(Indirect) requests in Natural Language Processing: a preliminary theoretical proposal

Alba Luzondo Oyón  
Universidad Nacional de Educación a Distancia  
España

Ricardo Mairal Usón  
Universidad Nacional de Educación a Distancia  
España

ONOMÁZEIN 51 (March 2021): 204-226  
DOI: 10.7764/onomazein.51.12  
ISSN: 0718-5758

Received: March 2019  
Accepted: October 2019
Abstract

This paper focuses on conventionalized and non-conventionalized indirect speech acts, and more concretely, on (indirect) requests. We do so within a Natural Language Processing environment called FunGramKB, which adheres to a cognitively-oriented Construction Grammar view of language. Here, conventionalized formulations like Can you X? are treated as constructions in their own right; that is, as entrenched form-meaning pairings and, thus, they are not considered indirect. By contrast, non-conventionalized formulations such as those instantiated by negative state remarks (e.g. I'm hungry) require degrees of inferencing for interpretation. Both types are computationally handled in different modules of FunGramKB. Our aim is to show that a cognitive version of Construction Grammar can offer a solution to the computational treatment of illocution.

**Keywords:** coded constructions; indirect speech acts; inference-based representations; Natural Language Processing; requests.
1. Introduction

Since the inception of Speech Act Theory (Austin, 1962), later developed in Searle (1969, 1979), scholars from various persuasions have devoted their attention to indirect speech acts or ISAs. In the classical Searlean view of indirectness, an utterance like *Could you close the door all the way?* counts as an indirect request because it is performed via another direct speech act, in this case, a question. Thus, using speech act terminology, we find that in the sentence above there is a mismatch between the literal meaning or literal illocutionary point of the utterance (i.e. a question about the hearer's ability to perform an action) and the intended (non-literal) illocutionary point, i.e. a request on the part of the speaker to have the addressee close the door. As pointed out in Ruytenbeek (2017: 295), this distinction between the direct or literal meaning of a speech act and its indirect or intended interpretation stems from Searle's (1975) view that the former is determined by the sentence-type in which it is realized (i.e. declarative, interrogative, and imperative). Whatever the discipline or area (e.g. robotics, linguistics, dialogue agents, etc.), most of the literature devoted to speech acts has maintained the direct-indirect dichotomy and, given that communication is often indirect (Ervin-Tripp, 1976), ISAs have received considerable attention. These, in turn, range from conventionalized expressions (e.g. *Can you pass the salt?*) to largely unconventional and highly context-dependent cases like *It is hot in here*, which require inferencing on the part of the hearer to reach the speaker's actual intended interpretation (e.g. a request, a complaint, etc.) (see Pérez, 2013, for an in-depth discussion of conventionalization).

For Natural Language Understanding (NLU), be it a matter of Human-Robot Interaction (HRI) or of intelligent software agents, the frequent indirectness of human discourse poses an obvious challenge, much more so in the case of non-standardized expressions. The complexity of machine interpretation of human intention has thus been the object of extended analysis over the years (e.g. Allen and Perrault, 1980; Perrault and Allen, 1980; Hinkelman and Allen, 1989; Traum, 1999; Jurafsky and Martin, 2000; Bouzouba et al., 2005; Wilske and Kuurijff, 2006; Nolan, 2014, 2017; Briggs and Scheutz, 2016; Trott et al., 2016; Trott and Bergen, 2017; Briggs et al., among others).

In this context, this article puts forward a linguistic approach to indirect requests in a Natural Language Processing (NLP) environment with a view to offering practitioners a theoretically viable set of tools for computational implementation. Thus, although the present proposal has not been implemented yet, it adds to the line of research developed in Trott et al. (2016) and Trott and Bergen (2017), for example, by providing additional evidence that Cognitive Linguistics (Dirven and Ruiz de Mendoza, 2014) and cognitively-oriented Construction Grammars (CxGs; Hoffmann and Trousdale, 2013) offer a workable alternative for the computational treatment of illocution. To do so, we present the way in which conventionalized and non-conventionalized indirect requests can be handled in the knowledge-base for NLP systems called FunGramKB (www.fungramkb.com). In it, conventionalized ISAs (e.g. *Can
(Indirect) requests in Natural Language Processing: a preliminary theoretical proposal

