Coding Negotiations with AI: Instructions and Validation for Coding Model 1

Ray Friedman[a1], Jeanne Brett[b13], Jaewoo Cho[a21], Xuhui Zhan[a21], Ningyu Han[a2], Sriram Kannan[a2], Yingxiang Ma[a2], Jesse Spencer-Smith[a2], Elisabeth Jackel[c3], Alfred Zerres[c3], Madison Hooper[a56], Katie Babbit[a6], Manish Acharya[a7], Wendi Adair[ii4], Soroush Aslani[g2], Tayfun Aykac[pp4], Chris Bauman[d4], Rebecca Bennett[ff4], Garrett Brady[x4], Peggy Briggs[o4], Cheryl Dowie[e4], Chase Eck[4], Igmar Geiger[f4], Frank Jacob[pp4], Molly Kern[cc4], Sujin Lee[bb4], Leigh Anne Liu[u4], Wu Liu[kk4], Jeffrey Loewenstein[v4], Anne Lytle[p4], Li Ma[w4], Michel Mann[s4], Alexandra Mislin[nn4], Tyree Mitchell[n4], Hannah Martensen née Nagler[pp4], Amit Nandkeolyar[t4], Mara Olekalns[z4], Elena Paliakova[h4], Jennifer Parlamis[jj4], Jason Pierce[i4], Nancy Pierce[gg4], Robin Pinkley[mm4], Nathalie Prime[pp4], Jimena Ramirez-Marin[h4], Kevin Rockmann[q4], William Ross[o4], Zhaleh Semnani-Azad[ee4], Juliana Schroeder[r4], Philip Smith[z4], Elena Stimmer[pp4], Roderick Swaab[j4], Leigh Thompson[hh4], Zhaleh Thompson[hh4], Cathy Tinsley[k4], Ece Tuncel[l4], Laurie Weingart[y4], Robert Wilken[m4], JingJing Yao[dd4], and Zhi-Xue Zhang[w4]

[1] Project leads.
[2] Data scientists who built the AI model.
[3] Negotiation coding experts.
[4] Data contributors.
[5] Statistical analyst.
[6] Coders.
[7] Website designer.
[a] Vanderbilt University
[b] Negotiation and Team Resources (NTR)
[c] University of Amsterdam
[d] University of California, Irvine
[e] University of Houston
[f] Aalen University
[g] University of Wisconsin, Whitewater
[h] IESEG School of Management
[i] UNC, Greensboro
[j] INSEAD
[k] Georgetown University

[l]Webster University
[m]ESPC Business School
[n]Louisiana State University
[o]University of Wisconsin, LaCross
[p]Anne Lytle & Associates
[q]George Mason University
[r]University of California, Berkley
[s]Leuphana University
[t]Indian Institute of Management Ahmedabad
[u]Georgia State University
[v]University of Illinois
[w]Guanghua School of Management Peking University
[x]Bocconi University
[y]Carnegie Mellon
[z]University of Melbourne
[bb]KAIST
[cc]City University of New York
[dd]IESEG School of Management
[ee]California State University, Northridge
[ff]University of Central Florida
[gg]University of North Carolina at Greensboro
[hh]Norhwestern University
[ii]University of Waterloo
[jj]University of San Fancisco
[kk]Hong Kong Polytechnic University
[mm]Southern Methodist University
[nn]American University
[pp]ESCP Business School

January 13, 2026

Abstract

This paper provides guidance for scholars wanting to use AI negotiation transcript coding model 1 of the Vanderbilt AI Negotiation Lab. This includes a link to the Lab's website, a description of model 1's codes and coding process, and a report of tests that support the model's validity. It also includes instructions for how to set up and submit your data. When reporting your results from this model, please cite this paper.

Introduction

The Vanderbilt AI Negotiation Lab was created to provide AI tools for negotiation researchers. This includes AI-assisted coding of negotiation transcripts, in English or other major languages. You can submit your transcripts for coding here: `https://AINegotiationLab-Vanderbilt.com/`. The site provides access to several coding models. For each model, the site provides: definitions of the codes, examples of each code, details of how the AI model was developed and validated, and instructions for how to format and submit your transcripts for coding. That information is also contained in this document.

These models were trained in most cases by learning from transcripts that have been human coded for specific prior projects (a full review of the development of this coding process can be found in Friedman et al, 2024). Each project used a different coding scheme, and even when codes were the same, they may have been interpreted slightly differently. The example sentences provided on the website allow you to see how each project used their codes, so you can decide which model fits your research needs.

