“If…”: Satisficing algorithms for mapping conditional statements onto social domains

Alejandro López-Rousseau
Instituto de Empresa, Madrid, Spain

Timothy Ketelaar
New Mexico State University, Las Cruces, USA

People regularly use conditional statements to communicate promises and threats, advices and warnings, permissions and obligations to other people. Given that all conditionals are formally equivalent—“if P, then Q”—the question is: When confronted with a conditional statement, how do people know whether they are facing a promise, a threat, or something else? In other words, what is the cognitive algorithm for mapping a particular conditional statement onto its corresponding social domain? This paper introduces the pragmatic cues algorithm and the syntactic cue algorithm as partial answers to this question. Two experiments were carried out to test how well these simple satisficing algorithms approximate the performance of the actual cognitive algorithm people use to classify conditional statements into social domains. Conditional statements for promises, threats, advices, warnings, permissions, and obligations were collected from people, and given to both other people and the algorithms for their classification. Their corresponding performances were then compared. Results revealed that even though these algorithms utilised a minimum number of cues and drew only a restricted range of inferences from these cues, they performed well above chance in the task of classifying conditional statements as promises, threats, advices, warnings, permissions, and obligations. Moreover, these simple satisficing algorithms performed comparable to actual people given the same task.

In the beginning was a warning, and the warning was “If you eat of the tree of knowledge, you will surely die” (Genesis 2:17). Since time immemorial people have used conditional statements like this to warn others of imminent dangers.
(e.g., “If you touch the fire, you will get burned”), promise them future rewards (e.g., “If you keep my secret, I will give you a gift”), permit them exceptional undertakings (e.g., “If you are strong enough, you can ride the horse”), and so on. Given that all conditional statements are formally equivalent—“if condition $P$ obtains, then consequence $Q$ ensues”—the question remains: When confronted with a conditional statement, how do people know whether they are facing a warning, a promise, a permission, or something else? Are there cognitive algorithms that map particular conditional statements onto their corresponding social domains? This paper introduces two algorithms as partial answers to this question.

**THE PRAGMATICS OF CONDITIONAL STATEMENTS**

Understanding the social content of conditionals in particular is an interesting and relevant step towards understanding the interpretation process of language in general in terms of adaptive reasoning algorithms. The study of how meaning is attached to verbal statements has been the province of a branch of cognitive psychology known as pragmatics. According to the pragmatics approach, arriving at the appropriate meaning of an utterance—be it a warning, a promise, or anything else—requires that the individual draws appropriate inferences. As such, the task of discerning the meaning of a statement turns out to be more of a process of utterance interpretation than utterance decoding (Sperber & Wilson, 1981, 1986). Consider the following utterances:

**Woman:** I’m leaving you.
**Man:** Who is he?

Most individuals interpret these utterances in the same way, that is, as statements occurring in a conversation between romantic lovers, one of whom wishes to end the relationship, while the other suspects infidelity. Yet, as pragmatics theorists quickly point out, none of these meanings can be directly recovered by decoding these utterances (see Sperber & Wilson, 1986). That is, there are no features (e.g., words) in these two utterances that directly translate into a clear statement of the nature of the relationship between the two speakers, their intentions, or an act of infidelity. Such meanings are not decoded from the words in an utterance, instead they are inferred from a variety of pragmatic (contextual) cues including the words in an utterance (Sperber & Wilson, 1986).

According to pragmatics theorists, individuals discern the meaning of an utterance by virtue of drawing certain inferences and not others. The sophistication of this human ability to draw appropriate inferences from utterances can be clearly seen in the case of irony (or sarcasm), where the individual correctly infers the speaker’s meaning even though the literal meaning of the speaker’s
statement is the opposite of their intended meaning (e.g., “Fred is such an honest guy, he lies only twice an hour!”).

