Journal of Economics and Behavioral Studies (ISSN: 2220-6140)
Vol. 13, No. 1, pp. 32-56, February 2021

How to Analyze Communication Data from Laboratory Experiments Without Being a Machine Learning Specialist

Benjamin Wegener
Ostwestfalen-Lippe University of Applied Sciences and Arts, Campusallee 12, Lemgo, Germany
benjamin.wegener@protonmail.com

Abstract: Recently, the analysis of communication has gained attention in experimental research. One important question is whether certain types of communication affect decisions differently than others. In this regard, Houser & Xiao (2011) present an approach for the classification of natural language messages. The primary limitation of their approach is its limited applicability to large message datasets. Therefore, Penczynski (2019) extends the methodological instruments by applying a machine learning classifier to experimental communication data. This is accompanied by the problem of a dearth of machine learning knowledge among experimenters. Hence, this paper presents an approach that employs a publicly available machine learning text analysis application. This makes it possible to analyze larger datasets based on small training datasets classified beforehand by human evaluators. As a first step, I use primary communication data reported by Charness and Dufwenberg (2006) to generate both training and test datasets. Following this approach, I am able to substantially replicate the original classification results obtained by Charness and Dufwenberg. The second step again involves messages from Charness and Dufwenberg as training data, while I take messages from a related trust game published by Deck et al. (2013) as a test, dataset. Promisingly, I am also able to replicate the classification results obtained by the external evaluators, as reported by Deck et al. The findings suggest that machine learning can be used to analyze large message datasets, both if the artificial intelligence is trained with data from the very same experiment and if it is trained with message data from a comparable experiment.

Keywords: *Laboratory Experiments; Communication; Classification of communication; Machine learning.*

1. Introduction

Experimental literature from economics and the social sciences in general provides rich data on the importance of natural language communication for decision-making in economic environments (see e.g. Isaac and Walker (1988)). In addition to demonstrating the importance of communication per se, the literature provides evidence that, in laboratory experiments, free-form communication results in better outcomes for the players as compared to restricted or pre-specified communication (for comparisons of free-form communication to pre-specified communication, see e.g. Lundquist et al. (2009); for an overview of related literature, see e.g. Agranov and Yariv (2018)). A major challenge associated with the use of free-form communication is that not only do the effects of the very existence of communication need to be analyzed, but also the content and intent of communication itself.[1] However, until now, very few methods have been available to gain deeper insight into free-form communication. Common approaches are, among others, the extraction of relevant keywords and the number of messages sent (see e.g. Huerta (2008); Moellers et al., (2017)). Besides analyzing keywords and message counts, economists have recently started to investigate whether certain types of communication (e.g. promises) might affect decisions differently than other types of communication (e.g. empty talk). Such investigations require the classification of natural language communication.

I am grateful for the helpful comments provided by participants of the 2018 annual meeting of the German Association for Experimental Economic Research (GfeW e.V.) and two anonymous referees. I would also like to thank Eva Tebbe, Korbinian von Blanckenburg, Christian Faupel, Philipp Russinger and Christoph Richert for their helpful comments.

[1] The intent of communication refers to a category in which the communication can be classified, e.g. ‘promise’ or ‘empty talk’.

In their well-cited paper, Charness and Dufwenberg (2006) present one of the first approaches of this kind. They classify messages from their experiment[2] into the two categories ‘empty talk’ and ‘promise’. In response, Houser and Xiao (2011) criticize this approach due to its subjective nature and describe an objective procedure for the classification of natural language messages. Related to the ESP game[3] (Ahn & Dabbish, 2004), which was used to label images and enhance the accessibility of their contents, H&X reports a coordination game that is used to ‘label’ – that is, to classify – natural language messages. External evaluators read a corpus of messages and decide for each individual message whether it is to be classified as ‘promise’ or as ‘empty talk’.[4] The participants of the game are incentivized financially so that they receive money if their evaluation matches the most common evaluation of the other participants.[5] Because the classification game of H&X is objective and easy to replicate, their game has been applied in many experiments (see e.g. Fischer and Normann (2019) and Huang and Xiao (2018)). Furthermore, H&X claims that its approach can clarify and extend the nature of conclusions reached, at a very small cost.

I argue that this is correct for a small dataset of messages, as one might see, for example, in the case of one-shot games with a small subject pool. On the other hand, their approach is no longer easy to replicate nor efficient for larger datasets. This might be why many experimenters analyze their data with fewer evaluators than H&X (see e.g. Ismayilov and Potters (2016) and Moellers et al., (2017)).[6] Even though this may save resources, the results potentially lack validity and robustness. I argue that human classification applied to a share of the messages from a laboratory experiment can be used as a training set for a machine learning approach. This is in line with the original intention of the ESP-Game – building a training dataset to label unlabeled pictures using a machine learning approach. In this way, larger datasets can be analyzed with only slightly more effort than small datasets. The most comparable publication to this study is Penczynski (2019). He presents a supervised machine learning approach to classifying messages according to their level in the level-k model of strategic reasoning. Research assistants classify messages from different laboratory experiments independently. Afterward, the classifications are reconciled and the research assistants have to agree upon one consistent classification.

[2] C&D implement a hidden-action trust game. Two participants are paired, one of whom is player A and the other player B. A has to choose ‘in’ or ‘out’. Without knowing A’s decision, B has to choose either ‘roll (a die)’ or ‘don’t roll (a die) ’. If A has chosen ‘out’, A and B receive $5 each. If A has chosen ‘in’ and B has chosen ‘don’t roll’, A receives $10 and B receives $14. If A has chosen ‘in’ and B has chosen ‘roll’, B receives $10 and rolls a six-sided die in order to determine the payoff of A. If the die yields a 1, A receives $0. If the die yields a number in the range from 2 through 6, A receives $12. The trust game played by C&D involves several treatments. Besides the explained approach, the authors vary in terms of whether preplay communication is allowed or not. If it is allowed, B can send a message to A or A can send a message to B. Additionally, the authors also change the payoff vector if A has chosen ‘out’ from (5, 5) to (7, 7).

[3] In the ESP game, two randomly paired participants label images without any pre-specified labels. Both are shown the same image. The participants do not know each other and are not allowed to communicate. The goal of the ESP game is to guess which image label the other participant has chosen. If the participants type in the same string while the image is on the screen, they move on to the next image. For every agreement, the participants get a certain number of points (Ahn & Dabbish, 2004).

[4] We do not discuss certain possible disadvantages of this approach. H&X point out that there are several aspects suitable for ongoing research, such as ‘lay’ evaluators. Furthermore, there is a discourse regarding the use of experts rather than non-experts, especially for deeper natural language tasks. Snow et al. (2008) analyze the use of non-experts recruited via Amazon Mechanical Turk as compared to experts. They find evidence that an average of four non-expert evaluators are required to emulate expert-level label quality.

[5] This is in contrast to the ESP game, which relies on the effect of entertainment and gamification as its incentives.

[6] H&X do not suggest a specific number of evaluators, but they argue that ‘The average opinion of a large number of evaluators, (…), is a reasonable way to infer (…) the way the message was likely interpreted.’

The classified datasets are used to employ a Random Forest Classifier while common steps of natural language processing are utilized (all steps are implemented in R). Although Penczynski (2019) obtains promising results and argues that the implementation of a Random Forest Classifier can be accomplished without much effort, the task of classifying natural language messages is anything but trivial. Besides determining a proper classification algorithm,[7] the preprocessing steps applied to the communication data require knowledge of stemming, lemmatization, part-of-speech tagging, feature engineering, and a multitude of other possible tasks. Therefore, I argue that an 'out-of-the-box approach can better fulfill the requirements of experimenters while reducing the effort and presupposed domain knowledge necessary to classify communication data. This study suggests a procedure utilizing such an 'out-of-the-box approach. This procedure involves the construction of a machine learning model based on IBM Watson Conversation[8]. The application is trained with a training dataset that encompasses human codings.

Based on these training messages, the machine learning classification algorithm learns the logical associations necessary to classify further messages. Consequently, it is able to classify messages in a test dataset. This study uses messages from C&D as well as Deck et al. (2013) (henceforth Deck et al.,) to test this approach. Using existing data allows me to compare the results generated by the machine learning classification algorithm to the original human coding results reported by C&D as well as Deck et al. To ensure the robustness of the approach, an estimation of the relative number of training set messages is needed. Therefore, software-generated results are cross-validated for various training set sizes with the original human classifications. In general, the results show that the approach is effective, both when training messages are taken from the very same experimental dataset as test messages, and when training messages are taken from an experimental dataset separate from that of the text messages. As a recommendation for use, this study provides rules of thumb regarding both the number of training messages necessary to generate good results and the distribution of messages per category.

2. Background

The approach presented in this paper focuses on adapting machine learning technology for unstructured data, i.e. text documents such as, in this case, chat messages. Although there will be no detailed discussion of the exhaustive literature on this topic, the essential approaches will be explained. Machine learning is the science of programming computers so that they can learn from data. Depending on whether the learning requires any type of supervision, machine learning systems can be classified as supervised or unsupervised.[9] In keeping with machine learning terminology, the classification of messages is to be assigned to the category of supervised machine learning, because the classification is supervised by the knowledge and intuition of human evaluators (Sebastiani (2002). The machine learning algorithm is capable of classifying natural language texts with a classification from a predefined set of classifications. This requires the availability of a corpus of manually pre-classified messages. These classifications must be provided by human evaluators. The pre-classified corpus of messages is the training dataset. The remaining corpus of unclassified messages is the test dataset. Both the training dataset and test dataset are subsets of the initial corpus of messages. There is no general rule on how to split the initial corpus (e.g., Hastie et al., (2009) suggest a split of 50/50).[10] Choosing

[7] Penczynski utilizes a Random Forest Classifier. In the area of classification algorithms, great progress has been made recently. Algorithms like XG Boost and Light Gradient Boosting Machine have outperformed Random Forest Classifier in several tasks (see, e.g., Chen & Guestrin (2016)).

[8] https://www.ibm.com/watson/

[9] Beside these categories there are some others, such as semi-supervised and reinforcement machine learning.

[10] It is obvious that these recommendations are dependent on the absolute number of available messages and that ongoing technical progress in the area of machine learning ensures ever better learning, also on the basis of small datasets.

too many messages for the training dataset could result in over fitting, i.e. the act of overgeneralizing in a training process by providing fewer training data for one set of cases.

The model interpretations become tighter and more specific the more messages are included in the training set so that the model performs poorly on a given random set. To preclude this possibility, the training dataset is split once more, so that there is an additional validation dataset to check for both over-fitting and under-fitting. In my approach, I adapt the machine learning techniques used to identify the intention of a message and to generate an appropriate response. To this end, this study uses the application IBM Watson Conversation.[11] I focus on the identification of the intention of a message. An intention is a purpose or goal that is expressed in natural language. In the following section, I explain the general methodological approach.