We plan to add more coding models and to update models as needed. If you have a set of coded transcripts for a coding scheme that you would like us to include, please contact us.

Model 1 – Aslani et al.(2014)

Model 1 uses the coding scheme of Aslani et al (2014), "Measuring negotiation strategy and predicting outcomes: Self-reports, behavioral codes, and linguistic codes," presented at the annual conference of the International Association for Conflict Management, Leiden, The Netherlands[1] (a link to this paper is available on the website).

As the authors describe in the 2014 paper, "we developed a 14-item code based on prior negotiation coding schemes (e.g. Adair & Brett, 2005; Gunia et al., 2011; Weingart et al., 1990) to measure participants' use of tactics. Major categories in the code were information, offers, substantiation, negative and positive reactions, and a miscellaneous category." (This project was later expanded and published as Aslani et al, 2022, but did not use these transcript codes.) The simulation used in their study was *The Sweet Shop* (new.negotiationexercises.com). The authors shared with us 75 transcripts they had collected and coded using human coders.

Each of the codes is shown on our website except for "Response Embarrassed" which appeared only once across all transcripts, making it hard to teach that code to the model. For each of the remaining 13 codes, we provide a definition of the code, a short explanation, and sample sentences from the transcripts. The sample sentences let you see how these scholars operationalized their codes, which is what Model 1 learned from and tries to reproduce when coding your transcripts. As with any coding scheme, different scholars might operationalize concepts slightly differently. You should decide if this coding scheme will be useful to you by reviewing how the authors used it.

Coding Process

Transcripts are coded in three steps.

1. **Unitization (you need to do this).** The model provides one code for each set of words or sentences that you identify as a unit in your Excel document. You can choose to have units be speaking turn, sentences, or thought units. The easiest to set up is speaking turns, since

switching between speakers is clearly identifiable in transcripts. The next easiest is sentences, since they are identified by one of these symbols: .?! However, different transcribers may end sentences in different places. The hardest unit to create is the thought unit since that takes careful analysis and can represent as much work as the coding itself. (See the NegotiAct coding manual for how to create thought units; Jackel et al, 2022[10]). Clarity of meaning runs the opposite direction. The longer the unit, the more likely there are multiple ideas in the unit, and less clarity for human or AI coders to know what part to code. Aslani et al (2014)[1] coded speaking turns, but 72% of their speaking turns contained just one sentence. The closest alignment with the training data would be for you to use sentences as the unit.

2. **Model Assigns Codes.** The model assigns a code to each unit you submit, based on in-context learning. Coding is guided by the prompt we developed and tested. For more on in-context learning see Xie and Min (2022)[14]. Our prompt for this model includes several elements:

   (a) Five fully coded transcripts. These transcripts were chosen from the 75 available transcripts in the following way. First, any combination of five was considered only if that set included all 13 codes. Second, five of those combinations were chosen at random to test. Third, the one that produced the highest level of match with human coders was retained.

   (b) Instructions to pay attention to who was speaking, such as “buyer” or “seller”.

   (c) Instructions to pay attention to what was said in the conversation before and after the unit being coded.

   (d) Supplementary instructions about the difference between “substantiation” and “information” since in early tests the model often coded substantiation as information, and vice versa. This confusion is not surprising since substantiation usually comes in the form of providing information, but with the purpose of supporting a specific offer or demand.

   (e) Additional examples of any codes where the five training transcripts did not contain at least 15 examples. We created enough additional examples (based on our understanding of the code) to bring the examples up to 15. We needed to add 12 examples of multi-issue offer, 12 examples of offer rejected, and 14 examples of Miscellaneous Off-Task.

3. **We Run the Model Five Times.** We automatically run the model five times, to assess consistency of results. As expected the results are not always the same, since with in-context learning the model learns anew with each run and may learn slightly differently each time. Variation is also expected since some units may reasonably be coded in several ways. By running the coding model five times, we get five codes assigned to each speaking unit. If three, four, or five of the five of the runs have the same code, we report the code and indicate the level of “consistency” of that code (three, four, or five out of five). If there are not at least three consistent results out of five runs, or if the model fails to assign a code, we do not report a model code. In these cases, the researcher needs to do human coding.

Validation of Model

This section reports the initial validation, using Claude Opus 3.0. This version of Opus was decommissioned on 31st December 2025, and replaced in our model with Opus 4.5. Updated validation of key measures are reported at the end of this section, and show equivalent restults for Opus 4.5.