Given the fact that (1) all conditional statements have the same logical form—“if P, then Q”—and (2) the claim from pragmatics that the intended meaning of a statement is inferred (rather than directly decoded) from pragmatic cues, how then does an individual actually decide whether a particular conditional statement is, say, a threat or a promise? One intriguing possibility is that inferences about the appropriate social domain for a conditional statement may be triggered by the presence of particular cues (e.g., particular words) in the utterance. Although a simple heuristic process for categorising statements into social domains (warnings, promises, permissions, etc.) would not necessarily provide the listener with the full meaning of the utterance, it could allow the recipient to achieve a quick and dirty approximation of the meaning of the statement.

**DRAWING INFERENCES FROM CONDITIONAL STATEMENTS**

There is a long tradition in psychology of studying the inferences that people draw about conditional statements beginning with Wason’s (1966) classic research on the selection task (e.g., Schaeken, Schroyens, & Dieussaert, 2001). In this task, participants are presented with four cards that have letters on one side and numbers on the other side (e.g., A, B, 1, 2), and then asked to select only those cards that need to be turned in order to test the conditional rule “If a card has a vowel on one side, then it has an even number on the other side.” This rule is formally equivalent to a logical “if P, then Q” rule, where the four cards correspond to P, not-P, not-Q, and Q, respectively. The typical finding in this task is that most participants fail to select the necessary not-Q card (i.e., the 1-card). This failure has been interpreted as a difficulty in reasoning according to the logic of modus tollens—“if P, then Q”; “not-Q”; “therefore not-P”—and people have been thus depicted as bad logical reasoners.

Further studies have shown that when the original task is provided with social content, people do better (e.g., Griggs & Cox, 1982). For example, participants are presented with four cards representing people at a bar that have their drinks on one side and their ages on the other side (e.g., beer, cola, 16, 20), and then asked to select only those cards that need to be turned in order to test the conditional rule “If a person is drinking beer, then she must be over 18 years old.” This rule is also formally equivalent to a logical “if P, then Q” rule, but the typical finding now is that most participants do select the necessary not-Q card (i.e., the 16-card), apparently reasoning according to modus tollens. People have been thus depicted as good social reasoners.

Different theoretical explanations have been offered for this social content effect on the Wason selection task. For example, Cheng and Holyoak (1985;
Cheng, Holyoak, Nisbett, & Oliver, 1986) suggest that people do not reason according to formal logic but to pragmatic reasoning schemas such as permissions and obligations. In a permission schema, when a given precondition is not satisfied (e.g., being over 18 years old), a given action must not be taken (e.g., drinking beer). Testing whether this holds amounts to selecting the \textit{not-Q} card in any Wason selection task that maps onto a permission schema (e.g., the bar scenario).

Alternatively, Cosmides and Tooby (1992; Cosmides, 1989) suggest that people reason according to evolved Darwinian algorithms such as social contracts and threats. In a social contract, you must pay a given cost (e.g., being over 18 years old) to take a given benefit (e.g., drinking beer). Testing whether this holds amounts to detecting cheaters, and to now selecting the \textit{not-Q} card in any Wason selection task that maps onto a social contract (e.g., the bar scenario).

Moreover, Gigerenzer and Hug (1992) suggest that cheating on a social contract depends on the pragmatic perspective of their participants. For example, whereas not working paid hours is cheating from the perspective of an employer, not paying worked hours is cheating from the perspective of an employee. Thus, this would lead to employers and employees selecting different cards to test the conditional rule “If an employee works some hours, then the employer must pay those hours.” In particular, employees would select the \textit{P} and \textit{not-Q} cards, and employers would select the \textit{Q} and \textit{not-P} cards. In sum, whereas their explanations differ, all authors above agree that proper reasoning about conditionals is not done according to a general logical formalism but to specific psychological mechanisms.

However, although both the pragmatic schema and the evolved algorithm explanations account for reasoning about conditional statements, these explanations still beg a fundamental question: When confronted with a conditional statement, how do people know whether they are facing a permission and not an obligation, or a social contract and not a threat, or neither of these but something else? In other words, if discerning the appropriate meaning of a conditional statement entails employing the right schema or algorithm, what is the mechanism for mapping a particular conditional statement onto its corresponding schema or algorithm? This paper is an attempt to provide an ecologically valid answer to this question by studying conditional statements as used in natural language.