3. Methodology

The starting point of my approach is an initial corpus of messages (M) and a predefined set of categories (C):

(1) M $:= \{m_{1,} \ldots, m_n \mid$ all messages from the initial corpus of messages$\}$

(2) C $:= \{c_1, \ldots, c_b \mid$ all predefined categories$\}$

The number of categories, or intents (e.g. 'empty talk' and 'promise'), is only limited by the number of messages from the initial corpus, i.e. an appropriate number of messages is needed for every category c_i ($i \in \{1, \ldots, b\}$) in C. In this sense, the number of categories might be much higher than just two. For ease of presentation, I discuss the methodology as a binary classification task with only two intents, subsequently referred to as 'intent 1' (c_1) and 'intent 2' (c_2), so that $b = 2$ and $i \in \{1; 2\}$.

Based on the messages and categories, the procedure includes seven steps. As a first step, the initial corpus of messages is split into a training dataset (M_{Tr}) and a test dataset (M_{Te}).

The allocation of messages to the training dataset and test dataset is given by:

(3) M_{Tr} $:= \{m_1, \ldots, m_k \mid$ randomized subset of M, with $k < n$ messages$\}$

(4) M_{Te} $:= \{M \setminus M_{Tr} \mid$ remaining $n - k$ messages from M$\}$

As a second step, to follow a supervised machine learning approach, the randomly chosen messages included in the training dataset have to be classified manually and independently by two or more human coders.[12] To obtain unambiguously classified training data, as a third step, the interrater agreement of the evaluators is to be checked using Cohen's Kappa or Fleiss' Kappa[13], depending on the number of evaluators (Landis and Koch (1977)). Landis and Koch differentiate between 'moderate agreement' ($0.41 \leq \kappa \leq 0.60$), 'substantial agreement' ($0.60 < \kappa \leq 0.80$), and 'almost perfect agreement' ($0.80 < \kappa \leq 1.00$). The interrater agreement should at least be 'substantial'. Only messages which are classified unanimously (in the case of two evaluators) or at least by the majority of evaluators (in the case of more than two evaluators) should be used for the training dataset.[14] Ambiguously categorized messages, i.e. messages with the same wording.

[11] There are several other applications that could have been used, as will be discussed in the conclusion and outlook section. I have chosen IBM Watson Conversation (since renamed IBM Watson Assistant) because it is one of the available machine learning software applications with the highest degree of maturity in text analysis.

[12] According to the literature, there are various ways to do so. As already stated, H&X argue that a classification by the authors themselves is subjective. Instead, they recommend involving external evaluators.

[13] Cohen's Kappa and Fleiss' Kappa are statistical measures for assessing the reliability of observer agreement for nominal scales between two (Cohen's Kappa) or more than two observers (Fleiss' Kappa) (see Cohen (1960); Landis and Koch (1977)). 'It is directly interpretable as the proportion of joint judgments in which there is agreement, after chance agreement is excluded' (Cohen (1960), p. 46).

[14] The original ESP game used a 'good label threshold', i.e. before a label was attached to an image, it must have been agreed upon by at least a specified number of evaluator pairs. The specified number is the threshold, which can be lenient or strict.

One of which is categorized with 'intent 1' and the other with 'intent 2', should be excluded.[15] Accordingly, the classified corpus of messages, i.e. the training dataset, has to be revised as follows:

(5) $M_{R\ Tr}$:= {M_{Tr}} \ {all inconclusively classified messages}

After classification and revision for all j ∈ {1, …, k} and for all i ∈ {1, 2} it is known whether $m_j \in C_1$ or $m_j \in C_2$ holds so that C_i:= {$m_{i1,}$ …, m_{ix} | all messages classified with the intent c_i}. The revised and classified training dataset is given by:

(6) $M_{R\ C\ Tr}$:= {m_1, …, m_h | classified massages from M_{Tr}; h ≤ k}

As a fourth step, the revised and classified training dataset ($M_{R\ C\ Tr}$) is split into two more subsets, the original training dataset ($M_{O\ Tr}$) and the validation dataset (M_{Va}). The different datasets can be summarized as follows:

(7) $M_{O\ Tr}$:= {m_1, …, m_g | randomized subset of $M_{R\ C\ Tr}$, with g < h messages}

(8) M_{Va} := {{$M_{R\ C\ Tr}$} \ {$M_{O\ Tr}$}| remaining h-g messages from $M_{R\ C\ Tr}$}

In order to specify the allocation of the messages (according to their categorization as a member of C_1 or C_2) in the original training dataset, the set can be defined by the cardinality of the set itself and its subsets C_1 and C_2 as $M_{O\ Tr} = M_{|O\ Tr|\ |C1|\ |C2|}$.[16]

The distribution of 'intent 1' and 'intent 2' in the revised and classified training dataset should be split proportionally between the original training dataset and the validation dataset. The validation dataset is needed to test the effectiveness of the original training dataset. As a fifth step, the original training dataset is uploaded on the IBM Watson Conversation workspace to train the machine learning model. After the machine learning model has been trained, the validation dataset is uploaded. The software then classifies validation messages as 'intent 1', 'intent 2', or the generic classification 'irrelevant'.[17] All messages that could not be assigned to one of the 'intent' classifications are classified as 'irrelevant.[18] In the sixth step, the whole revised and classified training dataset is checked for interrater agreement once more. The rater results are the classified messages yielded by human evaluators on the one hand and the classified messages of IBM Watson Conversation on the other. In addition to the interrater agreement for the whole revised and classified training dataset, the interrater agreement results for its subsets, the original training dataset, and the validation dataset also need to be analyzed.[19] As was the case in the third step, the target range for an interrater agreement should at least be 'substantial'. If the interrater agreement is outside the target range, over fitting or under fitting of the model may be the reason.

[15] E.g. a message like 'okay' could be labeled as a 'promise' or 'empty talk', depending on which kind of question is answered.

[16] In the previous step three, all messages with the same wording were excluded. Therefore, the cardinality of the sets is equal to the number of elements in the sets.

[17] The machine learning algorithms and the initial training data on which the artificial intelligence of IBM Watson Conversation is built are unknown. IBM has not published any information about the technologies and datasets used (Braun et al. (2017)). However, it seems likely that IBM validates the 'optimal' classification algorithm within the IBM Watson Conversation application, as IBM offers a stand-alone solution for the estimation of the best-performing classification algorithm with IBM Watson Auto AI.

[18] In the next section I will discuss further conclusions and applications of messages classified as 'irrelevant'.

[19] Depending on the employed machine learning algorithm(s), it is possible that the machine learning model is capable of classifying 100% of the messages correctly (as was the case for IBM Watson Conversation). In this case, the interrater agreement is an indicator for error detection in the context of training a model. For robustness checks, I implement a Multi-layer Perceptron (MLP) Classifier and a Random Forest (RF) Classifier in Python and test the same datasets on these classifiers. As a result, I observe 100% interrater agreement on the training data for the RF Classifier and 99.16% interrater agreement for the MLP Classifier on the training data (i.e. the MLP Classifier is capable of detecting 97.8% of all training datasets with 100% interrater agreement). The results can be found in Appendix B. The Python code will be provided upon request.

In such a case, the size of the original training dataset should be varied (once with more, once with fewer messages taken from the validation dataset M_{Va}). Depending on whether more or fewer messages result in a better interrater agreement, the final number of messages should be determined (as well as the proportional division between the particular classifications). Machine learning text classification models rely heavily on features such as words, numbers, and punctuation marks (among others). If there is not a sufficient number of features to separate the categories, a machine learning model will probably not work, although humans are still capable of classifying messages in such situations. Once an original training dataset that yields good results has been defined, the seventh step entails uploading the test dataset to classify all messages.

4. Results

In order to assess the approach, I apply it to available and classified message data from simple one-shot trust games. The starting point is the corpus of messages and classifications reported in the supplemental material of C&D.[20] All blank messages are excluded. Therefore, the initial corpus of messages consists of 81 messages:[21]

(9) $M_{CD} = \{m_1, \ldots, m_{81} \mid \text{all messages reported by C\&D, except blank messages}\}$

C&D defines three categories. Deleting all blank messages, the category 'no message' is excluded from the analysis.[22] I employ the remaining two categories, 'promise' and 'empty talk':

(10) $C_{CD} = \{c_1, c_2 \mid \text{with } c_1 = \text{'promise' and } c_2 = \text{'empty talk'}\}$

As the messages reported by C&D are already coded and checked for interrater agreement beforehand, for the data used in this study, the training dataset is similar to the revised training dataset, as well as the revised and classified training dataset, i.e. $M_{Te} = M_{R\ Tr} = M_{R\ C\ Tr}$. However, as this study is exploratory in nature, before defining the revised and classified training dataset, I need to include one further step, which is not part of the procedure described above. In this step, I estimate the relative share of the entire message corpus needed for the original training dataset, as well as the relative share of messages classified as 'promise' and 'empty talk'. To do so, I use an iterative heuristic. That is, I vary the overall size of the training set.

As well as the share of messages that are associated with either c_1 or c_2, to assess how many messages are needed to yield good results while minimizing the risk of overfitting. As a minimum, k is set to $k=|M_{O\ Tr}|=8$, which is about 10% of the size of the initial corpus or messages. In terms of the minimum number of messages per category, all of the tested training sets include at least one message classified as 'promise' and 'empty talk' respectively. Specifically, this implies that there are seven combinations of 'promise' and 'empty talk' messages when the training set includes eight messages in total. For the first case (a total of eight messages from C&D with one randomly chosen message coded as 'empty talk' and seven randomly chosen messages coded as 'promise'), the original training dataset is:

(11) $M_{1\_|O\ Tr|\ |Ce|\ |Cp|} = M_{8\ 1\ 7} = \{m_{ijep} \mid 1 \le j \le 8 \wedge e = 1 \wedge 1 \le p \le 7\}$,
with $1 \le i_1 < i_2 < \ldots < i_8 \le 81$ and $i_j \in \mathbb{N}$, $j=1; 2; \ldots; 8$.

For the second case, the original training dataset can be defined as:

(12) $M_{2\_|O\ Tr|\ |Ce|\ |Cp|} = M_{8\ 2\ 6} = \{m_{ijep} \mid 1 \le j \le 8 \wedge 1 \le e \le 2 \wedge 1 \le p \le 6\}$,

[20] Because of its similarity, I used all messages sent from B in the payoff vectors (5,5) and (7,7). Overall, C&D reported 91 messages in these payoff vectors, of which 10 were blank and without any text.