Validation occurred in several steps.

Step 1: Compare the model coding with humans by Aslani et al. (2014)[1]. To do this, we asked the model to code the 4968 units contained in the Aslani et al. (2014)[1] transcripts that were not selected for training. We looked at several criteria.

(a) **Consistency.** In our test, there was "perfect" consistency of model coding (five out of five runs of the model assigned the same code to a unit) for 4,752 of the units, "high" consistency (four out of five) for 126 of the units, "modest" consistency (three out of five) for two of the units, and 90 where the model did not report a code. Thus, 96% of codes had perfect consistency.

(b) **Match with human coding.** We assessed whether the model assigned the same code as the human coders. The overall match level for units where the model assigned a code was 73% (95% CI: .72, .75). To ensure that the model was not biased by matching more accurately with the human coder in early or later phases of the negotiation, we tested whether the match level was different for coding the first versus second half of all transcripts. The match level was 74% for the first half and 72% for the second half, suggesting no bias based on phase of the negotiation. We also looked at match by level of consistency (see Table 1). These results suggest that users may want to accept model-assigned codes only for those codes where the model achieves perfect consistency (five out of five).

Table 1: Match Percentage by Consistency Level, Validation Step 1

Level of Consistency	**Match with Human Codes**	**% Achieve This Consistency Level Among Those Assigned a Code**	**Number**	**Match**	**Percentage Match**
Modest Consistency	3 out of 5	.04%	0	not match	
			2	match	100%
High Consistency	4 out of 5	2.5%	82	not match	
			44	match	35%
Perfect Consistency	5 out of 5	97.5%	1,239	not match	
			3,513	match	74%

*90 cases did not reach the 3 out of 5 consistency threshold or the model failed to assign a code

We also calculated the Cohen's kappa, with the model codes as coming from one rater and the human coding as coming from a second rater. This calculation, compared to the "percentage match," accounts for matches that might occur based on chance. Cohen's kappa was calculated in R (R Core Team, 2022[12]) using the IRR package (Gamer & Lemon, 2019[7]). Cohen's kappa was equal to 0.69, with the no information rate of .27 (p-value of difference is smaller than .001). According to Landis and Koch (1977) this represents "substantial agreement", and according to

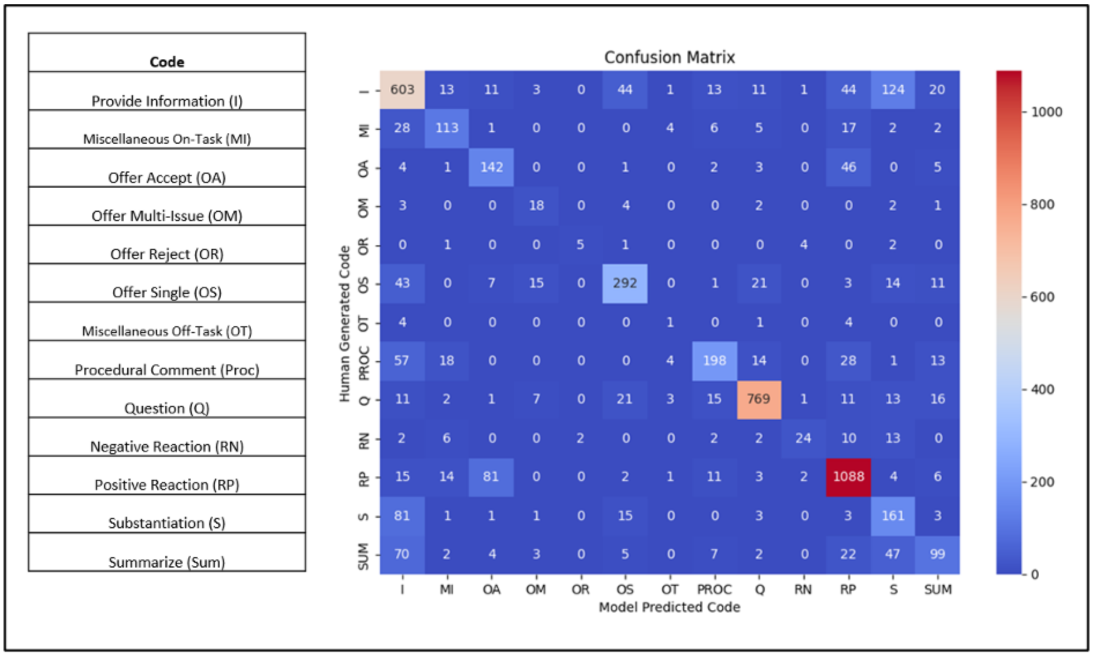
Fleiss (1981)[5] is "fair to good" agreement. Rather than relying on conventional categorical guidelines to interpret the magnitude of kappa, Bakeman (2023) argues that researchers should estimate observer accuracy or how accurate simulated observers need to be to produce a given value of kappa. The KappaAcc program (Bakeman, 2022[4]) was used to estimate observer accuracy, which was found to be 85%.