you close the window?) are considered constructions in their own right, that is, entrenched form-function pairings (Goldberg, 2006). These are computationally handled in the Constructicon of FunGramKB, the computational module where syntactic patterns of different size and complexity are codified. By contrast, non-standardized ISAs (e.g. It is cold in here) require inferencing and are thus calculated on the basis of contextual factors and world knowledge. Following Williams et al. (2014: 11), the robot or intelligent agent can interpret non-conventionalized ISAs on the basis of the interlocutor’s beliefs, which we here propose to store in the Cognicon of the knowledge-base in the form of scripts of the kind put forward in Ruiz de Mendoza (2014). Finally, from a computational perspective, our proposal adds to previous work developed within computational CxGs (Steels, 2012), while providing theoretical evidence that a cognitively-oriented construction-based approach (which also lies at the basis of FunGramKB) is relevant for the investigation of language processing. At the same time, the proposal makes the Constructicon sensitive to the essential requirements for an account of constructions which actually lie at the pragmatics-syntax interface (see Slabakova, 2011). Thus, ISAs are rooted in pragmatic meaning since they are, in principle, derivations from more basic actional meaning configurations. However, with use they acquire cognitive entrenchment and social conventionalization, two aspects of constructions which, while running parallel to each other, are closely interdependent. ISAs are thus on a par with other pragmatic and discourse phenomena which can attain constructional status, like topical and focal structure, referential constructions, and the different kinds of marking within the domains of politeness and figurative language (especially hyperbole and irony), and the collection of papers in Van Valin (2008).

The structure of the rest of the article is as follows. Section 2 reviews empirical evidence concerning indirect requestive meaning, to then put forward a distinction between coded constructions (e.g. Can you X?) and inferred representations (e.g. I have a terrible headache). Section 3 presents the architecture of FunGramKB, with special focus on the two modules that are relevant for our purposes here: the Constructicon, where coded illocutionary constructions are computationally handled (cf. section 4), and the Cognicon, in which non-standardized indirect requests, and, more concretely, negative state remarks, are treated (cf. section 5). Section 6 summarizes the main points covered throughout the present paper.

2. (Indirect) requests: a view from CxG

As noted in section 1, a distinction is posited between the literal/direct meaning and the non-literal or indirect meaning of an utterance like Can you shut the door? Regarding requests, which are the focus of our attention here, a further distinction is made between conventional or standardized indirect requests, such as the example above, and non-conventional ones (e.g. It is getting kind of late). While in Can you shut the door? or declarative You can shut the door the use of modal can and the second person pronoun cue for the activation of a request
interpretation, nothing in the form of *It is getting kind of late* triggers such a meaning. In this latter case, the cue is not textual but contextual.

Given this, various approaches have been posited to account for the way in which indirect requests are comprehended. One such an approach, briefly mentioned above, is known as the **Standard Pragmatic Model** (Searle, 1975), which works under the assumption that comprehension of indirect requests involves various stages: language users first compute the literal meaning of an utterance; they then determine if the literal meaning is appropriate in context and, if inappropriate, they generate a reasonable interpretation that makes sense in that specific communicative situation. Most empirical evidence, however, shows that the Standard Pragmatic Model seems largely untenable (cf. Gibbs, 1979, 1983, 1986, 1987, 1989, 1994, 2002; Coulson and Lovett, 2010; see Holtgraves, 2002: 28-33; Ruytenbeek, 2017, for comprehensive overviews). Thus, the approach known as the **Direct Access Model** maintains that “people do not necessarily analyze the literal interpretation of an indirect speech act during their immediate comprehension” (Gibbs, 2002: 473). In other words, given the appropriate context, it seems that language users can comprehend indirect requests directly without the need to compute the literal meaning first (Gibbs, 1986: 193). Additionally, Gibbs (2002: 479-480) notes that the conventionality of an expression has a facilitatory influence in understanding what speakers imply and, in fact, speakers “find highly conventionalized uses of metaphors, idioms, indirect speech acts, etc., very easy to understand”. In a similar fashion, Clark (1979) demonstrated that conventionalization and likelihood of the direct meaning in indirect requests correlate negatively, while in a series of experiments Holtgraves (1994) concluded that conventional indirect requests are quickly recognized as requests.