[21] As stated in the previous section, messages with the same wording, which were categorized with 'intent 1' and 'intent 2', should be excluded. I did not identify any such messages. There are two messages which could have fallen into this category: 'I'll choose to roll' and 'I will choose to roll'. Because they do not feature exactly the same spelling and were both assigned the intent 'promise', I decided to include both messages in the initial corpus of messages.

[22] This is due to the fact that this would distort the results, as a training of just one blank message with the intent 'no message' would always result in the 100% agreement of all blank messages with this intent.

with $1 \leq i_1 < i_2 < \ldots < i_8 \leq 81$ and $i_j \in \mathbb{N}$, j=1; 2; …; 8. For the upper limit in terms of training set size, there is no standard recommendation in the literature.

In order to limit the effort associated with manual evaluation, the upper limit is set to 2/3 of the total number of messages (which is 54). Thus, all possible solutions in terms of the allocation to 'empty talk' and 'promise' are varied within the range of 8 to 54 messages, i.e. |O Tr|=8; 9;…;54. In this way, I test 683 different combinations of |O Tr|, |Ce|, and |Cp|.[23] To achieve robust results, I test five randomly assigned original training datasets within the above-described limits for every possible variation of |O Tr|, |Ce|, and |Cp|, which amounts to a total of 3,415 datasets.[24] After the training of the model on IBM Watson Conversation has been finished for a certain original training dataset, the initial corpus of messages (which, in the case of this study, is composed of the original training dataset and the validation dataset) is uploaded and classified. With regard to the category 'empty talk', as in all other experiments concerning 'empty talk' intent.

It is very broad and can vary in terms of subject matter, dealing with topics such as the weather, politics, sports, and many more. Therefore, IBM Watson Conversation uses the category 'irrelevant', which encompasses all messages that could not be assigned to one of the pre-specified categories. Still, if the other categories are unambiguously classified, I argue that it is legitimate to assume that all messages that have been classified by IBM Watson Conversation as 'irrelevant' can be reclassified as 'empty talk'.[25] Applying this principle, I get the following interrater agreement results for the initial corpus of messages.[26] Of the 3,415 original training datasets, 2,476 yields a Kappa score on the initial corpus of messages that is at least substantial ($\kappa > 0.60$). Correspondingly, for the remaining 939 datasets, the Kappa score is equal to or smaller than the threshold level of 0.6.

Result 1: For more than 70% of training datasets, the machine learning-generated classifications on the initial corpus of messages are substantially similar to the classifications of C&D. Among the 939 poor-performing datasets, 925 have either eight or fewer 'empty talk' messages or eight or fewer 'promise' messages. Accordingly, there are a minimum number of messages per category that is necessary to train a valid model.

Result 2: The minimum number of messages per category necessary to train a valid machine learning model is roughly 10% of the initial corpus of messages. Checking for interrater agreement only on the messages of the validation dataset, 1,259 datasets have a Kappa score larger than 0.6 and 2,156 datasets a Kappa score equal to or smaller than the threshold of 0.6. I also checked the Kappa score on all original training datasets. However, these results are just for purposes of error detection, as IBM Watson Conversation was capable of classifying 100% of the messages of the original training dataset correctly. Aside from the agreement results, I am also interested in whether certain messages are regularly coded differently by the machine learning application in comparison to human coding. Therefore, I code the classifications of IBM Watson Conversation,

[23] In the reported and classified data of C&D, there are no more than 33 messages evaluated as 'empty talk'. To ensure that at least a (randomly selected) third of all 'empty talk' messages were part of the training dataset, the upper limit for 'empty talk' messages is set at a total of 22. Accordingly, the maximum number of messages classified as 'promise' in the original training dataset is 32.

[24] I used Stata to create the files.

[25] This is in line with the results. Among all 3,415 datasets, IBM Watson Conversation classified 17,114 messages under the category 'irrelevant'. 16,148 of them (i.e. 94.4%) were assigned by C&D to the category 'empty talk', while 966 messages (i.e. 5.6%) were classified as 'promise'. In the course of implementing both the MLP Classifier and the Random Forest Classifier in Python for robustness checks, I adopted a similar approach. Those messages with an estimated probability of 0.5 for both categories were assigned to category 1, i.e. 'empty talk'.

[26] The results were calculated with Stata. The Stata code, as well as the full classification of messages, can be supplied upon request.

as well as those of C&D, to ‘1’ for ‘empty talk’ and ‘2’ for ‘promise’. Based on these values, I determine the mean assessment overall training datasets and compare this mean value to the assessment of C&D. Overall, there are three individual messages where the mean deviates by more than 0.5 from the human coding.[27] For most of the messages, the deviation was less than 0.15.

Result 3: Over the full range of training datasets, the average assessment of the machine learning approach per message corresponds to the assessment of C&D for 78 out of 81 messages. Up to this point, the descriptive results have been discussed. An unambiguous training of this kind of category would be close to impossible, not to mention cost-prohibitive. For a deeper understanding, the models reported in Table 1a and Table 1b evaluate the impact of the total number of messages in the original training dataset (#Messages_J), the number of empty talk messages (#EmptyTalk_I), the number of promise messages (#Promise_K), and interaction terms of these three variables on Kappa.

Table 1a: Tobit Regression Results, All Messages Included

	(a)	(b_e)	(b_p)	(c_e)	(c_p)
#Messages_ J	0.0107*** (0.0002)	0.0113*** (0.0003)	0.0094*** (0.0004)	0.0042*** (0.0006)	0.0166*** (0.0007)
#EmptyTalk_I	--	-0.0019*** (0.0005)	--	-0.0180*** (0.0012)	--
#Promise_K	--	--	0.0019*** (0.0005)	--	0.0128*** (0.0009)
#Messages_ J X #EmptyTalk_I	--	--	--	0.0006*** (0.0000)	--
#Messages_ J X #Promise_K	--	--	--	--	-0.0004*** (0.0000)
Constant	0.3590*** (0.0073)	0.3648*** (0.0074)	0.3648*** (0.0074)	0.5450*** (0.0144)	0.1960*** (0.0141)
Obs.	3415	3415	3415	3415	3415
LR chi^2	1585.28^{a}	1600.89^{a}	1600.89^{a}	1803.07^{a}	1790.85^{a}

Standard errors are reported in parentheses. * p<0.1, **p<0.05, ***p<0.01, ^{a} p < 0.001.

Table 2b: Tobit Regression Results, All Messages Included

	(d_e)	(d_p)	(e_e)	(e_p)
#Messages_ J	-0.0026** (0.0011)	0.0193*** (0.0012)	0.0154*** (0.0012)	0.0100*** (0.0019)
#EmptyTalk_I	-0.0614*** (0.0029)	--	0.0329*** (0.0024)	--
#Promise_K	--	0.0463*** (0.0027)	--	-0.0151*** (0.0024)
#Messages_ J X #EmptyTalk_I	0.0023*** (0.0001)	--	-0.0006*** (0.0001)	--
#Messages_ J X #Promise_K	--	-0.0016*** (0.0001)	--	0.0001** (0.0001)
Constant	0.7393*** (0.0243)	0.0633** (0.0250)	0.0734* (0.0428)	0.6140*** (0.0650)
Obs.	1930	1930	1485	1485

[27] These messages are those from B to A in the (5, 5) treatment with the ID 10 in Session 1, with the ID 14 in Session 1, and in the (7, 7) treatment with the ID 6 in Session 1.

LR chi²	729.28[a]	605.33[a]	1192.30[a]	1105.51[a]

Standard errors are reported in parentheses. * p<0.1, **p<0.05, ***p<0.01, [a] p < 0.001.

Kappa as dependent variable (i.e. Kappa score overall messages, while irrelevant messages are coded as empty talk). Model (a) supports the finding that there is no substantial evidence of over-fitting effects, as there is a positive effect of the total number of messages constituting the original training dataset. Model (b) specifies the influence of the number of messages per category. The higher the number of promise messages included in a certain training dataset, the higher the Kappa score. By contrast, a higher number of empty talk messages go along with lower Kappa scores. This seems to confirm that the distinction between the two categories 'empty talk' and 'promise' relies more on an unambiguous training of the category 'promise'. Considering the interaction of the number of all messages and of messages per category in the training data as seen in the model (c), when the size of the training dataset increases, the positive effect of including more promise messages decreases. Again, the number of empty talk messages induces a contrary effect.

When the size of the training set increases, the negative effect of including more empty talk messages is smaller. Based on these findings, I split the total number of messages in the original training dataset into two sub-groups: (d) a group of sets with a small number of messages, so that J ≤ 30, and (e) a group of sets with a large number of messages, so that J > 30. Given a small training dataset size, the results discussed above are still valid. However, given a large training dataset size, contrasting results can be observed. In this case, a higher number of 'promise' messages go along with a reduction in Kappa, while an increasing number of 'empty talk' messages yield higher Kappa scores. This result might display over-fitting effects of including too many 'promise' messages. Whereas with small training set sizes, including more 'promise' messages seems to be vital to yield high Kappa scores, when the number of messages included in the training set is high enough, a higher share of 'empty talk' messages is to be preferred. Thus, a higher share of 'promise' messages could lead to an overfitting effect as the marginal utility of an additional 'promise' diminishes or becomes negative. In contrast, including more 'empty talk' will not lead to over-fitting effects because of its broad range of contents. In order to be able to get a recommendation for choosing a training set size, I analyze the distribution of Kappa, depending on the training set, size in more detail.