It is also worth noting that in many cases where scholars establish inter-coder reliability, there is a process of cross-rater discussion that is used to resolve initial differences of opinion between the two coders. In a study of inter-coder agreement, initial coder agreements in the 80% range began with coder agreement in the 40% range (Garrison, et al 2005[8]). Of course, in our case there can be no cross-rater discussion between a model and a human, taking away one step that is often used to achieve higher kappas. The closest we can get to that process is to have a third person view the cases of human-model disagreement to provide a judgement of which code was more correct. Also, the fact that so many codes need human-to-human discussions to resolve, suggests some inherent ambiguity about code assignments and opens up the possibility that several different codes might reasonably be assigned to some segments of transcripts.

(c) **Summary Data and Confusion Matrix.**We created a confusion matrix for all codes with perfect consistency (see Figure 1). The vertical axis shows human coding. The horizontal axis shows model coding. Also included below (see Table 2) are summary statistics showing which codes appeared most frequently in the human coding (Positive Response was most common representing 25.37% of the codes, while Miscellaneous Off-Topic was least common representing just .25% of the codes), and the human-model match level for each code. The highest level of human-model match was for Positive Response and Question while the lowest was for Miscellaneous Off Topic, Negative Responses, and Summarize. There appears to be a rough correlation between number of units and match percentage, suggesting that match percentage goes up when there are more examples of a code in the training transcript for the model to learn from and when there are more opportunities to find that code in the test transcripts. Miscellaneous Off Topic has the very lowest match percentage, perhaps because there are only ten of these units in the test transcripts, because it is a more ambiguous category, and/or because it was less conceptually central to Aslani et al and their coders.

In terms of absolute numbers of mismatches, the largest set is 124 human-coded Information codes that were coded as Substantiation by the model. This is an issue we recognized early in our testing, which resulted in added instructions in the prompt to reduce this mismatch. The fundamental problem is that Substantiation is often achieved by providing information, but to be Substantiation that information must support a particular argument or claim. There were also 81 cases of human-coded Substantiation that were coded as Information by the model. The next largest set of mismatches were 81 where humans assigned a code of Positive Responses while the model assigned a code of Accepting Offer, which is easy to imagine happening.

Closer Look at Mismatches: Mismatch Analysis. To assess the nature of these and other mismatches, we selected a random sample of 100 mismatches for closer examination. Given that original human coders may be just as likely to make errors (or simply vary in their judgments) as the model, we wanted to see if newly trained coders would see the Aslani-provided codes or the model-provided codes as more accurate. We trained two coders, who practiced coding transcripts until they reached a high level of agreement (kappa=.81). Then we provided these coders with the 100 speaking turns, as well as the two speaking turns preceding that speaking turn, along with the human and model codes. They were not informed which code came from the model or humans, and the order in which they saw the two codes was flipped halfway through the 100 samples to avoid order effects. They selected which code they saw as more accurate. This was done first separately by the two coders, and then they were asked to resolve through discussion any cases where they disagreed. In the end, these new coders thought the model-provided codes were correct 68% of the time and the human-generated codes 32% of the time. Based on this we can expect that the model is correct in 68% of the cases with mismatches, so we can trust that about 91% of the model codes are accurate.



