, Paul Thomas, and Cecile Paris. 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*, pages 61–69, Jeju, Korea, 2012.