In light of the results provided by empirical evidence, it could be argued that if idiomaticity or entrenchment of a surface form (e.g. *Can you pass the salt?* vs. *Are you able to pass the salt?*) can provide direct access to what is traditionally assumed to be indirect requestive meaning, then such conventional forms are not indirect at all. Following this rationale, within Cognitive Linguistics, some CxG approaches (e.g. Stefanowitsch, 2003; Ruiz de Mendoza and Baicchi, 2007; Baicchi and Ruiz de Mendoza, 2010; Ruiz de Mendoza, 2013, 2015; Ruiz de Mendoza and Galera, 2014; Baicchi, 2017) propose that traditional conventionalized ISAs have constructional status: their illocutionary force is now stably associated with and thus directly connected...
to their form. This is the case of Can you X?, Could you X?, Will you X?, Would you mind X?, If you will/can X, I need X, etc. In such conventionalized expressions, pragmatic information is part of the functional pole of these constructions (cf. Capelle, 2017, for a discussion). Thus, the view adopted here is that formal configurations like the ones above capture meaning implications that were originally obtained pragmatically. However, the frequent co-occurrence of a given inference with particular formal configurations leads to the inference becoming part of the meaning of the construction (cf. Bybee, 2013: 56). The same is true at other levels of grammatical analysis. For example, constructions like the well-known WXDY (Kay and Fillmore, 1999; e.g. What’s the child doing with a gun?) are also equipped with built-in pragmatics. WXDY conveys the meaning that the speaker is bothered or surprised by the action he is asking about. This meaning, however, is not directly derivable from the question itself: a literal question about what the child is doing, in the example above. Instead, such meaning is obtained via pragmatic implication, which is now stably associated with the morphosyntax of WXDY by virtue of its use in contexts in which it is evident that the speaker already knows the answer to the literal question. As a result, the addressee’s attention is redirected away from the content of the question towards the speaker’s attitude about the content (Ruiz de Mendoza, 2015: 261).

With this in mind, we argue, following Ruiz de Mendoza and Mairal (2008), Ruiz de Mendoza (2013) and Ruiz de Mendoza and Galera (2014), for two different, yet combined, ways of making meaning. These take the form of coded constructions and inferred representations. While coded or conventionalized constructions arise from lexicogrammar, inferred meaning implications are calculated by means of contextual factors and world knowledge. In the domain under scrutiny here, illocutionary constructions (Ruiz de Mendoza and Baicchi, 2007) are taken to be entrenched pairings of form and a given pragmatic meaning, which is now part of the semantics of the construction. Inference-based representations, by contrast, cover non-conventionalized expressions which may be constructions at other levels of analysis (e.g. at the level of argument-structure; Goldberg, 1995) but whose interpretation as requests is obtained on inferential grounds (e.g. I have a terrible headache). All in all, this proposal can roughly be equated to the two strategies used for the machine interpretation of indirect speech acts, i.e. the ‘idiomatic’ approach and the ‘inferential’ approach, which in turn correspond to the processes needed to understand conventional and non-conventional utterances (see Wilske and Kruijff, 2006).

We now turn to a description of the architecture of the knowledge-base in which the codification of inference-based and coded illocution is carried out.

---

3 We use the terms ‘codified’, ‘conventional’ and ‘entrenched’ interchangeably, since the three refer to the same phenomenon, although each emphasizes a specific perspective: grammatical, social, and cognitive, respectively.
3. The architecture of FunGramKB

FunGramKB is a user-friendly online environment for the semiautomatic construction of a multipurpose lexico-conceptual knowledge base for NLP systems (see www.fungramkb.com and the publications therein). FunGramKB has been designed to be reused in other NLP tasks, and in particular in those that focus on NLU (e.g. machine translation). In addition, the knowledge base offers a multilingual environment that currently supports various western languages (e.g. Spanish, English, Italian, etc.).

As shown in Figure 1, FunGramKB comprises three major knowledge levels. The lexical and grammatical levels are language-specific. By contrast, the conceptual level is language-independent and therefore shared by all the languages included in the knowledge base. This type of modular approach, in which lexical-semantic knowledge is connected to a target language, while ontological knowledge is language-independent, is consistent with other approaches that are geared towards the development of knowledge bases designed for NLU (see Ovchinnikova, 2012).

![FIGURE 1](source: www.fungramkb.com)
What follows is a brief description of the knowledge levels specified in Figure 1 that are relevant for our analysis:

The grammatical level, or Grammaticon (cf. section 4), is the repository of ‘constructional schemata’, that is, machine-tractable representations of linguistic constructions of a varied nature (e.g. argument-structure, illocutionary structure, etc.). It comprises several Constructicons that computationally implement the four construction-types included in the usage-based constructionist model known as the Lexical Constructional Model (LCM; Ruiz de Mendoza, 2013, 2014; Ruiz de Mendoza and Galera, 2014). Unlike the rest of CxG approaches, in the LCM (and therefore, in FunGramKB), the constructicon, i.e. the network of constructions of a given language, is distributed across several types of world-knowledge structures or Idealized Cognitive Models (ICMs; Lakoff, 1987). These give shape to the semantic pole of both constructions and inference-based representations. Thus, the LCM puts forward a fine-grained taxonomy of ICMs that includes the following parameters, which are regarded as more central than others: (i) the situational or non-situational (propositional) nature of the cognitive model; and (ii) the degree of genericity that they involve, which underlies the distinction between low-level and high-level ICMs. We elaborate on these notions in sections 4 and 5.