Code
Provide Information (I)
Miscellaneous On-Task (MI)
Offer Accept (OA)
Offer Multi-Issue (OM)
Offer Reject (OR)
Offer Single (OS)
Miscellaneous Off-Task (OT)
Procedural Comment (Proc)
Question (Q)
Negative Reaction (RN)
Positive Reaction (RP)
Substantiation (S)
Summarize (Sum)

Confusion Matrix

Human Generated Code	I	MI	OA	OM	OR	OS	OT	PROC	Q	RN	RP	S	SUM
I	603	13	11	3	0	44	1	13	11	1	44	124	20
MI	28	113	1	0	0	0	4	6	5	0	17	2	2
OA	4	1	142	0	0	1	0	2	3	0	46	0	5
OM	3	0	0	18	0	4	0	0	2	0	0	2	1
OR	0	1	0	0	5	1	0	0	0	4	0	2	0
OS	43	0	7	15	0	292	0	1	21	0	3	14	11
OT	4	0	0	0	0	0	1	0	1	0	4	0	0
PROC	57	18	0	0	0	0	4	198	14	0	28	1	13
Q	11	2	1	7	0	21	3	15	769	1	11	13	16
RN	2	6	0	0	2	0	0	2	2	24	10	13	0
RP	15	14	81	0	0	2	1	11	3	2	1088	4	6
S	81	1	1	1	0	15	0	0	3	0	3	161	3
SUM	70	2	4	3	0	5	0	7	2	0	22	47	99

Figure 1: Confusion Matrix, Validation Step 1

Human Code	**% of units Across All Transcripts**	**Model Match %**
RP	25.37	88.45%
I	18.79	66.96%
Q	18.46	86.46%
OS	8.59	70.41%
PROC	7.13	57.76%
S	5.64	59.27%
SUM	5.49	36.94%
OA	4.28	69.38%
MI	3.79	63.78%
RN	1.33	40.00%
OM	0.61	60.00%
OR	0.27	38.46%
OT	0.25	8.33%

Table 2: Mach Percentage by Code, Validation Step 1

Step 2: Match with Human coding for Different Simulations.

The first step of validation involved matching human and model codes where the negotiation simulation used for training was the same as the negotiation simulation used for testing the model (The Sweet Shop). But users may have transcripts from any number of simulations or real-world negotiations, not just the simulation used in the Aslani et al (2014)[1] study. Therefore, we wanted to test how well the model would match human coders who applied the Aslani et al model to transcripts using other simulations. We selected a set of 6 Transcripts from a study that used the Cartoon simulation, and 6 transcripts from a study that used the Les Florets simulation. Since these transcripts were not initially coded using the Aslani codes, we needed to train two coders to use the Aslani codes. After initial training, they reached a level of inter-coder reliability of k=.81. They coded the transcripts separately and came together to discuss any cases where they disagreed and assign a code. This provided the human codes for a set of Cartoon and LesFlorets simulations. These transcripts were then coded using our model. The 12 transcripts had 2711 speaking turns, of which 2679 were single sentences. The model had perfect consistency for 87.85 of the speaking turns, high consistency for 10.1% of the speaking turns, and modest consistency for .02% of the speaking turns. There were 44 cases of less than 3 out of 5 consistency. The match percentage was 72% for high consistency codes, and 65% for perfect consistency codes (see Table 3). Overall, the match percentage was 65% (95% CI: .63, .67). This was lower than our prior tests, as expected, because these transcripts did not have the same issues and topics as training transcripts (which used The Sweet Shop simulation). For that reason, these results may better represent the model's effectiveness with most transcripts. We also checked to see if one set of transcripts did better than the other. The match percentage was also 64.90% for just the Les Florets transcripts and 65.16% for just the Cartoon transcripts, suggesting that the model should do just as well with transcripts using different simulations.

We also calculated the Cohen's kappa. The weighted Cohen's kappa was .56 with the no information rate of .44 (p-value of difference is smaller than .001). This kappa according to Landis and Koch (1977) is "moderate agreement", and according to Fleiss (1981)[5] is "fair to good" agreement. Rather than relying on conventional categorical guidelines to interpret the magnitude of kappa, Bakeman (2023) argues that researchers should estimate observer accuracy or how accurate simulated observers need to be to produce a given value of kappa. The KappaAcc program (Bakeman, 2022)[4] was used to estimate observer accuracy, which was found to be 79%.