Figure 1: True and Predicted Distribution of Kappa (Basis for Prediction: Model (c))

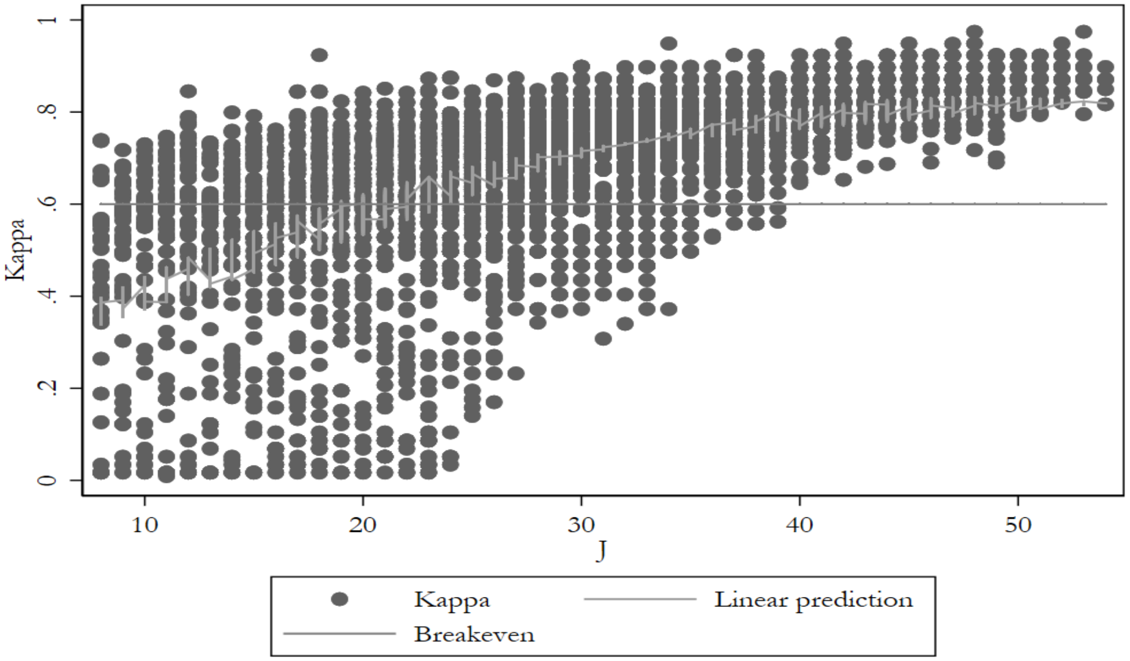

Figure 1 shows the observed Kappa scores for each training set size, as well as the Kappa scores predicted by the model (c). The predicted breakeven point, i.e. the point at which all predicted Kappa scores are higher than 0.6, is J = 23. The true breakeven point is at J = 40. The figure also suggests that there are three groups of training set sizes: (1) those that might result in very low Kappa scores, (2) those that allow for high Kappa scores regularly, but that also involve a certain risk of bad coding, (3) and those that generate Kappa scores that are consistently high. In other words, there seem to be three groups of training set sizes that differ in terms of the variance of Kappa. To determine whether this visual impression is verified analytically, I conduct a cluster analysis with three clusters, defined by the distinctive range of training set sizes. Table 2 presents the results. As expected from the analysis of Figure 1, each of the three clusters encompasses a range of J that represents one of the three groups discussed above. While there are Kappa scores > 0.9 in all three clusters, the minimum Kappa score per cluster is substantially higher in the third cluster as compared to the first and the second. In the same vein, and as expected from the analysis of Figure 1, the standard deviation of Kappa is smaller in the former cluster as compared to the latter two. I conclude that with training set sizes from the third cluster, the probability of accurate evaluations is very high.

With training set sizes from the second cluster, the mean standard deviation of Kappa is 1/3 smaller than with training sets from the first cluster. Consequently, I consider training sets from cluster 1 to be undesirable. I recommend using a training set size that is not smaller than a quarter of the whole set of messages, while training sets that include 1/2 of the total number of messages seem to be sufficiently informative to generate high Kappa scores, even if the number of messages per category cannot be influenced.

Result 4: Training sets that include 50% of the initial corpus of messages yield high Kappa scores even if the distribution of messages over categories is unknown. Yet it is also possible to yield high Kappa scores with far less than 50% of the messages included in the initial corpus of messages. As already stated in the sixth step of the methodology, the size of the original training dataset, as well as the proportion of the messages per category in the training dataset, should be varied (once with more, once with fewer messages taken from the validation dataset M_{Va}). Depending on whether more or fewer messages result in a better interrater agreement, the final number of messages should be determined (as well as the proportion of messages drawn from the various classification categories).

Suggestion: If the distribution of messages over categories is unknown and the experimenter is unable to generate human codings for 50% of the initial corpus of messages, I recommend varying the number of messages in the training set so that interrater agreement is maximized. However, for small message datasets, I advise against using training set sizes smaller than 25% of the initial corpus of messages.

Table 3: Descriptive Statistics for Three Clusters of Training Sets, With Distinctive Ranges of Size_Training_Set

	# Training sets	Mean	SD	Min	Max
Cluster 1 – by J					
Kappa_All	1270	0.54	0.21	0.01	0.92
Number_ET	1270	8.74	5.42	1	22
Number_P	1270	8.83	5.42	1	23
Size_Training_Set	1270	17.57	4.62	8	24
Cluster 2 – by J					
Kappa_All	1380	0.70	0.12	0.14	0.95
Number_ET	1380	11.84	6.19	1	22
Number_P	1380	18.97	6.95	3	32
Size_Training_Set	1380	30.82	3.68	25	37
Cluster 3 – by J					
Kappa_All	765	0.82	0.06	0.56	0.97
Number_ET	765	16.67	4.11	6	22
Number_P	765	26.67	4.11	16	32
Size_Training_Set	765	43.33	4.11	38	54

Indeed, it must be considered that the same training data could have been used to classify far more messages. Depending on the experimental environment and the restricted instructions of the trust game reported by C&D, I argue that more 'promise' messages from more subjects would have been very similar in their structure and utilized phrases.[28] Therefore, I used the above-discussed training data, based on the C&D messages, to classify messages from a comparable experiment reported by Deck et al. In this way, I am also able to demonstrate the scalability of machine learning models for the classification of messages from laboratory experiments. The results are reported in Appendix A, while additional econometric analysis appears in Appendix B. If we now consider the messages reported from C&D along with those reported by Deck et al. as one initial corpus of messages, the results indicate that training sets that include around 1/3 of the total number of messages seem to be sufficiently informative to generate highly substantial classification results. Overall, comparing the results of the validation data derived from C&D messages with validation data from the Deck et al. messages, we see even better results in the case of the Deck et al. messages.

5. Conclusion and Outlook

This paper reports to the best of my knowledge, the first results of an 'out-of-the-box machine learning classification approach utilized to classify messages from a laboratory experiment. Using the knowledge and intuition of human evaluators who classify a corpus of messages, I am able to replicate the classification results presented by C&D and Deck et al. with far less than 50% of the messages from the initial corpus of messages used as training data. With training sets that encompass 50% of the initial corpus of messages, the machine learning algorithm robustly yields high agreement with the human coders, irrespective of the distribution of messages over categories. These results are especially interesting for datasets encompassing far more messages than reported by C&D. Using this approach, large message datasets can be analyzed with the same effort as small message datasets. Furthermore, the study provides evidence of the scalability of the approach. Using training data from one message corpus (reported by C&D) to classify another message corpus (reported by Deck et al.,), I achieve robust and highly substantial results. These findings rely heavily on the similarity of both studies and the instructions of their evaluators.

However, these findings are nevertheless interesting for experimenters who conduct replication studies or split their experimental sessions into several time slots. In the first case, already-labeled data from the original study can be used to train the machine learning model. In the latter case, the communication data from the first-run sessions can already be used to train the machine learning models. Thereby, the process of running experimental sessions and analyzing communication data can be parallelized and the time required to finish a publication can be shortened. Besides the potential to more efficiently analyze large datasets, another advantage of this approach is that the evaluation is history-independent. This means that there is no human bias in the process of evaluating a vast corpus of messages. Although many experimenters consider this in their instructions, it is unlikely that human evaluators are able to act consistently over time.[29] The machine learning algorithm instead evaluates all messages independently and with the same reliability for all messages, as the same algorithm is used for all messages.[30] In their experiment, Nielsen et al. (2019) classify mes-

[28] It is logical to assume that this assumption is not valid for the 'empty talk' category.

[29] See, e.g., coding rules of Deck et al.: 'The unit of observation is a single message'; 'Your job is to capture the content of the message (…). Think of yourself as a "coding machine".'

[30] Natural language understanding services, like IBM Watson Conversation, improve over time. The reproducibility of the results is therefore only conditionally possible. The data was collected in the period from October 2 to October 16' 2018. In this time, I worked on the IBM Watson Conversation version 2018-09-20 (the service has since been renamed IBM Watson Assistant). Therefore, the very same approach with the same messages, exactly as described in this paper, might lead to different (presumably better) results.

sages into numerous categories.[31] In doing so, they try to achieve deeper insight into the rich chat content. Their results show relatively low agreement rates for most of the categories.

I acknowledge that my approach works well for a binary classification task with two categories, but problems might arise when trying to apply it to fields with a broader range of categories, especially if the amount of potential training data for each category is restricted. The agreement of the machine learning classifications with the human coding of the communication data, reported by C&D and Deck et al. relies on the separability of the categories 'empty talk' and 'promise'. The separability is inherent in the diversity of these two categories, as they are characterized by different features (i.e. the words of the messages). The less the categories are characterized by different, and therefore unique, features with regard to their category, the more difficult it is for a machine learning model to separate these categories.[32] Therefore, the application of such machine learning models to less separable categories should be handled with care. A further challenge is the analysis of multiparticipant chats (see, e.g., Uthus and Aha (2013)). Tracking the intention in synchronous discussions within a single message corpus is difficult even in the context of human evaluation, let alone machine analysis. While the classification in a one-shot game with preplay communication is based on the perception of a single isolated message, the classification of messages in a multi-participant chat rests on who is talking to whom, which question is being answered, and many other aspects.

In this paper, the approach was tested using an application developed by IBM: IBM Watson Conversation. In ongoing research, I evaluate other applications that might perform as well as this one or even better (see, e.g., Braun et al., (2017)). Besides this, an open question is whether different applications perform better or worse on different experimental tasks. In addition, the expansion of this approach to other natural language understanding functionalities, e.g. the identification of entities and sentiments, ought to be considered. I am also interested in the further possibilities of adaption and scalability, as well as the influence of external evaluators. Therefore, I want to see whether one could use already-trained models to classify the messages of further laboratory experiments, as already exemplified in the results section. In general, a machine learning model is most reliable if the training dataset follows a distribution similar to that of future datasets, which are still to be classified (Mitchell (1997)). Therefore, I will use already-trained models based on the classifications reported by C&D and H&X to classify the messages reported, e.g., by Ismayilov and Potters (2016).[33]

I do not focus on the impact of single messages on the performance of a machine learning model. However, I observe some messages which were, in the mean, evaluated differently by IBM Watson Conversation than by the human evaluators. This might be due to the fact that these messages entailed contradictory content. Messages with several sentences might be inconsistent if the first sentence of a message is classified as 'promise', whereas the second sentence is classified as 'empty talk'. Assigning one category to this message in the machine learning model is logically false. In further research, I will also clarify whether it is possible to identify in advance messages that will lower the performance of the machine learning model and how to handle such multi-class classified messages. This study has presented a novel approach to analyzing experimental com-

[31] Their experiment was a replication of an experiment by C&D, but using teams instead of individuals.

[32] Nielsen (2019), for example, employs the categories 'weak promise' and 'strong promise'.