The conceptual level is made up of three sub-modules. The Onomasticon, which handles episodic knowledge, stores information about instances of entities and events (e.g. 9/11, Jim Morrison). The Cognicon, to which we devote our attention in section 5, contains procedural or situated knowledge in the form of scripts (Schank and Abelson, 1977; e.g. ‘going to a restaurant’). Finally, the Ontology, in which semantic knowledge is stored, is defined as a hierarchical or IS-A structured catalogue of the concepts that a person has in mind. The Ontology is divided into the three subontologies: #EVENT (for verbal predicates), #ENTITY (for nous), and #QUALITY (which covers adjectives and adverbs).

In FunGramKB, conceptual schemata in each of these modules play a key role to draw inferences along the process of text understanding. The three conceptual submodules described above employ the same expressive conceptual representation language, i.e. COREL, for the formal codification of different kinds of knowledge. The metalanguage, in turn, serves as input to a reasoning engine.

Let us briefly illustrate the grammar of this formal language. In the subontology #EVENT, the conceptual unit +NEED_00 is defined in COREL as follows:

---

4 Through COREL FunGramKB can be transduced into different computational formalisms in order to simulate human reasoning (e.g. logic, production rules, conceptual graphs, etc.). FunGramKB proponents are working on an automated cognizer with human-like defeasible reasoning abilities which will be in charge of making inferences and draw conclusions from the information stored in the different conceptual modules.
(1) Conceptual unit: +NEED_00 (subordinate of +WANT_00)
   a. Thematic Frame (TF): (x1: +HUMAN_00) Theme (x2) Referent
   b. Meaning Postulate (MP): +(e1: +WANT_00 (x1) Theme (x2) Referent (f1: +MUCH_00) Quantity (f2: (e2: +BE_01 (x2) Theme (x3: m +IMPORTANT_00) Attribute (f3: x1) Goal)) Reason)

As a subordinate concept of +WANT_00, +NEED_00 inherits part of its structure. The TF of any conceptual unit specifies the number and type of participants involved in an event, which in the case of +NEED_00 would read as "a human (x1) needs an unspecified entity (x2: referent)". In turn, MPs are set of one or more logically connected predications (i.e. ‘e1’, ‘e2’, ‘e3’, etc.) through which concepts are characterized. +NEED_00 is thus defined in COREL as 'a human (x1) wants something (x2) much because (x2) is very important to him/her'. Semantic distinctions between concepts (e.g. +WANT_00 and +NEED_00) are codified through satellites (e.g. Reason in (1b)), which are introduced by an ‘f’ followed, in this case, by another predication (‘e2’, in (1b)).

4. Coded illocution in FunGramKB: the Grammaticon

In FunGramKB, the Grammaticon is the repository of constructional schemata of different languages (cf. Periñán, 2013; Luzondo and Ruiz de Mendoza, 2015, 2017). As advanced in section 3, the linguistic basis for the Grammaticon is found in the LCM, whose architecture includes four construction-types or constructional layers, namely: argument-structure (e.g. the resultative; Goldberg, 1995), implicational structure (e.g. WXDY), illocutionary structure (e.g. Can you X?; Panther and Thornburg, 1998), and discourse structure (e.g. X Let Alone Y; Fillmore, Kay and O’Connor, 1988). Each of these construction-types, whose semantic pole rests on a different type of ICM, is respectively housed in the Level-1 (L1), L2, L3 and L4-Constructicon of FunGramKB. Since systems for NLU would require the codification and interpretation of constructions of varied size, nature and complexity, the implementation of the LCM into the architecture of FunGramKB seems suitable.

As noted in Eppe et al. (2016), who apply Embodied CxG (Bergen and Chang, 2005) in robotics, CxG involves the paring of form and meaning and allows to model higher-level concepts like transitivity, intrasitivity, etc. This, they argue, is done in an abstract way that is independent from the verbal predicate that may take part in a given construction. Think, for example, of the following caused-motion realization: People almost laughed us out of Austin (Corpus of Contemporary American English (COCA), 1999). The meaning pole of the caused-motion construction is based on what the LCM calls a high-level non-situational ICM, that is, a frame-like structure that results from a process of abstraction over shared elements of several verbal predicates such as push, kick, throw, pull, etc. In other words, the origin of the caused-motion construction, whose semantics is schematically represented as ‘X CAUSES Y TO MOVE Z’ (Goldberg, 1995), arises from abstracting common elements away from actions that can set other objects in motion thus causing them to change location. As such, the sentence above makes
use of *laugh* in a causal-transitive sense that we do not normally assign to this inherently intransitive and non-causal verb (e.g. *I’ve never laughed so hard in my life*, COCA 2015), which can only take a complement that is governed by the preposition *at* (e.g. *People laughed at him*, COCA 2015). In ‘*laugh someone out*’, the caused-motion construction contributes meaning to the overall interpretation that cannot arise from the verb by itself. This is why argument-structure constructions are grounded in high-level propositional knowledge, which can be codified independently of the particular verbs that are integrated into them. In FunGramKB, for example, configurations like the caused-motion construction reside in the L1-Constructicon and are represented in abstract terms (via attribute-value matrixes) as follows:

**FIGURE 2**
Caused-motion construction (Luzondo and Ruiz de Mendoza, 2015: 83).

Like argument-structure constructions, illocutionary constructions are coded linguistic constructions. However, unlike them, illocutionary structure consists of fixed (e.g. *Can/ Could/Will you*) and variable elements (e.g. *X = sing for me, please*), which are highly specialized to convey specific illocutionary values. Similarly, both argument-structure and illocutionary structure are based on high-level ICMs. The nature of the cognitive model is nonetheless different. Whereas the meaning side of argument-structure constructions is
based on non-situational or propositional models, illocutionary constructions are grounded in *situational* ICMs, since they are coherently related conventional series of events. Recall that in FunGramKB, illocutionary structure is formalized in the L3-Constructicon. This is done via the COREL representations in (2)-(4), through which coded illocutionary meaning is captured:

(2) Requesting-type 1:
\[+(e1: +REQUEST_01 (x1: <SPEAKER>)Theme (x2: +DO_00 (x3: <HEARER>)Theme (x4: +WANT_00 (x1)Theme (x4)Referent))Referent)Goal (f1: (e4: pos +HELP_00 (x3)Theme (x1)Referent) | (e5: pos n +HELP_00 (x3)Theme (x1)Referent))Result)\]
Formal configurations: *Can you X (please)?, Could you X (please)?, Can’t you X (please)?, I wonder if you could (please) X.*

(3) Requesting-type 2:
\[+(e1: +SAY_00 (x1: <SPEAKER>)Theme (x2: +DESIRE_01 (x1)Theme (x3: +DO_00 (x4: <HEARER>)Theme (x5)Referent))Referent)Referent (x4)Goal (f1: (e4: pos +HELP_00 (x4)Theme (x1)Referent) | (e5: pos n +HELP_00 (x4)Theme (x1)Referent))Result)\]
Formal configurations: *Would you Mind X?, Will you X (please)? Would you X (please)? Wont’ you X (please)?*

(4) Requesting-type 3:
\[+(e1: +SAY_00 (x1: <SPEAKER>)Theme (x2: +NEED_00 (x1)Theme (x3: +DO_00 (x4: <HEARER>)Theme (x5)Referent))Referent)Referent (x4)Goal (f1: (e4: pos +HELP_00 (x4)Theme (x1)Referent) | (e5: pos n +HELP_00 (x4)Theme (x1)Referent))Result)\]
Formal configurations: *I need X, I need you to X, I want X, I want you to X.*

As can be seen in (2)-(4), at this level, constructions related in meaning (e.g. *Can you give me a lift?, Could you push the blue box a bit?*), are grouped under constructional domains (i.e. Requesting types-1, 2, 3). Each of these domains displays one COREL schema codifying the semantics of related sets of specific constructions. In (2), the speaker requests something from the hearer and what s/he requests is for the hearer to do what the speaker wants (e.g. *Can you open the door?*). As a result, the hearer may help the speaker or not. (3), in turn, appeals to the hearer’s willingness to carry out the desired action (e.g. *Will you answer the phone, please?*). Finally, in (4) the speaker makes the hearer aware that s/he is in need of something that the speaker may or may not provide. This approach is consistent with the one by Trott et al. (2016), who propose a way to handle utterances like *Could you push the blue box 5 inches north?* by means of building rules in which certain grammatical forms map onto certain discourse moods. To that end, the authors introduce the construction called “illocutionary modal command” (see (5)), which includes an inverted modal yes/no question that recasts the mood of the speech act as an imperative:
(5) **Construction** Illocutionary-Modal-Command
subcase of IndirectSpeechAct
constructional constituents
  core: S-With-Modal-Inversion
  optional end: QMark
meaning constraints
  self.m.mood ← “Imperative”
  self.m.addressee ← → core.m.profiledParticipant
  self.m.speechAct ← @indirect

Through such a rule, the argument-structure value of the utterance *Could you push the blue box 5 inches north?*, which is that of a polar interrogative (i.e. a question about someone’s ability to do X), is assigned a command interpretation. The difference between the structure in (5) and the ones shown in (2)-(4) is that the latter capture the semantics of coded illocutionary patterns in a finer-grained fashion. Apart from that, both approaches propose a similar solution: to provide conventionalized expressions such as *Could/would/might you X?* with constructional status, thus having directive illocutionary force directly linked to a set of formal strings.