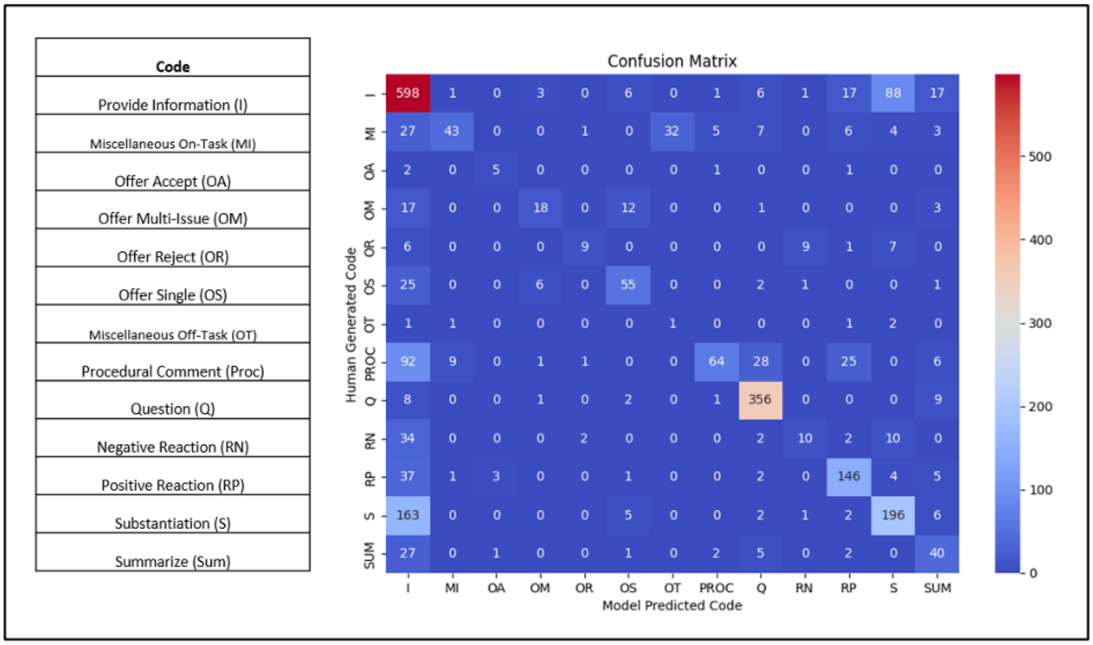
Table 3: Match Percentage by Consistency Level, Validation Step 2

Level of Consistency	**Match with Human Codes**	**% Achieve This Consistency Level Among Those Assigned a Code**	**Number**	**Match**	**Percentage Match**
Modest Consistency	3 out of 5	2%	34	not match	
			22	match	39%
High Consistency	4 out of 5	10.1%	75	not match	
			200	match	72%
Perfect Consistency	5 out of 5	87.8%	839	not match	
			1541	match	65%

*44 cases did not reach the 3 out of 5 consistency threshold or the model failed to assign a code

The proportion of speaking units that fell into each category were roughly similar to what we saw in the first validation tests, with most speaking units being: Information, Question, and Response Positive. In this set of transcripts Substantiation was also fairly common (see Table 4). As with the first validation test, model-human match percentage appears to be highly correlated with number of codes.

The confusion matrix (see Figure 2) shows that, once again, the largest number of mismatches comes from Information/Substantiation. It also shows that nearly all of the mismatches were cases where the model assigned a code of “information” when the humans assigned various other codes.



Code
Provide Information (I)
Miscellaneous On-Task (MI)
Offer Accept (OA)
Offer Multi-Issue (OM)
Offer Reject (OR)
Offer Single (OS)
Miscellaneous Off-Task (OT)
Procedural Comment (Proc)
Question (Q)
Negative Reaction (RN)
Positive Reaction (RP)
Substantiation (S)
Summarize (Sum)

	I	MI	OA	OM	OR	OS	OT	PROC	Q	RN	RP	S	SUM
I	598	1	0	3	0	6	0	1	6	1	17	88	17
MI	27	43	0	0	1	0	32	5	7	0	6	4	3
OA	2	0	5	0	0	0	0	1	0	0	1	0	0
OM	17	0	0	18	0	12	0	0	1	0	0	0	3
OR	6	0	0	0	9	0	0	0	0	9	1	7	0
OS	25	0	0	6	0	55	0	0	2	1	0	0	1
OT	1	1	0	0	0	0	1	0	0	0	1	2	0
PROC	92	9	0	1	1	0	0	64	28	0	25	0	6
Q	8	0	0	1	0	2	0	1	356	0	0	0	9
RN	34	0	0	0	2	0	0	0	2	10	2	10	0
RP	37	1	3	0	0	1	0	0	2	0	146	4	5
S	163	0	0	0	0	5	0	0	2	1	2	196	6
SUM	27	0	1	0	0	1	0	2	5	0	2	0	40