[33] Ismayilov and Potters (2016) utilized the trust game reported by C&D and employed external evaluators to classify the messages. Because of this, I want to apply the classifications of H&X, as Ismayilov and Potters (2016) used external evaluators as well. However, in contrast to H&X, Ismayilov and Potters (2016) incentivized the evaluators just for the completion of the task. There were neither additional nor variable earnings if the evaluation matched that of any other nor the majority of the other evaluators. Another essential difference is that the messages reported by Ismayilov and Potters (2016) are much longer. This might lead to ambiguous classification results, as one message could consist of sentences that were classified as 'empty talk' in an isolated evaluation as well as sentences classified as 'promise' in an isolated evaluation.

munication data. The study showed several advantages, in particular, greater efficiency of machine learning message classification as compared to a human coding procedure. So far, this study has ignored the question of whether experimenters will trust such machine learning-generated results. Penczynski (2019), for example, analyzes the influence of particular features or tokens (i.e. words, numbers, and other components of natural language text) on the performance of a Random Forest Classifier.

By doing so, the human 'judges' of the classification algorithm could use their intuition to evaluate whether they would trust the machine. The paper of Ribeiro et al. (2016) is comparable in this regard. They use a model-agnostic approach to explain the outcomes of almost every classification algorithm, among them some so-called 'black box' classifiers. Inexperienced machine learning users are tasked with testing the results of their explanation approach. Thereby, Ribeiro et al. (2016) show that the users are more trusting in the machine learning models, and have the ability to tune hyperparameters of the models if they are guided by the model-agnostic approach. Besides these technical questions associated with the implementation of machine learning in the analysis of experimental data, I see an equally great challenge in convincing experimenters that results can be trustable, even if they derive from a black-box algorithm. Democratizing artificial intelligence is one approach to tackling this problem. IBM and other companies are working on tools to foster greater understandability of machine learning models and the interpretability of their results. However as shown in this paper, expectations regarding artificial intelligence must remain realistic.

References

Agranov, M. & Yariv, L. (2018). Collusion through communication in auctions, *Games and Economic Behavior*, 107, 93-108.

Ahn, L. v. & Dabbish, L. (2004). Labeling images with a computer game. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 319-326.

Braun, D., Hernandez-Mendez, A., Matthes, F. & Langen, M. (2017). Evaluating natural language understanding services for conversational question answering systems. Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, 174-185.

Charness, G. (2006). Promises and partnership. *Econometrica,* 74(6), 1579-1601.

Chen, T. & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 785-794.

Cohen, J. (1960). A Coefficient of agreement for nominal scales. *Educational and Psychological Measurement,* 20(1), 37-46.

Deck, C., Servátka, M. & Tucker, S. (2013). An examination of the effect of messages on cooperation under double-blind and single-blind payoff procedures. *Experimental Economics,* 16(4), 597-607.

Fischer, C. & Normann, H. T. (2019). Collusion and bargaining in asymmetric Cournot duopoly - An experiment. *European Economic Review,* 111, 360-379.

Hastie, T., Friedman, J. & Tibshirani, R. (2009). The elements of statistical learning: Data mining, inference, and prediction. New York: Springer.

Houser, D. & Xiao, E. (2011). Classification of natural language messages using a coordination game. *Experimental Economics,* 14(1), 1-14.

Huang, L. & Xiao, E. (2018). Peer effects in public support for Pigouvian taxation. Working Paper.

Huerta, J. (2008). Relative rank statistics for dialog analysis. Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, 965-972.

Isaac, R. & Walker, J. (1988). Communication and free-riding behavior: The voluntary contribution mechanism. *Economic Inquiry,* 26(4), 585-608.

Ismayilov, H. & Potters, J. (2016). Why do promises affect trustworthiness, or do they? *Experimental Economics,* 19(2), 382-393.

Khan, A., Baharudin, B., Lee, L. H. & Khan, K. (2010). A review of machine learning algorithms for text-documents classification. *Journal of Advances in Information Technology,* 1(1), 4-20.

Landis, J. & Koch, G. (1977). An application of hierarchical kappa-type statistics in the assessment of majority agreement among multiple observers. *Biometrics*, 33(2), 363-374.

Lundquist, T., Ellingsen, T., Gribbe, E. & Johannesson, M. (2009). The aversion to lying. *Journal of Economic Behavior & Organization,* 70(1-2), 81-92.

Mitchell, T. (1997). Machine Learning. New York: McGraw-Hill.

Moellers, C., Normann, H. T. & Snyder, C. M. (2017). Communication in vertical markets: Experimental evidence. *International Journal of Industrial Organization,* 50, 214-258.

Nielsen, K., Bhattacharya, P., Kagel, J. & Sengupta, A. (2019). Teams promise but do not deliver. *Games and Economic Behavior,* 117, 420-432.

Penczynski, S. P. (2019). Using machine learning for communication classification. *Experimental Economics,* 22(4), 1002-1029.

Ribeiro, M., Singh, S. & Guestrin, C. (2016). Why should I trust you? Explaining the predictions of any classifier. Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 1135-1144.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computer Surveys,* 34(1), 1-47.

Snow, R., O'Connor, B., Jurafsky, D. & Ng, A. (2008). Cheap and fast - But is it good?: Evaluating non-expert annotations for natural language tasks. In Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, 254-263.

Uthus, D. & Aha, D. (2013). Multiparticipant chat analysis: A survey. *Artificial Intelligence,* (199-200), 106-121.

Appendix A: Additional Results on Deck et al. Message Corpus: Deck et al. implement a one-shot hidden action trust game of C&D in a single-blind and a double-blind payoff procedure. The single-blind payoff procedure is an exact replication of the trust game reported by C&D. However, both in the single-blind and in the double-blind payoff conditions, Deck et al. implement some slight modifications compared to C&D.[34] This is especially interesting, as I would like to know more about further scalability of a machine learning model trained by exogenous data on similar experimental results.[35] In the first stage, the procedure is the same as in the previous section, as data reported by C&D is used for the training data. Therefore, I consider a total number of 3415 training datasets. In contrast to the first series of datasets, the validation data set is not composed of the remaining messages. Instead, for all 3415 training datasets, the validation dataset is composed of the messages and classifications reported in the supplement material of Deck et al. (2013).[36] Again, all blank messages are excluded. Therefore, the validation dataset consists of 44 messages:[37]

(1) $M_{\text{Va De}} = \{m_1, \ldots, m_{44} \mid \text{all messages reported by Deck et al., except blank messages}\}$

Using the messages reported by Deck et al., the question arises whether the human classification results are comparable to C&D. I argue that this is permitted, as Deck et al. also use a weak definition of promise. Furthermore, they employ three evaluators to evaluate their messages, as well as the messages of C&D.

[34] To their surprise, Deck et al. were not able to replicate the results given by C&D. Therefore, they point out several notable differences in the procedure (besides the payoff procedures): (1) Deck et al. conducted their experiment in a lab, not in a classroom; (2) the subject pool was different, as their experiments were conducted in the south eastern US, not in California; (3) Deck et al. used a curtain to separate As from Bs; (4) they did not include elicited subjects beliefs; (5) In C&D strict preplay communication from B to A was implemented. '(…) Messages were sent before As made their decisions, and Bs roll decision was made after. The decisions were made on separate forms.' In the setup of Deck et al., '(…), roll choices were made at the top and messages could be written at the bottom of the same form. (…) While there is no way to control the order in which subjects in the B role complete the response form, it is likely that many completed the top portion first.' This way, Bs had the opportunity to send messages about what they have done or what they plan to do, whereas the Bs in the setup of C&D can send messages exclusively based upon what they plan to do.

[35] I do not discuss how to identify 'similar' experimental setups. Although, from a theoretical point of view, it is possible to automatically classify the instructions of experimental designs with a machine learning model.

[36] I used all messages sent by B in the single-blind and in the double-blind condition. Deck et al. reported 74 messages in these conditions of which 30 were blank messages without any text.

[37] As stated in the main paper, messages with the same wording, which were categorized with 'intent 1' and 'intent 2', should be excluded. I did not identify any such messages.

Agreeing on 89% (i.e., κ = 0.766) of the reported messages by C&D, the evaluators seem to have the same understanding of a weak promise statement. As with the messages from C&D, the categories from Deck et al are not adopted one to one. Deleting all blank messages, the two categories 'promise' and 'empty talk' are employed:[38] (2) $C_{De} = \{c_{1,} c_2 \mid \text{with } c_1 = \text{'promise' and } c_2 = \text{'empty talk'}\}$

Applying the machine learning training data models from the C&D messages on the message corpus reported by Deck et al. I check for interrater agreement between the classifications given by Deck et al. and the classifications given by IBM Watson Conversation. I get the following interrater agreement results for the initial corpus of messages. Of the 3,415 training datasets, in 1,743 cases the Kappa score on the validation dataset ($M_{Va\ De}$) is at least substantial (κ > 0.60). Compared to the results discussed before, where 1,259 validation data sets[39] yield a Kappa score larger than 0.6, the machine learning model seems capable of performing and generalizing better on the test data. Correspondingly, for the remaining 1,672 data sets, the Kappa score is equal to or smaller than 0.6. Within these 1,672 datasets, 1,390 datasets have either eight or fewer 'empty talk' messages or eight or fewer 'promise' messages. This is in line with the previous results, indicating that a minimum number of messages per category are needed for a valid training of the model.

Table A1a: Tobit Regression Results, All Messages Included

	(a)	**(b_e)**	**(b_p)**	**(c_e)**	**(c_p)**
#Messages_ J	0.0133*** (0.0003)	0.0137*** (0.0004)	0.0124*** (0.0005)	0.0052*** (0.0008)	0.0213*** (0.0009)
#EmptyTalk_I	--	-0.0013** (0.0006)	--	-0.0205*** (0.0016)	--
#Promise_K	--	--	0.0013** (0.0006)	--	0.0147*** (0.0012)
#Messages_ J X #EmptyTalk_I	--	--	--	0.0007*** (0.0001)	--
#Messages_ J X #Promise_K	--	--	--	--	-0.0005*** (0.0000)
Constant	0.2073*** (0.0098)	0.2108*** (0.0010)	0.2108*** (0.0010)	0.4264*** (0.0195)	0.0026 (0.0192)
Obs.	3415	3415	3415	3415	3415
LR chi^2	1394.76^a	1398.66^a	1398.66^a	1556.87^a	1554.59^a

Standard errors are reported in parentheses. * p<0.1, **p<0.05, ***p<0.01, ^{a} p < 0.001.