5. **Inferred illocution in FunGramKB: the Cognicon**

Within HRI, Trott and Bergen (2017) sketch a solution for the interpretation of non-conventionalized indirect requests (e.g. *This kitchen is a pigsty, It’s cold in here*). The proposed general rule is the following:

An utterance that is a possible indirect request has an increased likelihood of being a request if the listener, L, has a reason to believe that the speaker, S, believes that L has the ability and/or obligation to fulfill the request, and that the request is not already being fulfilled.

Trott and Bergen (2017: 130)

For example, a negative state remark (Holtgraves, 1994) like *I’m hungry*, can be understood as a request to start lunch/dinner if the addressee believes that the speaker believes that the addressee has the ability and/or obligation to aid the speaker in his/her need. According to Trott and Bergen’s (2017: 131) proposal for implementation, the robot could weigh the likelihood of different interpretations for *I’m hungry* (e.g. request, complaint, etc.) using a model of what the speaker knows.

Readers may have noted that the rule stated above is based on Gibbs’ (1987) **mutual knowledge hypothesis**. Contra Sperber and Wilson’s (1982, 1986) **relevance hypothesis**, Gibbs (1987) reviews empirical evidence supporting the assumption that hearers do use the
knowledge and beliefs they share with their interlocutors in the process of interpreting utterances. One such piece of psycholinguistic evidence is provided by Clark and Marshall (1981). These authors claim that people ordinarily rely on three kinds of co-presence heuristics as evidence for inferring mutual beliefs (Gibbs, 1987: 576): (i) linguistic co-presence (i.e. interlocutors take as common ground all of their communicative exchanges up to and including the present moment), (ii) physical co-presence (i.e. we take as common ground what we have and are currently experiencing), and (iii) community membership, which is the one that concerns us here. This latter source includes information that is universally known in a community and that can be represented by means of mental structures like Schank and Abelson’s (1977) scripts and Minsky’s (1975) frames. Given this, a cognitively plausible NLP system would need to reflect knowledge constructs shared by speakers of a given linguistic community, which is where a linguistic model based on a taxonomy of ICMs, like the LCM, stands out as particularly appealing for human-machine applications. As we have seen, the LCM postulates the existence of four basic ICM types: low-level/high-level propositional or situational ICMs. Recall that situational cognitive models are conventional series of events that are coherently related to one another, thus giving rise to complex scenarios. In its turn, high-level ICMs arise from speakers’ cognitive ability to derive common structure from more specific or low-level ICMs, thus creating more general or abstract representations (see Ruiz de Mendoza, 2013: 244-245; Ruiz de Mendoza and Galera, 2014: 60-72). For example, from low-level situational models like asking for one’s allowance, begging, stating one’s needs and/or desires, etc., we can abstract away shared conceptual structure and arrive at a high-level situational ICM: the speech act of requesting (cf. Baicchi and Ruiz de Mendoza, 2010). Thus, for the LCM, illocutionary meaning is based on high-level situational ICMs that capture socio-cultural conventions regulating everyday interaction among people. One such a convention that is part of speakers’ mutual knowledge and beliefs (cf. community comembership and universality of knowledge, in Clark and Marshall (1981: 35-38)) is the high-level situational model in (6), which underlies both coded and inferred requests:

(6) “If it is manifest to A that a particular state of affairs is not beneficial to B, and if A has the capacity to change that state of affairs, then A should do so”.

(Ruiz de Mendoza and Baicchi, 2007: 17)

Let’s see how the stipulation in (6) could be codified so that the machine could infer the intended request meaning of non-coded utterances like It’s cold in here, I’m hungry, I’m thirsty, It’s too hot in here, Isn’t it noisy in here?, This kitchen is dirty, etc. These are but possible realizations of negative state remarks, i.e. nonconventional forms through which speakers perform requests by means of asserting (or questioning) the existence of a negative state if there is some action that the addressee can perform to remedy such a negative state (Holtgraves, 1994: 1207).
Our contention is that it is part of speakers’ mutual knowledge that negative states are not beneficial to those that are affected by them and that, if a speaker makes the hearer aware of an ongoing negative situation, then the socio-cultural convention in (6) is activated. This activation is cued by our shared knowledge that people are expected to transform a negative state of affairs into a positive one if they have the capacity and willingness to do so. One way in which the machine can infer the intended request meaning arising from negative state remarks is to have high-level situational models in its knowledge store. In the architecture of FunGramKB, the natural repository for structures of this kind is the Cognicon, the conceptual module that stores procedural knowledge by means of scripts, which correspond to situational ICMs.