Figure 2: Confusion Matrix, Validation Step 2

Human Code	**% of units Across All Transcripts**	**Model Match %**
I	28.49	79.04%
S	15.39	52.77%
Q	14.32	92.75%
RP	13.24	83.75%
PROC	9.09	28.16%
MI	5.79	32.94%
OS	3.67	60.61%
SUM	3.26	52.27%
RN	2.52	14.70%
OM	2.30	35.48%
OR	1.26	32.35%
OA	0.41	45.45%
OT	0.26	14.28%

Table 4: Mach Percentage by Code, Validation Step 1

In order to assess the mismatches, we collected a random sample of 100 sentences with mismatches, along with the two prior sentences and the human and model codes. We then took off the column labels and randomly mixed the order of the codes. Since the human coding in this case was done by our coding team, we wanted a different person to select which of the two codes was more correct. This was done by the first author. The results are shown in Table 5. About half of the time the human code was deemed accurate, but also in about half of the cases either the model was deemed accurate, or both the model and human code selections were feasible. Sometimes this was because different parts of the sentence focused on different things, or it was unclear if (for example) if "information" provided was just an expression of the speaker's priorities, or a way to back up their demands ("substantiation"). Looking, then, at the 31% of speaking units that were mismatch, perhaps half of them might still be deemed accurate.

Code Selection		**Count**	
Clear Choice	Human Code is Correct	47	68
	Model Code is Correct	21	
Both Correct	Human Code is Correct but Both are Feasible	11	24
	Model Code is Correct but Both are Feasible	8	
	Both are Equally Correct	5	
Both Incorrect	Human and Model Both Incorrect	6	6
Not Understood	Could not Understand the Sentence	2	2

Table 5: Assessment of 100 Sample Mismatches

Confirm Validation of Model Using Claude Opus 4.5 (replaced Opus 3.0 on 1.1.26)

On 12.31.25, the version of Claude's Opus model we had used for Model 1 (Version 3.0) was decommissioned and we replaced it with the most current model, Opus 4.5. Below we report reruns of the main analyses which show that the Opus 4.5 validation results are consistent with the validation results reported above for Opus 3.0.

Step 1: **Confirm Validation.** Compare the model results (using Opus 4.5) with humans results from Aslani et al (2014). We asked the model to code the first half of all the Aslani et al (2014) transcripts that were not selected for training (2,711 speaking units). The overall match level for units where the model assigned a code was 72%, (within the 95% CI of the match results found with Opus 3.5, which was 73%). We calculated Cohen's kappa, with the model codes as one rater and the human codes as a second rater, using the KappaAcc program (Bakeman, 2022). Cohen's Omnibus Kappa was .67 (nearly identical to the .69 Kappa when using Opus 3.5), with the no information rate of 25% (p-value of the difference is <.001). Estimated observer accuracy was 84% (nearly identical to the 85% estimated when using Opus 3.5). These results show that the coding performance of Opus 4.5 is nearly identical to that of Opus 3.0.

Step 2: **Confirm Validation.** Match with human coding for different simulations (using Opus 4.5). We used the same 12 transcripts used in Step 2 analysis for Opus 3.5 reported above, which contained 2711 speaking turns. The match percentage was .68, within the 95% CI for match results using Opus 3.5. We calculated Cohen's kappa, with the model codes as one rater and the human codes as a second rater, using the KappaAcc program (Bakeman, 2022). Cohen's Omnibus Kappa was .61 (higher than the .56 Kappa when using Opus 3.5), with the no information rate of .29 (p-value of the difference is <.001). Estimated observer accuracy was 81% (nearly identical to the 79% estimated when using Opus 3.0). These results show that the coding performance of Opus 4.5 is nearly identical to that of Opus 3.0.

Formatting Your Transcripts

Set up your transcripts for analysis by putting them into an excel sheet. The format should be as shown below. Label the first column "SpeakerName" and list whatever names you have for those speakers (e.g., buyer/seller, John/Mary). Label the second column "Content" and include the material (in English or other major languages) that is contained in your unit of analysis (which may be a speaking turn, a sentence, or a thought unit). Also include columns for ResearcherName, Email, and Institution and include that information in the next row. If you use *speaking turns* then speakers will alternate, and the format will look like this:

SpeakerName	**Content**	**ResearcherName**	**Email**	**Institution**
Buyer	*All words in speaking turn...*	*Your name*	*Your email*	*Your institution*
Seller	*All words in speaking turn...*			
Buyer	*All words in speaking turn...*			
Seller	*All words in speaking turn...*			
Buyer	*All words in speaking turn...*			
etc	etc			