**CONDITIONALS STATEMENTS AND DOMAIN SPECIFICITY**

Linguists such as Fillenbaum (1975, 1976, 1977) have shown that everyday reasoning about conditional statements is domain specific. That is, reasoning about threats is not the same as reasoning about promises or other social
domains. For example, whereas conditional threats (e.g., “If you hurt me, I’ll beat you up”) can be paraphrased as disjunctives (e.g., “Don’t hurt me or I’ll beat you up”), conditional promises (e.g., “If you help me, I’ll take you out”) cannot be paraphrased as disjunctives (e.g., “Don’t help me or I’ll take you out”). Moreover, domain-specific reasoning about conditionals is not necessarily logical. For example, conditionals (e.g., “If you order food, you must pay for it”) invite for some inferences (e.g., “If I don’t order food, I mustn’t pay for it”) that are logically invalid (i.e., “if not-P, then not-Q”) but make perfect social sense nonetheless. Finally, domain-specific reasoning about conditional statements is not triggered by their general form—“if P, then Q”—but by their specific content and context. For example, although a conditional promise (e.g., “If you help me, I’ll take you out”) and a conditional threat (e.g., “If you hurt me, I’ll beat you up”) are formally equivalent, one is regarded as a promise and the other as a threat by virtue of their distinct consequences, namely, a benefit and a cost for the listener, respectively.

**THE PRAGMATIC CUES APPROACH**

Given the constraints of time and cognitive resources that typically confront individuals in the world, the cognitive algorithm for classifying conditional statements into social domains is assumed to be a *satisficing* algorithm: A simple serial procedure sufficing for satisfactory classifications in most cases (Gigerenzer, Todd, & ABC Research Group, 1999; Simon, 1982). Take as an example a situation in which someone tells you “If you move, I’ll kill you.” You better know with accuracy and speed that this conditional is a threat to react appropriately. But exactly how do you know that this particular statement is a threat? Certainly the content and context of the conditional statement, as conveyed by linguistic cues (e.g., the word “kill” instead of the word “kiss”) and non-linguistic cues (e.g., a mean look instead of a nice smile), provide some guidance. Although the actual cognitive algorithm that individuals employ when classifying conditional statements into social domains probably includes both kinds of cues, the first satisficing algorithm introduced here includes only linguistic cues for simplification. Moreover, although the actual cognitive algorithm probably includes syntactic, semantic, and pragmatic linguistic cues, this algorithm includes only pragmatic cues for the simple reason that social domains are essentially pragmatic. Finally, although the cognitive algorithm probably includes all social domains, this algorithm only includes six domains given their natural relevance and historical precedence in the literature (e.g., Cheng & Holyoak, 1985; Cosmides & Tooby, 1992; Fillenbaum, 1975). The domains are the following: promises (e.g., “If you help me, I’ll take you out”), threats (e.g., “If you hurt me, I’ll beat you up”), advice (e.g., “If you exercise, you’ll be fit”), warnings (e.g., “If you smoke, you’ll get sick”), permissions (e.g., “If you work now, you can rest later”), and obligations (e.g., “If you order food,
you must pay for it’’). In sum, a satisficing algorithm for classifying conditionals by pragmatic cues was analytically derived, and consequently called the **pragmatic cues** algorithm.

### The pragmatic cues algorithm

The pragmatic cues algorithm is a binary decision tree based on three pragmatic cues that sequentially prune the tree until one of six social domains is left (see Figure 1). The cues are the following:

1. Is the conditional statement’s consequent $Q$ meant as a benefit for the listener? If it is meant as a benefit for the listener, the conditional statement represents a promise, an advice, or a permission. If it does not, the conditional statement represents a threat, a warning, or an obligation.
2. Does the conditional statement’s consequent $Q$ involve an act of the speaker? If it does involve an act of the speaker, the conditional statement represents a promise or a threat, depending on the first cue value. If it does not, the conditional statement represents an advice, a permission, a warning, or an obligation, depending on the first cue value.
3. Does the conditional statement’s consequent $Q$ make an act of the listener possible or necessary? If it does make an act of the listener possible, then the conditional statement represents a permission or an obligation, depending on the first two cue values. If it does not, the conditional statement represents an advice or a warning, depending on the first two cue values.