Table A1b: Tobit Regression Results, All Messages Included

	(d_e)	**(d_p)**	**(e_e)**	**(e_p)**
#Messages_ J	0.0012 (0.0015)	0.0223*** (0.0016)	0.0153*** (0.0023)	0.0241*** (0.0034)
#EmptyTalk_I	-0.0614*** (0.0040)	--	0.0376*** (0.0046)	--

[38] Deck et al. used the categories named 'promise', 'non-promise message' and 'blank'. They followed in their definition of the promise category C&D. As C&D define the category promise as '(…) any statement of intent (…)', Deck et al. define their 'promise category" as 'a promise or statement of intention (…)'. The 'empty talk category' from Deck et al. is defined as 'a message that is not blank, but does not contain a promise statement of intention (…)'.
As Deck et al. employed three evaluators, some messages had not been classified in the same category by all three evaluators. In these cases, the classification of the majority of the evaluators is employed.

[39] The validation data sets were composed of the remaining messages from C&D.

#Promise_K	--	0.0536*** (0.0036)	--	-0.0074* (0.0044)
#Messages_ J X #EmptyTalk_I	0.0022*** (0.0002)	--	-0.0006*** (0.0001)	--
#Messages_ J X #Promise_K	--	-0.0018*** (0.0001)	--	-0.0002 (0.0001)
Constant	0.5777*** (0.0329)	-0.1254*** (0.0338)	-0.0404 (0.0816)	0.1654 (0.1214)
Obs.	1930	1930	1485	1485
LR chi^2	527.51[a]	506.49[a]	699.77[a]	676.56[a]

Standard errors are reported in parentheses. * p<0.1, **p<0.05, ***p<0.01, [a] p < 0.001.

Kappa as dependent variable (i.e. Kappa-score overall messages, while irrelevant messages are coded as empty talk). Regarding the Tobit regression results, as shown in Table A1a and Table A1b, I observe largely similar results compared to using the messages from C&D as validation data. Again, I observe that for small training sets, Kappa benefits from including more promise messages, while in bigger training sets, including more empty talk messages is to be preferred. The regression results are also in line with the cluster results shown in Table A2. Regarding Cluster 3, the mean Kappa score indicates an 'almost perfect agreement'. Within this Cluster 3 (as in Cluster 2) I also observe that a higher share of promise messages regularly goes along with higher Kappa scores. Overfitting on the Deck et al. messages driven by promise messages in larger training sets seems to be less strong compared to the results on the C&D messages. This is supported by the finding that the coefficient of the variable indicating the number of promise messages in the model (e_p) is very small. In fact, the reason might be that there is no subset from the original training dataset and the validation dataset with the corpora of messages utilized as test datasets.

Table A2: Descriptive Statistics for Three Clusters of Training Sets, With Distinctive Ranges of Size_Training_Set

	# Training sets	**Mean**	**SD**	**Min**	**Max**
Cluster 1 – by J					
Kappa_All	945	0.52	0.21	0.01	0.92
Number_ET	945	7.85	4.78	1	20
Number_P	945	7.85	4.78	1	20
Size_Training_Set	945	15.7	3.85	8	21
Cluster 2 – by J					
Kappa_All	1420	0.66	0.15	0.02	0.95
Number_ET	1420	11.5	6.31	1	22
Number_P	1420	16.5	7.26	1	32
Size_Training_Set	1420	28	3.72	22	34
Cluster 3 – by J					
Kappa_All	1050	0.8	0.07	0.5	0.97
Number_ET	1050	15.67	4.82	3	22
Number_P	1050	25.67	4.82	13	32
Size_Training_Set	1050	41.33	4.82	35	54

Appendix B: Additional Econometric Analyses: I estimate models similar to those in Table 1 with Kappa scores calculated based on the validation-messages only, i.e. messages that have been part of the training set are excluded when calculating the agreements between that specific training set and the original coding of C&D. The results are presented in Table B1 and show that the findings discussed above are also valid if I consider Kappa scores calculated on validation data only.

Table B1a: Tobit Regression Results on Test Data Only

	(a)	(b_e)	(b_p)	(c_e)	(c_p)
#Messages_ J	0.0063*** (0.0003)	0.0076*** (0.0003)	0.0034*** (0.0004)	-0.0003 (0.0006)	0.0097*** (0.0007)
#EmptyTalk_I	--	-0.0042*** (0.0005)	--	-0.0221*** (0.0013)	--
#Promise_K	--	--	0.0042*** (0.0005)	--	0.0137*** (0.0010)
#Messages_ J X #EmptyTalk_I	--	--	--	0.0006*** (0.0000)	--
#Messages_ J X #Promise_K	--	--	--	--	-0.0003*** (0.0000)
Constant	0.3453*** (0.0079)	0.3572*** (0.0079)	0.3572*** (0.0079)	0.5580*** (0.0155)	0.2095*** (0.0153)
Obs.	3415	3415	3415	3415	3415
LR chi^2	544.95^a	611.62^a	611.62^a	828.63^a	735.55^a

Standard errors are reported in parentheses. * p<0.1, **p<0.05, ***p<0.01. *Kappa_Test* as dependent variable (i.e. training messages are excluded and irrelevant messages are coded as empty talk). ^{a} p < 0.001.

Table B1b: Tobit Regression Results on Test Data Only

	(d_e)	(d_p)	(e_e)	(e_p)
#Messages_ J	-0.0054*** (0.0012)	0.0119*** (0.0012)	0.0118*** (0.0020)	0.0050* (0.0030)
#EmptyTalk_I	-0.0600*** (0.0030)	--	0.0297*** (0.0040)	--
#Promise_K	--	0.0483*** (0.0028)	--	-0.0139*** (0.0038)
#Messages_ J X #EmptyTalk_I	0.0021*** (0.0001)	--	-0.0006*** (0.0001)	--
#Messages_ J X #Promise_K	--	-0.0015*** (0.0001)	--	0.0002 (0.0001)
Constant	0.7166*** (0.0248)	0.0782** (0.0254)	0.0486 (0.0710)	0.5779*** (0.1057)
Obs.	1930	1930	1485	1485
LR chi^2	585.29^a	504.71^a	313.10^a	285.60^a

Standard errors are reported in parentheses. * p<0.1, **p<0.05, ***p<0.01. *Kappa_Test* as dependent variable (i.e. training messages are excluded and irrelevant messages are coded as empty talk). ^{a} p < 0.001.

FE-Model C&D: The following model results in Table B2a and Table B2b depict fixed effects regressions. Within each model, fixed effects for training set size are included. The variable 'Messages_J_Rel' is defined as the absolute number of messages in the specific training set, divided by the total number of messages in the whole corpus of messages. Likewise, the variables 'Number_EmptyTalk_I_Rel' and 'Number_Promise_Rel' are defined as the absolute number of promise or empty talk messages in the specific training set, divided by the total number of promise or empty talk messages in the whole corpus of messages. The model results hold qualitatively if corresponding variables representing absolute numbers are included.

Table B2a: FE Regression Results C&D, All Messages Included

	Models Estimating the Influence of an Increasing Number of Promise Messages				
Limitations on J	--	--	--	J<31	J>30
	(a_p)	(b_{p1})	(b_{p2})	(d_p)	(e_p)
#Messages_J_Rel	0.9330*** (0.0222)	FE	--	FE	FE
#EmptyTalk_I_Rel	FE	--	FE	--	--

#Promise_K_Rel	--	0.0758*** (0.0232)	0.5529*** (0.0131)	0.4189*** (0.0353)	-0.4801*** (0.0169)
Constant	0.3357*** (0.0082)	0.6395*** (0.0086)	0.4712*** (0.0052)	0.4842*** (0.0092)	1.0174*** (0.0087)
Obs.	3415	3415	3415	1930	1485
R²	0.3714	0.2759	0.2759	0.1712	0.0238

Standard errors are reported in parentheses. * p<0.1, **p<0.05, ***p<0.01. *Kappa* as dependent variable (i.e. Kappa-score overall messages, while irrelevant messages are coded as empty talk). FE: the fixed effects are estimated for that specific variable.

Table B2b: FE Regression Results C&D, All Messages Included

	Models Estimating the Influence of an Increasing Number of Empty Talk Messages				
Limitations on J	--	--	--	J<31	J>30
	(a_e)	**(b_{e1})**	**(b_{e2})**	**(d_e)**	**(e_e)**
#Messages_J_Rel	0.8865*** (0.0223)	FE	--	FE	FE
#EmptyTalk_I_Rel	--	-0.0521*** (0.0160)	0.361*** (0.0091)	-0.288*** (0.0243)	0.3301*** (0.0117)
#Promise_K_Rel	FE	--	FE	--	--
Constant	0.3521*** (0.0081)	0.685*** (0.0062)	0.5373*** (0.0037)	0.6672*** (0.0082)	0.6299*** (0.0054)
Obs.	3415	3415	3415	1930	1485
R²	0.3714	0.0768	0.0768	0.0107	0.4795

Standard errors are reported in parentheses. * p<0.1, **p<0.05, ***p<0.01. *Kappa* as dependent variable (i.e. Kappa-score overall messages, while irrelevant messages are coded as empty talk). FE: the fixed effects are estimated for that specific variable.

The models indicate that…
(a) …overall training set sizes, Kappa increases when the number of promise messages increases, given a fixed number of empty-talk messages (Models a_p and b_{p2}),
(b) …overall training set sizes, Kappa increases when the number of empty talk messages increases, given a certain number of promise messages (Models a_e and b_{e2}),
(c) …overall training set sizes, given a certain training-set-size, substituting empty talk messages by promise-messages yields higher Kappa scores than the other way around (Models b_{p1} and b_{e1}),
(d) …with small training set sizes, result (c) holds (Model d_p and d_e), yet, with training set sizes >30, substituting empty-talk-messages by promise-messages yields lower Kappa scores than the other way around (Model e_p and e_e). The results hold when Kappa is calculated from test data only (see Table B3a and Table B3b).

Table B3a: FE Regression Results C&D on Test Data Only

	Models Estimating the Influence of an Increasing Number of Promise Messages				
Limitations on J	--	--	--	J<31	J>30
	(a_p)	**(b_{p1})**	**(b_{p2})**	**(d_p)**	**(e_p)**
#Messages_J_Rel	0.6221*** (0.0235)	FE	--	FE	FE
#EmptyTalk_I_Rel	FE	--	FE	--	--
#Promise_K_Rel	--	0.1855*** (0.0247)	0.3686*** (0.0139)	0.5440*** (0.0346)	-0.3953*** (0.0277)
Constant	0.3050*** (0.0087)	0.4599*** (0.0091)	0.3954*** (0.0056)	0.3481*** (0.0090)	0.7893*** (0.0143)
Obs.	3415	3415	3415	1930	1485

R^2	0.1475	0.1480	0.1480	0.1509	0.0244

Standard errors are reported in parentheses. * p<0.1, **p<0.05, ***p<0.01. *Kappa_Test* as dependent variable (i.e. training messages are excluded and irrelevant messages are coded as empty talk). FE: The fixed effects are estimated for that specific variable.

