Connecting morphology and semantics: What Japanese particles tell us about the meaning of questions

Goal: To characterize the meaning of questions

Let’s start by thinking about how questions are formed in Japanese

(1) Taroo-ga hon-o kaimasita.

Taroo- SUBJ book- OBJ bought.POLITE

‘Taro bought a book.’

(2) Taroo-ga nani-o kaimasita ka?

Taroo- SUBJ what- OBJ bought.POLITE Q

‘What did Taro buy?’

Interesting: Building questions in Japanese involves two parts: ka and the question word.

Moreover: These pieces occur outside of questions.

(3) Taroo-ga nani-ka-o kaimasita.

Taroo- SUBJ what- Q- OBJ bought.POLITE

‘Taro bought something.’

Incidentally: English does something like this, but on a smaller scale (where ~ somewhere, how ~ somehow but not who ~ *somewho).

Subgoal: To formally characterize the contribution of Japanese ka and nani to the meanings of (2) and (3). (In order to achieve the main goal)

The plan:

• Discuss a means of formalizing meaning
• Introspect about the meaning of questions
• Formalize a semantics for questions like (2)
• Formalization a semantics for indefinites like nanika in (3)
• Discuss some predictions about questions with multiple question words
• Consider extensions to languages other than Japanese.

I. Formalizing meaning (a fairly standard approach)

Before characterize the meaning of questions, let’s start with characterizing meanings in general.

A sentence like (4) can be true (if Quinn left) or false (if he didn’t).

(4) Quinn left.

It depends on the state of the world—in some “possible worlds” it is true, in others false.

Suppose: To know the meaning of a sentence is to know its truth conditions. (i.e. to know what the world has to look like for it to be true)

We can formalize the meaning of (4) as a function from states of the world to true/false.

(5) true, if \( w \) is a world where Q left.

false, if \( w \) is a world where Q didn’t leave.

Compositionality. The meaning of the whole is a function of • the meanings of its parts • the way it is put together.

So, Quinn and left each contribute to the determination of the truth conditions of (4).

Suppose \( \mathcal{Q} \) (and in general any name) picks out a person in the world.

(6) \( \mathcal{Q} \)

What about \( \mathcal{left} \)? Well, it is a function—given a person (the alleged “leaver”), you get truth conditions (true in worlds where the alleged leaver in fact left).

(7) \( x \leftarrow \mathcal{left} \leftarrow \mathcal{w} \)

Let’s write (7) in a more compact way.

(8) \( \leftarrow \mathcal{left} \leftarrow = \lambda x . \mathcal{w} . x \leftarrow \mathcal{w} \).

A \( \lambda x . A \) is a function that takes argument \( x \) and returns \( A \).

Here’s why we’re doing this: Recall that Japanese questions and indefinites are built from ka and from questions words. Assuming compositionality, each contributes something to the meaning. Our goal is to come up with something like (8) for Japanese ka and nani.
II. Introspecting about the meaning of questions (mainly from Hamblin 1958)

We just saw that statements—intuitively—can be either true or false (depending on the world state). But not questions.

(9) Who broke the toaster?

They do communicate something, though. For example, we know from (9) that (10) is not a possible answer, but (11) is fine.

(10) It always rains on the Fourth of July. Not a possible answer
(11) Homer broke the toaster. A possible answer

Hypothesis: Questions tell us which propositions are possible answers.

“Given a proposition \( p \), a question \( Q \) determines if \( p \) is or is not a possible answer.”

Sounds like a function.

Sounds like a function which takes propositions as arguments and returns true or false.

(12) \( S = \{ a_1, a_2, a_3, \ldots, a_n, \ldots \} \).

You can characterize this set with a “member of” function: (a characteristic function)

\[ \lambda a \cdot a \in S. \]

For any \( a \), (13) returns true if \( a \) is in \( S \), false otherwise.

The information in (13) is equivalent to the information in (12).

\[ \equiv \] A function that takes an argument \( a \) and returns true or false can also be thought of as a set of the \( a \)'s for which the function returns true.

Brief excursus into set theory: Another way to look at functions that return true or false.

(14) \( \lambda p . \exists x \in \text{humans} . p = \lambda w . x \text{ broke the toaster in } w \)

‘A proposition \( p \) is in the set if and only if (”\( \lambda p . \)”) there is an \( x \) in the set of humans (”\( \exists x \)”) such that \( p \) is \( x \text{ broke the toaster}. ‘

\( (p = \lambda w . x \text{ broke the toaster in } w) \)

Ok. We know (I) the meaning of a declarative can be characterized by its truth conditions, and (II) the meaning of a question can be characterized by its answerhood conditions. Now—Let’s look more closely at \( ka \) and question words in Japanese.