If you use *sentences* or *thought units* then it is possible that speakers may appear several times in a row, and the format will look like this:

SpeakerName	**Content**	**ResearcherName**	**Email**	**Institution**
Buyer	*All words in sentence or thought unit...*	*Your name*	*Your email*	*Your institution*
Buyer	*All words in sentence or thought unit...*			
Buyer	*All words in sentence or thought unit...*			
Seller	*All words in sentence or thought unit...*			
Seller	*All words in sentence or thought unit...*			
Buyer	*All words in sentence or thought unit...*			
etc	etc			

Create one Excel file for each transcript. Name each file in the following way:
YourName_StudyName_1
YourName_StudyName_2

YourName_StudyName_3
Etc.

Submit Your Transcript

To submit your transcript for the model to code, drag and drop one or several transcript files into the submission section of the website. The content can be in English or other major languages. It will likely take about 10 minutes for Claude to process each transcript, although this can vary based on how much demand Claude has at the moment you submit your files. **Do not close your window while you are waiting for results - you will lose your results.** Once the analysis for each transcript is complete, you will receive the output in a csv file that is automatically downloaded to your download folder. We suggest submitting just a few files at a time, so that you can check the output before doing too many analyses. The output files will include:

- Transcript Name
- Speaker
- The text (which could be a thought unit, sentence, or speaking turn)
- The code assigned to that text
- Consistency score for that code

If you have any questions, contact the Vanderbilt AI Negotiation Lab.

References

[1] Aslani et al. (2014), "Measuring negotiation strategy and predicting outcomes: Self-reports, behavioral codes, and linguistic codes," presented at the annual conference of the International Association for Conflict Management, Leiden, The Netherlands.

[2] Aslani, S., Ramirez-Marin, J., Brett, J., Yao, J., Semnani-Azad, Z., Zhang, Z. X., ... & Adair, W. (2016). Dignity, face, and honor cultures: A study of negotiation strategy and outcomes in three cultures. *Journal of Organizational Behavior, 37*(8), 1178-1201.

[3] Adair, W. L., & Brett, J. M. (2005). The negotiation dance: Time, culture, and behavioral sequences in negotiation. *Organization Science, 16*, 33-51.

[4] Bakeman, R. (2022). KappaAcc: A program for assessing the adequacy of kappa. *Behavior Research Methods.* `https://doi.org/10.3758/s13428-022-01836-1`

[5] Fleiss, J.L. (1981). Statistical methods for rates and proportions (2nd ed.). New York: John Wiley. ISBN 978-0-471-26370-8.

[6] Friedman, R., Brett, J., Cho, J., Zhan, X. et al. (2024). An application large language models to coding negotiation transcripts: The Vanderbilt AI negotiation lab. (forthcoming)

[7] Gamer, M., Lemon, J., Fellows, I., & Singh P. (2019) irr: Various coefficients of interrater reliability and agreement. R package version 0.84.1. `https://CRAN.R-project.org/package=irr`

[8] Garrison, D. Cleveland-Innes, M., Koole, M. & Kappelman, J. (2006). Revisiting methodological issues in transcript analysis: Negotiated coding and reliability. The Internet and Higher Education. 9. 1-8. 10.1016/j.iheduc.2005.11.001.

[9] Gunia, B. C., Brett, J. M., Nandkeolyar, A. K., & Kamdar, D. (2011). Paying a price: Culture, trust, and negotiation consequences. *Journal of Applied Psychology, 96*, 774-789.

[10] Jackel, E., Zerres, A., Hamshorn de Sanchez, C., Lehmann-Willenbrock, & N., Huffmeier, J. (2022), "NegotiAct: Introducing a comprehensive coding scheme to capture temporal interaction patterns in negotiations," *Group and Organization Management.* (See supplementary file for coding guidelines.)

[11] Landis, J.R. & Koch, G.G. (1977). The measurement of observer agreement for categorical data. *Biometrics.* 33 (1): 159–174. doi:10.2307/2529310.

[12] R Core Team (2022). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/.

[13] Weingart, L. R., Thompson, L. L., Bazerman, M. H., & Carroll, J. S. (1990). Tactical behavior and negotiation outcomes. *International Journal of Conflict Management, 1*, 7-31.

[14] Xie. S.M. & Min, S. (2022). How does in-context learning work? A framework for understanding the differences from traditional supervised learning. Stanford AI Lab Blog, Aug 1. https://ai.stanford.edu/blog/understanding-incontext/