Take again as example the conditional statement ‘‘If you move, I’l kill you.’’ The pragmatic cues algorithm would process this conditional statement by

![Figure 1. The pragmatic cues algorithm.](image-url)
applying its first cue, and asking whether the conditional statement’s consequent \( Q \) is meant as a benefit for the listener. Because being killed is not a benefit for the listener, the algorithm would follow its ‘‘no’’ branch to the second cue, and ask whether the conditional statement’s consequent \( Q \) involves an act of the speaker. Because the killing is done by the speaker, the algorithm would follow its ‘‘yes’’ branch to the threat domain, and stop there. Thus, according to this algorithm, the conditional statement ‘‘If you move, I’ll kill you’’ is a threat. Different conditional statements are mapped onto different domains as can be verified by applying the algorithm to the following three examples: ‘‘If you smoke, you’ll get sick’’, ‘‘If you work now, you can rest later’’, and ‘‘If you exercise, you’ll be fit’’ (see Figure 1).

The pragmatic cues algorithm is meant to be simple by including the minimum possible of three cues to classify six domains. The algorithm is also meant to be serial by adopting the sequential form of a decision tree, which further simplifies the classification process by discarding already three domains from consideration after the first cue, and possibly two more domains after the second cue. And the algorithm is meant to be satisficing by producing correct classifications in most but not all cases. In this regard, the pragmatic cues algorithm could misclassify any conditional statement belonging to other (social) domains and/or depending on other (pragmatic) cues. For example, the algorithm would misclassify the conditional fact ‘‘If water boils, it evaporates’’, the conditional request ‘‘If you leave the room, please close the door’’, and the conditional promise ‘‘If I get a raise, I’ll quit smoking’’ (see Figure 1). Still, the pragmatic cues algorithm would correctly classify most conditional promises, threats, advices, warnings, permissions, and obligations.

**Overview of Experiment 1**

Given that a vast number of complex and/or parallel and/or optimising alternative algorithms could be used for this categorisation task, an experiment was designed to empirically test how well the more parsimonious pragmatic cues algorithm approximates the performance of the actual cognitive algorithm that people use to classify conditionals statements into social domains. Briefly, conditional statements for promises, threats, advices, warnings, permissions, and obligations were collected from people, and given to both other people and the pragmatic cues algorithm for their classification. Their corresponding performances were then compared.

Evidently, it was expected that people would correctly classify all of other people’s conditional statements except for obvious generation and/or interpretation errors. It was also expected that the pragmatic cues algorithm would correctly classify most conditional statements. And both people and the pragmatic cues algorithm were expected to perform far above chance. However, it was also expected that the pragmatic cues algorithm would perform somewhat
worse than the actual cognitive algorithm that people use to classify conditionals statements into social domains. This is the case because the pragmatic cues algorithm was designed as a simple satisficing algorithm that draws only a restricted range of inferences from a minimum number of cues, whereas people might have access to a larger number of cues and inferences.

In sum, the pragmatic cues algorithm would have to perform both as bad as chance and far worse than people in order for it to be rejected as an approximation of the actual cognitive algorithm people use to classify conditionals statements into social domains. Here rests the power of this empirical test to discriminate between the proposed analytical-cues algorithm and any other random-cues algorithm.

**EXPERIMENT 1**

**Method**

*Participants and materials.* Sixty-two properly informed and protected students at the University of Munich volunteered for this experiment, originally in German. Typewritten booklets contained the instructions for the participants.