In the Cognicon of FunGramKB, scripts are composed of one or several predications (i.e. e₁, e₂, e₃ ... eₙ) that are arranged within a linear temporal framework, in which every predication is represented by a node in a graph (for details, see Periñán, 2012; Ruiz de Mendoza, 2013). For example, a fragment of the “eating at a restaurant” script is shown in (7) (Ruiz de Mendoza, 2014: 379):

(7) @EATING_AT_RESTAURANT_00 script
*(e₁: +ENTER_00 (x₁)Agent (x₂: +CUSTOMER_00)Theme (x₃)Location
(x₄)Origin (x₅: +RESTAURANT_00)Goal (f₁: (e₂: +BE_01 (x₂)Theme (x₆: +HUNGRY_00)Attribute))Reason)₅
*(e₃: +TAKE_01 (x₇: +WAITER_00)Agent (x₂)Theme (x₅)Location (x₈)Origin
(x₉: +TABLE_00)Goal)
*(e₄: +SIT_00 (x₂)Theme (x₁₀: +CHAIR_00)Location (e₅: +BE_02
(x₁₀)Theme (x₉)Location (f₂: m +NEAR_00)Position))
*(e₆: +TAKE_01 (x₇)Agent (x₁₁: $MENU_00 | $WINE_LIST_00)Theme
(x₅)Location (x₁₂)Origin (x₉)Goal)
...

(‘A customer generally goes into a restaurant because s/he is hungry. A waiter shows
the customer to the table and the customer sits on a chair near the table. The waiter
gives the customer the menu and wine list’)

Temporal relations between predications in FunGramKB, which follow Allen’s (1983) interval temporal model, are represented through graphs, as shown in Figure 3 for (7):

---

₅ An asterisk at the beginning of a predication indicates that such a predication is defeasible. In (e₁) the use of the defeasible operator (*) indicates that customers generally go into restaurants because they are hungry, although this may not always be the case.
FIGURE 3
Temporal-knowledge representation in FunGramKB scripts (taken from Periñán, 2013: 198).

Scripts in the knowledge-base can trigger other scripts. For example, @EATING_AT_RESTAURANT_00 can function as the host script of @PAYING_00, which works as a guest script. In addition, a single predication within a script can include script activators through which participants from different scripts such as those above are mapped onto one another. As a result, the system can infer that the waiter may be the one who charges customers for the meal, the one who uses the customer’s credit card and gives the customer a receipt, etc. This means that the Cognicon stores scripts separately but, via script activators, different scenarios are interconnected, thus forming a mesh-like graph of scripts.

Having explained the structure of the Cognicon, we can now address the way in which negative state remarks could be computationally treated, so that the machine could infer the requestive meaning of non-conventionalized indirect utterances. Following the LCM, we propose to formalize the socio-cultural convention in (6), which underlies the semantic pole of both coded and inferred-based requests, by means of the following script:

(8) @REQUESTING_00 script
  +(e1: +KNOW_00 (x1: <HEARER>)Theme (x2: (e2: +NEED_00 (x3: <SPEAKER>)Theme (x4)Referent)))Referent)
  *(e3: obl +HELP_00 (x1)Theme (x3)Referent (f1: (e4: +DO_00 (x1)Theme (x5: (e5: +NEED_00
The script in (8) translates as follows: ‘the hearer knows that the speaker is in need of something (i.e. (x4)). Generally, the speaker is then obliged to help the speaker by doing what the speaker needs (x4)’.

Similarly, negative state remarks expressing different speakers’ states of need can be captured by means of scripts. To illustrate, consider the following representations:

(9) @BEING_THIRSTY_AND/OR_HUNGRY_00
*((e1: +BE_01 (x1: +HUMAN_00)Theme (x2: +HUNGRY_00 | +THIRSTY_00)Attribute)(e2: +BE_01 (x2)Theme (x3: +BAD_00)Attribute (f1:x1)Goal))
*(e3: pos +NEED_00 (x1)Theme (x5: (e3: +INGEST_00 (x1)Agent (x4: +FOOD_00 | +BEVERAGE_00)Theme (x6)Location (x7)Origin (x8)Goal))Referent)
[Temporal relation: e1 → e3 [Before ^ Equals]]

(10) @FEELING_COLD_OR_HOT_00
*((e1: +FEEL_00 (x1)Agent (x2: +HUMAN_00)Theme (x3: +COLD_00 ^ +HOT_00)Attribute)(e2: +BE_01 (x3)Theme (x4: +BAD_00)Attribute (f1:x2)Goal))
*(e3: pos +NEED_00 (x2)Theme (x6: (e4: +FEEL_00 (x7)Agent (x2)Theme (x5: $HOT_01 ^ p +COLD_00)Attribute))Referent)
[Temporal relation: e1 → e3 [Before ^ Equals]]