Table B3b: FE Regression Results C&D on Test Data Only

	Models Estimating the Influence of an Increasing Number of Empty Talk Messages				
Limitations on J	--	--	--	J<31	J>30
	(a_e)	**(b_{e1})**	**(b_{e2})**	**(d_e)**	**(e_e)**
#Messages_J_Rel	0.3899*** (0.0269)	FE	--	FE	FE
#EmptyTalk_I_Rel	--	-0.1275*** (0.0170)	0.1589*** (0.0110)	-0.374*** (0.0238)	0.2717*** (0.0190)
#Promise_K_Rel	FE	--	FE	--	--
Constant	0.3872*** (0.0097)	0.5708*** (0.0066)	0.4687*** (0.0044)	0.5857*** (0.0080)	0.4703*** (0.0088)
Obs.	3415	3415	3415	1930	1485
R^2	0.1475	0.0094	0.0094	0.0586	0.1711

Standard errors are reported in parentheses. * p<0.1, **p<0.05, ***p<0.01. *Kappa_Test* as dependent variable (i.e. training messages are excluded and irrelevant messages are coded as empty talk). FE: The fixed effects are estimated for that specific variable.

Table B4a: FE Regression Results Deck et al. All Messages Included

	Models Estimating the Influence of an Increasing Number of Promise Messages				
Limitations on J	--	--	--	J<31	J>30
	(a_p)	**(b_{p1})**	**(b_{p2})**	**(d_p)**	**(e_p)**
#Messages_J_Rel	1.0872*** (0.0291)	FE	--	FE	FE
#EmptyTalk_I_Rel	FE	--	FE	--	--
#Promise_K_Rel	--	0.0169 (0.0301)	0.6443*** (0.0172)	0.4626*** (0.0429)	-0.7053*** (0.0315)
Constant	0.2053*** (0.0108)	0.5845*** (0.0111)	0.3633*** (0.00689)	0.3785*** (0.0112)	1.0811*** (0.0163)
Obs.	3415	3415	3415	1930	1485
R^2	0.3433	0.2342	0.2342	0.1465	0.0321

Standard errors are reported in parentheses. * p<0.1, **p<0.05, ***p<0.01. *Kappa* as dependent variable (i.e. Kappa-score overall messages, while irrelevant messages are coded as empty talk). FE: the fixed effects are estimated for that specific variable.

Table B4b: FE Regression Results Deck et al. All Messages Included

	Models Estimating the Influence of an Increasing Number of Empty Talk Messages				
Limitations on J	--	--	--	J<31	J>30
	(a_e)	**(b_{e1})**	**(b_{e2})**	**(d_e)**	**(e_e)**
#Messages_J_Rel	1.1362*** (0.0355)	FE	--	FE	FE
#EmptyTalk_I_Rel	--	-0.0116 (0.0207)	0.4629*** (0.0145)	-0.318*** (0.0295)	0.4849*** (0.0217)
#Promise_K_Rel	FE	--	FE	--	--

Constant	0.1880*** (0.0129)	0.5946*** (0.0081)	0.4254*** (0.0058)	0.5806*** (0.0099)	0.5119*** (0.0100)
Obs.	3415	3415	3415	1930	1485
R²	0.3433	0.0882	0.0882	0.0076	0.3508

Standard errors are reported in parentheses. * p<0.1, **p<0.05, ***p<0.01. *Kappa* as dependent variable (i.e. Kappa-score overall messages, while irrelevant messages are coded as empty talk). FE: the fixed effects are estimated for that specific variable.

The models indicate that…

(a) …overall training set sizes, Kappa increases when the number of promise messages increases, given a fixed number of empty-talk messages (Models a_p and b_{p2}),

(b) …overall training set sizes, Kappa increases when the number of empty talk messages increases, given a certain number of promise messages (Models a_e and b_{e2}),

(c) …overall training set sizes, Kappa neither significantly increases when empty talk messages are substituted by promise messages, nor when messages are substituted the other way around (Models a_e and b_{e2}),

(d) …with small training set sizes, given a certain training set size, substituting empty talk messages with promise messages yields higher Kappa-scores than the other way around (Model d_p and d_e), yet, with training set sizes >30, substituting empty talk messages by promise messages yields lower Kappa-scores than the other way around (Model e_p and e_e). That is, most of the qualitative results derived from FE models on C&D messages also hold when considering Deck et al. messages. The only difference is that over the whole range of training sets, there is no clear preference for including either more empty talk or more promise messages, given a certain size of the training set. Yet, there is a clear preference for either of the two categories when considering different ranges of training set sizes.

Appendix C: Additional Results on MLP and Random Forest Classifier: Table C1 shows the mean Kappa score (over five datasets) on the MLP Classifier for the whole message corpus. I employed the same five training data sets for each stratification of the training data corpus of the categories 'empty talk' and 'promise' as in the main section of this paper while employing IBM Watson Conversation.

Table C1: Mean Kappa-Scores on Whole Corpus C&D Using MLP Classifier

	No. messages empty talk in the training data set																				
No. messages promise in training data set	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**
2	-	-	-	-	0.44	0.39	0.45	0.34	0.36	0.35	0.36	0.39	0.28	0.51	0.37	0.35	0.26	0.36	0.27	0.16	0.21
3	-	-	-	0.58	0.61	0.56	0.48	0.48	0.48	0.54	0.52	0.44	0.30	0.36	0.35	0.46	0.26	0.33	0.29	0.39	0.29
4	-	-	0.58	0.47	0.58	0.64	0.57	0.49	0.55	0.51	0.63	0.64	0.48	0.39	0.56	0.51	0.37	0.47	0.42	0.52	0.49
5	-	0.41	0.44	0.49	0.65	0.56	0.59	0.60	0.58	0.51	0.46	0.64	0.55	0.57	0.55	0.65	0.60	0.52	0.62	0.47	0.49
6	0.40	0.41	0.57	0.61	0.65	0.69	0.66	0.57	0.63	0.66	0.44	0.67	0.57	0.55	0.47	0.52	0.60	0.64	0.61	0.48	0.62
7	0.26	0.49	0.44	0.66	0.65	0.59	0.64	0.57	0.61	0.65	0.66	0.72	0.68	0.63	0.56	0.66	0.59	0.65	0.66	0.51	0.67
8	0.10	0.49	0.64	0.50	0.71	0.65	0.68	0.71	0.67	0.67	0.66	0.71	0.71	0.61	0.72	0.65	0.63	0.64	0.60	0.60	0.61
9	0.24	0.56	0.50	0.67	0.66	0.54	0.68	0.69	0.77	0.68	0.71	0.72	0.67	0.73	0.64	0.71	0.68	0.69	0.59	0.73	0.67
10	0.17	0.61	0.49	0.54	0.63	0.68	0.69	0.69	0.67	0.76	0.63	0.67	0.69	0.77	0.65	0.65	0.71	0.73	0.71	0.64	0.68
11	0.39	0.53	0.55	0.52	0.67	0.70	0.62	0.66	0.72	0.77	0.72	0.67	0.67	0.65	0.77	0.69	0.72	0.78	0.75	0.68	0.75
1	0.2	0.3	0.4	0.6	0.5	0.6	0.6	0.6	0.6	0.7	0.7	0.6	0.7	0.5	0.7	0.7	0.6	0.6	0.7	0.7	0.6

2	9	4	4	2	5	6	8	7	9	2	1	5	8	9	5	6	8	9	0	6	5
13	0.19	0.43	0.34	0.51	0.61	0.50	0.70	0.73	0.73	0.75	0.68	0.79	0.73	0.78	0.76	0.73	0.76	0.69	0.76	0.75	0.73
14	0.27	0.54	0.53	0.57	0.53	0.67	0.68	0.74	0.67	0.72	0.74	0.75	0.75	0.72	0.73	0.77	0.73	0.78	0.72	0.76	0.73
15	0.11	0.15	0.37	0.65	0.66	0.59	0.67	0.74	0.71	0.70	0.76	0.80	0.74	0.73	0.80	0.79	0.77	0.79	0.77	0.78	0.78
16	0.33	0.33	0.28	0.64	0.55	0.65	0.69	0.66	0.71	0.69	0.61	0.56	0.78	0.68	0.78	0.75	0.80	0.72	0.78	0.81	0.81
17	0.12	0.30	0.58	0.63	0.55	0.76	0.72	0.61	0.55	0.72	0.78	0.75	0.74	0.83	0.79	0.79	0.81	0.78	0.78	0.78	0.79
18	0.18	0.42	0.51	0.56	0.55	0.59	0.67	0.76	0.70	0.74	0.76	0.79	0.74	0.72	0.75	0.74	0.79	0.80	0.80	0.85	0.80
19	0.26	0.27	0.37	0.68	0.57	0.66	0.72	0.71	0.65	0.77	0.78	0.76	0.73	0.79	0.78	0.79	0.81	0.79	0.77	0.83	0.81
20	0.10	0.37	0.54	0.47	0.68	0.60	0.68	0.73	0.74	0.74	0.79	0.79	0.70	0.79	0.79	0.81	0.81	0.78	0.75	0.80	0.82
21	0.32	0.35	0.56	0.56	0.58	0.69	0.68	0.72	0.77	0.71	0.67	0.76	0.79	0.81	0.76	0.81	0.78	0.83	0.84	0.84	0.80
22	0.15	0.32	0.44	0.56	0.54	0.64	0.72	0.76	0.71	0.74	0.68	0.81	0.71	0.77	0.80	0.82	0.80	0.79	0.80	0.80	0.65
23	0.19	0.37	0.41	0.57	0.63	0.57	0.72	0.70	0.73	0.71	0.79	0.79	0.76	0.78	0.83	0.79	0.83	0.80	0.82	0.79	0.84
24	0.26	0.39	0.46	0.58	0.54	0.63	0.57	0.61	0.75	0.73	0.68	0.75	0.78	0.80	0.80	0.81	0.80	0.80	0.80	0.87	0.81
25	0.28	0.30	0.52	0.52	0.55	0.65	0.65	0.71	0.72	0.70	0.73	0.70	0.80	0.83	0.79	0.80	0.81	0.82	0.79	0.81	0.79
26	0.30	0.25	0.42	0.63	0.64	0.58	0.65	0.61	0.72	0.75	0.73	0.77	0.78	0.83	0.84	0.82	0.82	0.81	0.85	0.85	0.81
27	0.17	0.42	0.50	0.54	0.54	0.50	0.62	0.63	0.79	0.71	0.72	0.75	0.75	0.81	0.86	0.80	0.81	0.83	0.84	0.87	0.81
28	0.12	0.27	0.43	0.53	0.49	0.62	0.67	0.75	0.74	0.72	0.76	0.76	0.77	0.77	0.83	0.86	0.83	0.80	0.84	0.86	0.83
29	0.11	0.36	0.40	0.48	0.58	0.55	0.61	0.70	0.72	0.70	0.81	0.79	0.76	0.72	0.82	0.78	0.80	0.80	0.85	0.87	0.78
30	0.16	0.30	0.39	0.58	0.60	0.61	0.62	0.64	0.71	0.57	0.76	0.73	0.81	0.79	0.81	0.81	0.77	0.79	0.84	0.89	0.87
31	0.20	0.41	0.47	0.50	0.45	0.69	0.65	0.67	0.69	0.76	0.68	0.61	0.78	0.74	0.84	0.85	0.83	0.84	0.81	0.87	0.88
32	0.17	0.26	0.41	0.52	0.57	0.65	0.62	0.73	0.71	0.69	0.80	0.73	0.83	0.81	0.81	0.81	0.82	0.87	0.87	0.86	0.86

Table C2 shows the mean Kappa score (over five datasets) on the MLP Classifier for the training message corpus. I employed the same five training data sets for each stratification of the training data corpus of the categories 'empty talk' and 'promise' as in the main section of this paper while employing IBM Watson Conversation.