*Design and procedure.* This experiment had three conditions: the generation, evaluation, and algorithm conditions. In the generation condition, 50 participants separately provided each a written conditional promise, advice, permission, threat, warning, and obligation, for a total of 300 conditionals. The instructions were the following:

We are interested in how you use *if-then* statements to communicate a promise, advice, permission, threat, warning, or obligation to someone else. Please write an example of each.

**Promise**  If ____________________________________________,
then ____________________________________________.

For instance, one participant wrote the statement “If you don’t study, then you’ll fail the exam” as an example of a warning. Thus, this conditional was one of the 50 warnings provided in the generation condition.

In the evaluation condition, three judges separately classified each of the 300 randomly ordered, nonlabelled conditionals as a promise, advice, permission, threat, warning, or obligation. The instructions were the following:

This booklet contains 300 *if-then* statements. For each statement, please state if the speaker meant it as a promise (P), an advice (A), a permission (E), a threat (T), a warning (W), or an obligation (O) for the listener.

1. If you don’t study, then you’ll fail the exam. _____
Each conditional was then classified into the domain agreed upon by two out of the three judges. For example, the three judges wrote that the speaker meant the statement “If you don’t study, then you’ll fail the exam” as a warning (W) for the listener. Thus, this conditional was classified as a warning in the evaluation condition.

In the algorithm condition, nine judges separately provided the pragmatic cue values for each of the 300 randomly ordered, nonlabelled conditionals. There were three judges per cue. The instructions for the first cue were the following:

This booklet contains 300 if–then statements. For each statement, please state if its then-part is meant as a benefit for the listener (Y), or not (N).

1. If you don’t study,
   then you’ll fail the exam. _____

The instructions for the second cue were: “This booklet contains 300 if–then statements. For each statement, please state if its then-part involves an act of the speaker (Y), or not (N).” Finally, the instructions for the third cue were: “This booklet contains 300 if–then statements. For each statement, please state if its then-part makes an act of the listener possible or necessary (Y), or not (N).”

Each conditional was then assigned the cue values agreed upon by two out of three judges per cue, and classified into the domain obtained following the pragmatic cues algorithm. For example, the three judges wrote that the then-part of the statement “If you don’t study, then you’ll fail the exam” is not meant as a benefit for the listener (N), does not involve an act of the speaker (N), and does not make an act of the listener possible or necessary (N).¹ Thus, following the pragmatic cues algorithm, this conditional was classified as a warning in the algorithm condition.

Participants were tested individually in all conditions.

Results and discussion

Figure 2 shows the percentage of conditional promises, advices, permissions, threats, warnings, and obligations provided in the generation condition that were correctly classified as such in the evaluation and algorithm conditions.

Results show that people classified most conditional statements correctly across domains (average: 94%; range: 88% to 100%), and that the pragmatic cues algorithm did almost as well as people (average: 85%; range: 68% to 94%). Both the algorithm’s and people’s classifications were far better than chance

¹The last cue value was agreed upon by two out of the three judges.
(17%), and their misclassifications were randomly distributed across domains.\(^2\) These findings indicate that the pragmatic cues algorithm approximates well the performance of the cognitive algorithm for mapping conditional statements onto social domains. The small difference is probably due to the additional cues the cognitive algorithm depends on. These findings thus suggest that the parsimoniously simple, serial, and satisficing pragmatic cues algorithm might be an integral part of the cognitive algorithm for classifying conditionals.

But what other cues and domains might also be integral parts of the cognitive algorithm? Besides pragmatic cues, the cognitive algorithm probably includes semantic, syntactic, and nonlinguistic cues. Moreover, the cognitive algorithm probably includes orders, requests, and other social domains. The second satisficing algorithm introduced here explores the role of syntactic cues in the cognitive algorithm for classifying conditional statements.