(11) @FEELING_TIRED_00
*((e1: +FEEL_00 (x1)Agent (x2: +HUMAN_00)Theme (x3: +TIRED_00)Attribute)(e2: +BE_01 (x3)Theme (x4: +BAD_00)Attribute (f1:x2)Goal))
*(e3: pos +NEED_00 (x2)Theme (x5: (e4: +REST_00 (x6)Theme (x2)Location))Referent)
[Temporal relation: e1 → e3 [Before ^ Equals]]

(12) @FEELING_SICK_00
*((e1: +FEEL_00 (x1)Agent (x2: +HUMAN_00)Theme (x3: +SICK_00)Attribute)(e2: +BE_01 (x3)Theme (x4: +BAD_00)Attribute (f1:x2)Goal))
*(e3: pos +NEED_00 (x2)Theme (x5: (e4: +HELP_00 (x6)Theme (x2)Referent))Referent)
[Temporal relation: e1 → e3 [Before ^ Equals]]

It is part of our mutual knowledge that the scenarios in (9)-(12) are commonly felt as negative, i.e. being thirsty/hungry, feeling hot, cold, tired or sick, etc. As a result of these situations, the affected participant may (pos operator), need the following: to eat and/or drink something (e.g. (9)), to warm up or feel less hot (e.g. (10)), to rest somewhere (e.g. (11)), or s/he may need some kind of help (e.g. (12)). Nodes in the graph can also represent script activators, thus creating more complex scenarios. For example, the @REQUESTING_00 script can be the host of the
scripts in (9)-(12), which are called for as guest scripts. This is done by mapping participants in
the host script with participants in the guest script, as shown in (13) and (14):

\begin{align*}
(13) \text{@REQUESTING_00} \\
\tau (e6: \text{@BEING_THIRSTY_AND/OR_HUNGRY_00 (x1: x3)}) \\
\end{align*}

\begin{align*}
(14) \text{@REQUESTING_00} \\
\tau (e6: \text{@FEELING_TIRED_00 (x2: x3)}) \\
\end{align*}

The representation in (13) shows that the (x1) participant in the guest script @BEING_THIRSTY_
AND/OR_HUNGRY_00, that is, the entity that is hungry and/or thirsty, is mapped onto the
participant in the host script @REQUESTING_00, i.e. the speaker who is requesting something
because s/he is in need of something that the hearer, by social-convention, should supply.
In (14), the entity feeling tired (x2) is connected to the participant performing the request.
It is by means of internal script activators that result in the mesh-like graph of scripts that
the machine could draw inferences when faced with non-conventionalized ISAs. Besides this,
because all the scripts in (9)-(11) contain the conceptual unit +NEED_00, the reasoning engine
would connect them as different states of need:

\begin{figure}
\centering
\includegraphics[width=\textwidth]{example.png}
\caption{Example of script-based graph}
\end{figure}

6. Conclusion

This paper has provided readers with a theoretical computational treatment of illocution.
Concurring with some cognitive-linguistic approaches to CxG and in line with empirical evi-
dence, we have challenged the assumed direct-indirect dichotomy and have proposed that
conventionalized forms (e.g. Can you pass the salt?) are not indirect at all. Instead, these should be taken as entrenched form-function pairings in which illocutionary force is stably associated with and thus directly connected to a given form. These contrast with inferred representations whose interpretation as requests is obtained on inferential grounds. Both structures have been treated in separate components of the architecture of the lexico-conceptual knowledge-base for NLP systems, FunGramKB. Thus, in the Grammaticon, which is based on the LCM, we have codified the semantics of illocutionary constructions. The Cognicon, by contrast, is the module in which we have proposed to house inferred representations by means of scripts. Here, the role of ICMs, and more particularly, high-level situational knowledge, has proved valuable for the codification of socio-cultural conventions (cf. mutual knowledge), which the machine could use to derive the intended request meaning of negative state remarks.

Finally, our discussion has provided additional evidence that CxG is a fruitful avenue for NLU within Artificial Intelligence and, more particularly, for the complex issue of the linguistic codification and recognition of human intention.

7. References


JUARAFSKY, Dan, and James H. MARTIN, 2000: Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition, New Jersey: Prentice Hall.


PERIÑAN, Carlos, 2013: “Towards a model of constructional meaning for natural language understanding” in Brian NOLAN and Elke DIERDICHSEN (eds.): Linking Constructions into Functional
**Linguistics: The Role of Constructions in Grammar,** Amsterdam/Philadelphia: John Benjamins, 205-230.