Table C2: Mean Kappa-Scores on Training Data C&D Using MLP Classifier

No. messages promise in training data set	No. messages empty talk in the training data set																				
	2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**
2	-	-	-	-	1.00	0.80	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.80	1.00
3	-	-	-	1.00	1.00	1.00	1.00	1.00	0.96	1.00	1.00	1.00	1.00	0.80	0.93	1.00	1.00	1.00	1.00	1.00	1.00
4	-	-	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
5	-	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.80	1.00	0.93	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
6	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.80	1.00	1.00	1.00	0.80	1.00	0.96	0.98	0.96	1.00	1.00
7	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.80	1.00
8	0.80	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.98
9	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
10	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.94	1.00	1.00	1.00
11	1.00	1.00	1.00	1.00	0.91	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
12	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.80	1.00	1.00	0.96	1.00	1.00	0.99	1.00
13	1.00	1.00	0.80	1.00	1.00	0.80	1.00	1.00	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
14	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
15	1.00	0.60	1.00	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
16	1.00	1.00	0.80	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.80	0.80	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.98	1.00
17	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.80	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
18	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.98	1.00	1.00	1.00	0.97	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00
19	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00	1.00	0.99
20	0.73	1.00	1.00	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
21	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.98	1.00	1.00	1.00	1.00	1.00	0.99
22	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.80
23	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.97	1.00	1.00	1.00	0.99	1.00	1.00	0.99	1.00	1.00	1.00
24	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99
25	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
26	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

27	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00
28	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00	1.00
29	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00
30	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.80	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00	1.00
31	0.96	1.00	1.00	1.00	0.80	1.00	1.00	1.00	1.00	0.99	1.00	0.80	1.00	1.00	1.00	1.00	1.00	0.98	1.00	1.00	0.99
32	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Table C3 shows the mean Kappa score (over five datasets) on the Random Forest Classifier for the whole message corpus. I employed the same five training data sets for each stratification of the training data corpus of the categories 'empty talk' and 'promise' as in the main section of this paper while employing IBM Watson Conversation.

Table C3: Mean Kappa-Scores on Whole Corpus C&D Using Random Forest Classifier

	No. messages empty talk in the training data set																				
No. messages promise in training data set	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**
2	-	-	-	-	0.10	0.05	0.05	0.05	0.18	0.05	0.06	0.11	0.03	0.13	0.05	0.08	0.11	0.10	0.07	0.04	0.04
3	-	-	-	0.44	0.29	0.43	0.15	0.25	0.22	0.16	0.16	0.25	0.09	0.17	0.15	0.20	0.12	0.13	0.15	0.17	0.09
4	-	-	0.49	0.47	0.46	0.35	0.49	0.25	0.27	0.30	0.32	0.26	0.32	0.28	0.20	0.34	0.12	0.13	0.15	0.16	0.22
5	-	0.40	0.58	0.64	0.75	0.62	0.40	0.34	0.47	0.43	0.41	0.47	0.40	0.13	0.29	0.26	0.22	0.32	0.17	0.24	0.25
6	0.21	0.47	0.58	0.65	0.75	0.59	0.62	0.53	0.57	0.62	0.51	0.29	0.42	0.32	0.35	0.26	0.34	0.38	0.18	0.38	0.47
7	0.20	0.49	0.49	0.71	0.58	0.74	0.67	0.62	0.46	0.61	0.68	0.68	0.50	0.50	0.36	0.59	0.39	0.39	0.44	0.42	0.47
8	0.19	0.43	0.31	0.69	0.75	0.74	0.76	0.72	0.72	0.48	0.67	0.73	0.58	0.50	0.65	0.59	0.54	0.60	0.53	0.43	0.43
9	0.24	0.13	0.67	0.62	0.72	0.75	0.76	0.76	0.73	0.70	0.74	0.74	0.69	0.63	0.65	0.57	0.56	0.63	0.59	0.57	0.48
10	0.08	0.27	0.20	0.59	0.72	0.80	0.73	0.78	0.77	0.81	0.74	0.74	0.71	0.75	0.69	0.65	0.66	0.69	0.73	0.61	0.63
11	0.08	0.32	0.39	0.59	0.70	0.75	0.78	0.74	0.82	0.83	0.75	0.68	0.78	0.72	0.78	0.79	0.74	0.68	0.63	0.66	0.69
12	0.11	0.25	0.36	0.64	0.61	0.66	0.77	0.77	0.76	0.79	0.80	0.72	0.72	0.77	0.75	0.78	0.70	0.66	0.69	0.72	0.62
13	0.09	0.13	0.42	0.57	0.65	0.75	0.74	0.80	0.79	0.82	0.78	0.80	0.80	0.83	0.75	0.77	0.78	0.78	0.76	0.70	0.70
14	0.07	0.22	0.36	0.58	0.62	0.73	0.74	0.77	0.81	0.81	0.80	0.82	0.84	0.80	0.78	0.78	0.76	0.80	0.72	0.77	0.75
15	0.28	0.13	0.33	0.35	0.70	0.74	0.76	0.78	0.83	0.81	0.81	0.85	0.79	0.79	0.81	0.86	0.83	0.84	0.77	0.81	0.80
16	0.08	0.20	0.35	0.39	0.56	0.77	0.76	0.77	0.78	0.82	0.81	0.83	0.87	0.81	0.86	0.82	0.75	0.80	0.78	0.86	0.87
17	0.09	0.13	0.41	0.42	0.61	0.71	0.79	0.75	0.82	0.78	0.80	0.81	0.83	0.84	0.85	0.84	0.86	0.84	0.84	0.80	0.84

18	0.08	0.37	0.25	0.41	0.47	0.68	0.80	0.80	0.77	0.82	0.82	0.84	0.82	0.83	0.87	0.83	0.84	0.85	0.84	0.85	0.86
19	0.07	0.11	0.27	0.47	0.54	0.67	0.75	0.77	0.77	0.79	0.85	0.83	0.84	0.86	0.87	0.84	0.85	0.86	0.84	0.84	0.85
20	0.07	0.30	0.31	0.62	0.63	0.55	0.74	0.73	0.79	0.81	0.84	0.82	0.81	0.85	0.84	0.84	0.84	0.84	0.86	0.86	0.86
21	0.09	0.23	0.31	0.34	0.37	0.63	0.79	0.73	0.77	0.81	0.83	0.80	0.86	0.84	0.85	0.85	0.85	0.87	0.90	0.88	0.86
22	0.08	0.13	0.19	0.48	0.58	0.66	0.75	0.75	0.79	0.82	0.80	0.85	0.83	0.84	0.87	0.85	0.89	0.87	0.88	0.88	0.89
23	0.07	0.16	0.24	0.34	0.55	0.72	0.73	0.69	0.76	0.80	0.84	0.87	0.85	0.85	0.87	0.85	0.89	0.86	0.90	0.87	0.90
24	0.16	0.24	0.14	0.45	0.43	0.62	0.72	0.81	0.80	0.81	0.84	0.83	0.84	0.86	0.86	0.88	0.89	0.87	0.87	0.91	0.88
25	0.09	0.11	0.21	0.26	0.46	0.64	0.67	0.71	0.77	0.78	0.82	0.82	0.84	0.86	0.89	0.88	0.86	0.90	0.92	0.90	0.86
26	0.16	0.17	0.15	0.33	0.64	0.51	0.67	0.77	0.79	0.72	0.81	0.85	0.85	0.86	0.88	0.91	0.89	0.88	0.91	0.89	0.90
27	0.08	0.15	0.16	0.28	0.43	0.59	0.77	0.64	0.78	0.80	0.78	0.83	0.83	0.86	0.89	0.85	0.87	0.93	0.89	0.88	0.86
28	0.10	0.11	0.21	0.41	0.52	0.65	0.70	0.77	0.76	0.81	0.74	0.80	0.86	0.85	0.90	0.90	0.86	0.88	0.93	0.93	0.89
29	0.07	0.13	0.15	0.60	0.37	0.55	0.61	0.70	0.79	0.86	0.83	0.87	0.87	0.80	0.86	0.85	0.87	0.86	0.88	0.92	0.88
30	0.07	0.17	0.16	0.22	0.61	0.67	0.60	0.72	0.80	0.81	0.82	0.84	0.85	0.85	0.85	0.87	0.85	0.90	0.91	0.93	0.90
31	0.08	0.18	0.25	0.41	0.50	0.60	0.75	0.70	0.79	0.79	0.80	0.83	0.88	0.83	0.86	0.90	0.90	0.89	0.89	0.92	0.91
32	0.09	0.13	0.16	0.48	0.42	0.64	0.61	0.65	0.78	0.77	0.84	0.85	0.86	0.88	0.89	0.87	0.89	0.88	0.90	0.91	0.90

Table C4 shows the mean Kappa score (over five datasets) on the Random Forest Classifier for the training message corpus. I employed the same five training data sets for each stratification of the training data corpus of the categories 'empty talk' and 'promise' as in the main section of this paper while employing IBM Watson Conversation.

Table C4: Mean Kappa-Scores on Training Data C&D Using Random Forest Classifier

		No. messages empty talk in the training data set																				
No. messages promise in training data set		**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**
	2	-	-	-	-	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	3	-	-	-	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	4	-	-	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	5	-	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	6	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	7	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	8	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

9	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
10	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
11	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
12	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
13	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
14	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
15	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
16	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
17	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
18	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
19	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
20	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
21	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
22	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
23	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
24	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
25	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
26	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
27	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
28	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
29	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
30	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
31	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
32	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00