THE SYNTACTIC CUES APPROACH

The inclusion of syntactic cues in the cognitive algorithm can be exemplified by means of conditional requests (e.g., “If you call, please send my regards”),

\(^2\) Except for the algorithms’ misclassifications of advices mostly as permissions (11 out of 16). For example, the advice “If you talk more, then you can solve your problems” was also misclassified as a permission. This is due to the algorithm’s third cue (i.e., Does the conditional’s consequent \(Q\) make an act of the listener possible or necessary?), where “possible” can broadly mean “plausible” or narrowly mean “permissible”. A cue’s rephrasing might better convey the intended narrow meaning (e.g., Does the conditional’s consequent \(Q\) involve an authority making an act of the listener possible or necessary?).
which usually initiate their consequents with the word “please”. This syntactic cue signals a following request. Whether a request actually follows is then determined by additional pragmatic cues. Thus, the cognitive algorithm is here assumed to include syntactic cues as early detectors of social domains, which are later (dis)confirmed by pragmatic cues.

Take conditional threats as a less obvious example. Threats are typically used to induce people in doing something they are not doing (e.g., “If you don’t pay, I’ll break your arm”). Thus, conditional threats usually include the word “not” in their antecedents. This syntactic cue detects an imminent threat, which is then (dis)confirmed by the pragmatic cues proposed above (see pragmatic cues algorithm). In sum, to explore the role of syntactic cues in the cognitive algorithm for classifying conditional statements, a satisficing algorithm for detecting threats by syntactic cues was designed, and consequently called the syntactic cue algorithm.

The syntactic cue algorithm

The syntactic cue algorithm is a binary decision tree based on just a single syntactic cue that prunes the tree into threats and nonthreats (see Figure 3). The cue is the following:

1. Does the conditional statement’s antecedent P contain the word “not”? If it does contain the word “not”, the conditional statement represents a threat. If it does not, the conditional statement represents no threat.

Take again as example the conditional statement “If you don’t pay, I’ll break your arm.” The syntactic cue algorithm would process this conditional statement by applying its only cue, and asking whether the conditional statement’s antecedent P contains the word “not”. Because it does, the algorithm would follow its “yes” branch to the threat domain, and stop there. Thus, according to the algorithm, the conditional statement “If you don’t pay, I’ll break your arm” is a threat. Evidently, the syntactic cue algorithm would not detect any conditional threat excluding “not” from its antecedent (e.g., “If you testify, I’ll cut out your

![Figure 3. The syntactic cue algorithm.](image)
tongue”), and would wrongly detect any conditional nonthreat including “not” in its antecedent (e.g., “If you don’t mind, I’ll invite you to dinner”). Still, the algorithm is expected to correctly detect most conditional threats.

Overview of Experiment 2

Given that the actual cognitive algorithm for classifying conditional statements no doubt depends on additional pragmatic cues, a second experiment was designed to empirically test how well the minimalistic syntactic cue algorithm approximates the performance of the actual cognitive algorithm people use in detecting conditional threats. Briefly, conditional statements in the form conditional threats and nonthreats were collected from people, and given to both other people and the syntactic cue algorithm for their classification. Their corresponding performances were then compared.

Again, it was expected that people would correctly classify all of other people’s conditional threats except for obvious generation and/or interpretation errors. It was also expected that the pragmatic cues algorithm would correctly classify most conditional threats. Although both people and the pragmatic cues algorithm were expected to perform far above chance, it was expected that the actual cognitive algorithm that people employ to classify conditionals statements into social domains would perform somewhat better than the syntactic cue algorithm. This is because the syntactic cue algorithm was designed as a simple satisficing algorithm that relies on just a single syntactic cue to discriminate threats from nonthreats, whereas people might have access to a larger number of cues and inferences.

Also again, the syntactic cue algorithm would have to perform both as bad as chance and far worse than people in order for it to be rejected as an approximation of the actual cognitive algorithm people use to classify conditional threats. Here rests the power of this empirical test to discriminate between the proposed analytical-cue algorithm and any other random-cue algorithm.

EXPERIMENT 2

Method

Participants and materials. Thirty-seven properly informed and protected students at the University of Munich volunteered for this experiment, originally in German. Typewritten booklets contained the instructions for the participants.

Design and procedure. This experiment had three conditions: the generation, evaluation, and algorithm conditions. In the generation condition, 33 participants separately provided each six written conditional threats (three)
and nonthreats (three), for a total of 200 conditionals.\(^3\) The instructions were the following:

We are interested in how you use *if-then* statements to communicate a threat to someone else. Please write three examples of threats and three examples of non-threats.

Threat If ________________________________,
then ________________________________.

For instance, one participant wrote the statement “If you’re not quiet, then I’ll bawl you out” as an example of a threat. Thus, this conditional statement was one of the 100 threats provided in the generation condition.

In the evaluation condition, three judges separately classified each of the 200 randomly ordered, nonlabelled conditional statements as a threat or nonthreat. The instructions were the following:

This booklet contains 200 *if-then* statements. For each statement, please write if the speaker meant it as a threat (T) or a nonthreat (N) for the listener.

1. If you’re not quiet,
then I’ll bawl you out. _____

Each conditional statement was then classified as a threat or nonthreat as agreed upon by two out of the three judges. For example, the three judges wrote that the speaker meant the statement “If you’re not quiet, then I’ll bawl you out” as a threat (T) for the listener. Thus, this conditional was classified as a threat in the evaluation condition.

In the algorithm condition, one judge provided the syntactic cue value for each of the 200 randomly ordered, nonlabelled conditional statements. The instructions for the one cue were the following:

This booklet contains 200 *if-then* statements. For each statement, please write if its *if*-part contains the word “not” (Y), or not (N).

1. If you’re not quiet,
then I’ll bawl you out. _____

Each conditional statement was then assigned the cue value given by the judge, and classified into the domain obtained following the syntactic cue algorithm. For example, the judge wrote that the *if*-part of the statement “If you’re not quiet, then I’ll bawl you out” contains the word “not” (Y). Thus, following the syntactic cue algorithm, this conditional statement was classified as a threat in the algorithm condition.

Participants were tested individually in all conditions.

\(^3\) One participant was asked to provide four threats and four nonthreats.
Results and discussion

Figure 4 shows the percentage of hits (i.e., threats classified as threats), misses (i.e., threats classified as nonthreats), and false alarms (i.e., nonthreats classified as threats) for conditional threats in the evaluation and algorithm conditions.

Results show that people correctly detected most conditional threats (88% hits and 12% misses), and that the syntactic cue algorithm did just 20% worse than people (67% hits and 33% misses). Both the algorithm’s and people’s hit rates were better than chance (50%), and their false alarm rates were low (2% and 17%, respectively). These findings indicate that the syntactic cue algorithm tends to approximate the performance of the cognitive algorithm in detecting conditional threats. The observed difference is certainly due to the additional pragmatic cues the cognitive algorithm depends on. These findings thus suggest that the minimalistic syntactic cue algorithm might be another integral part of the cognitive algorithm for mapping conditional statements onto social domains. More generally, these findings suggest that the cognitive algorithm might include syntactic cues as early detectors of social domains, which might be later (dis)confirmed by pragmatic cues.

A follow-up study was designed to further test whether the syntactic cue algorithm is an integral part of the cognitive algorithm for classifying conditional statements. In this study, 100 properly informed and protected students at the University of Munich volunteered for separately answering the following typewritten question, originally in German:4

![Bar chart](image)

**Figure 4.** The percentage of hits, misses, and false alarms for threats by condition.

---

4 Options (a) and (b) were counterbalanced.
Suppose a speaker says the following to a listener:

If you don’t do that, then . . .

What do you think the speaker means by it: (a) or (b)?

(a) A threat for the listener.
(b) No threat for the listener.

Notice that the conditional statement “If you don’t do that, then . . .” has a content- and context-free antecedent, and no consequent. Thus, only the syntactic cue algorithm could be possibly used to classify this conditional statement as a threat. In fact, results show that most participants (88%) thought that the speaker means a threat for the listener by the conditional statement. This finding indicates that the syntactic cue algorithm was possibly used to classify the conditional as a threat. This finding suggests again that the syntactic cue algorithm might be an integral part of the cognitive algorithm for mapping conditional statements onto social domains. Moreover, this finding suggests that the cognitive algorithm might depend on syntactic cues alone to detect social domains, when additional pragmatic cues are not available.

A second follow-up study was designed to control for a possible task-demand effect on the results of Experiment 2 and the first follow-up study. Specifically, these results may be affected by contrasting threats with no threats instead of specific social domains like promises. Thus, the syntactic cue algorithm may well discriminate between threats and no threats, but may not discriminate between threats and other specific social domains. To test this possibility, the 300 conditionals generated in Experiment 1 were classified following the syntactic cue algorithm (for details, see algorithm condition of Experiment 2). Figure 5 shows

![Figure 5](image-url)
the percentage of conditional threats classified as threats (hits), and conditional warnings, obligations, permissions, advices, and promises classified as threats (false alarms) by the algorithm.

Results show that the syntactic cue algorithm correctly detected most conditional threats (64%). This hit rate was far better than chance (17%), and similar to the hit rate in Experiment 2 (67%). Except for warnings (58%), the false alarm rate was uniformly low (obligations 4%, permissions 4%, advices 8%, and promises 2%, respectively). On average (15%), this false alarm rate was also similar to the false alarm rate in Experiment 2 (17%). These findings indicate that the syntactic cue algorithm does not discriminate between threats and warnings, but does indeed discriminate between threats and four specific social domains (plus nonthreats). Certainly, the syntactic cue algorithm has to be revised to include warnings (see Figure 6), which are only pragmatically discriminated from threats by their consequences (see pragmatic cues algorithm). These findings thus suggest that the results of Experiment 2 and the first follow-up study were not a task-demand effect. Again, these findings suggest that the syntactic cue algorithm might be an integral part of the cognitive algorithm for mapping conditional statements onto social domains. Generally, these findings suggest that the cognitive algorithm might include syntactic cues as early detectors of social domains, which might be later (dis)confirmed by pragmatic cues.

CONCLUSIONS

Are there cognitive algorithms that map particular conditional statements onto their corresponding social domains? Actual people no doubt have access to a vast number of cues and inferences that they can use to map conditional statements onto their social domains. The challenge in the current study was to determine whether relatively simple algorithms could perform as well as actual people. This was done by introducing two simple satisficing algorithms for classifying conditional statements into their appropriate social domains: the pragmatic cues and syntactic cue algorithms. Results revealed that even though these algorithms utilised a minimum number of cues and drew only a restricted

![Diagram](image-url)

Figure 6. The revised syntactic cue algorithm.
range of inferences from these cues, they performed well above chance in the
task of classifying conditional statements as promises, threats, advices, warn-
ings, permissions, and obligations. Moreover, these simple satisficing algorithms
performed comparable to actual people given the same task.

Gigerenzer (1995) has proposed that the psychological and linguistic
approaches to studying reasoning about conditional statements could be inte-
grated into a two-step research programme. According to Gigerenzer, the first
step would be to model the cognitive algorithm that maps conditional statements
onto social domains. The second step would be to model the cognitive module
that reasons and acts accordingly in each domain. This paper is an attempt to
specify the first step of the proposed programme by demonstrating how simple
satisficing algorithms can approximate the performance of people.

Manuscript received October 2001
Revised manuscript received January 2003
PrEview proof published online October 2003

REFERENCES
391–416.
approaches to training deductive reasoning. Cognitive Psychology, 18, 293–328.
Cosmides, & J. Tooby (Eds.), The adapted mind: Evolutionary psychology and the generation of
Fillenbaum, S. (1976). Inducements: On the phrasing and logic of conditional promises, threats, and
warnings. Psychological Research, 38, 231–250.
and Reasoning, 1, 289–400.
Gigerenzer, G., & Hug, K. (1992). Domain specific reasoning: Social contracts, cheating, and per-
York: Oxford University Press.
Schaeeken, W., Schroyens, W., & Dieuussaert, K. (2001). Conditional assertions, tense, and explicit
Harvard University Press.
UK: Penguin.