III. Questions without \( ka \), and “flexible functional application”

We’ll start with question words, because it turns out they can be isolated. That is, you can ask questions in Japanese without \( ka \) morpheme:

(15) \( \text{dare}-\text{ga kimasita } \text{ka?} \)

who-SUBJ came.POLITE \( Q \)

‘Who came?’

(16) \( \text{dare}-\text{ga kimasita } ? \)

who-SUBJ came.POLITE

‘Who came?’

Note: There are certain restrictions on this “\( ka \)-drop” (Yoshida & Yoshida 1997)

It can never happen in embedded questions.

It can only happen in yes/no questions with certain verbs.

Point: It’s systematic enough to suspect it isn’t just “phonology”

That is, it is not unreasonable to think that \( ka \) is in fact missing in (16)

(17) \( \lambda p \exists x \in \text{people} . p = \lambda w . x \text{ came in } w \)

i.e. the set of propositions of the form \( x \text{ came} \) for \( x \) a person.

As a first step, consider the minimally different sentence (18) (just like Quinn left from before)

(18) \( \text{Taroo}-\text{ga kimasita} \)

Tar-who-SUBJ came.POLITE

‘Taroo came.’

(19) \( a. \text{[kimasita] } = \lambda w . x \text{ came in } w. \)

just like \( \text{left} \) from before given \( \text{Quinn} \), yields Quinn left.

b. \( \text{[kimasita]} (\text{[Taroo]} ) = \lambda w . \text{Taroo came in } w. \)

Now, suppose we replace Taroo with dare in (19b)—it should end up meaning (17). What could we assign as the meaning of \( dare \) to get this result?
An idea due to Hamblin (1973):

Suppose what (or nani) is not a single thing (like, say, Taro), but a set.

\\[
\text{\textbf{[\textit{Taroo]}} points to a person in the world.}
\text{\textbf{[\textit{dare]}} is a set of things which point to people in the world}
\text{for example, \{Taro, Akira, Hanako, Shigeru, Kazuko, … \}}
\]

(16) \textbf{dare-ga kimasita ?}
\textit{who-SUBJ came.POLITE}
\textit{‘Who came?’}

(19) \textbf{b’}. \text{[\textit{kimasita \small (\textit{\textit{dare})} =}}
\text{\{\textit{kimasita \small (\{\textit{Taro, Akira, Hanako, Shigeru, Kazuko, … \}) = \?}}

This creates a type mismatch:
\text{\textbf{[\textit{kimasita] \small is a function from people to truth conditions (just like English \small [left] was)}
\text{It needs a person-type argument,}
\text{but in (16), it \small gets a set of person-type things.}

So what do we do? Consider a “real life” analogy:

Suppose we think of a vending machine is a function—
a function that given a quarter returns a gumball.

What if you arrive with a bag of quarters?
The machine does not take bags… it takes quarters.

This suggests a natural way of looking at this: (Know LISP? It’s a lot like mapcar)

• You have a function that can apply to individuals
• You have a set of individuals

so

• Apply the function to each individual in the set (separately).
• When you are done, you have a set of results (instead of just one).

Let’s call this flexible functional application. (see also Rooth 1985, Rullmann & Beck 1997)

In our specific example, \textbf{[\textit{kimasita] \small is a function from people to propositions.}

Recall that, Given \textbf{[\textit{Taroo] \small it yields the proposition Taro came.}
So,
Given \textbf{[\textit{dare] \small it yields a set of propositions, one for each member of \textbf{[\textit{dare] \small.}

Specifically, \textit{the propositions like x came, for each x in the set \textbf{[\textit{dare].}

That sounds familiar. That’s the meaning of the question \textit{who came?}—the propositions which are possible answers.
That is, it’s what we were after as a representation for (16).

**Conclusion:**
If we suppose that, unlike \textbf{[\textit{Taroo]}, \textit{[\textit{dare] \small is a set, a reasonable approach to type mismatch resolution automatically yields the desired representation.**
Idea: On the surface, (20) looks like (25)… BUT, supposing ka is an $\exists$-type quantifier, it is interpreted in two places, one part outside the verb as in (26).

(20) dare-ka-ga kimasita.
who-Q-SUBJ came.POLITE
‘Someone came.’

(25) qp kimasita dare ka
Surface

(26) qp ka qp kimasita dare t ka
Interpretation

Now: $[kimasita]$ is a function (like $[left]$) that needs a person-type argument. And: We hypothesized before that $[dare]$ is a set of person-type things. So: “t o” gets $[dare]$, a set, as an argument, gives $[kimasita]$ its argument, a person-type thing. That is, “t o” is a function from sets of individuals to individuals

Idea: $[t o]$ is a choice function. It takes a set as its argument and returns a member of that set.

The idea: A choice function applied to the set of integers will yield an integer—different choice functions yield different integers… So, in (26), $[t o]$ takes $[dare]$ as its argument, returns a member (which will necessarily be a person, since $[dare]$ is a set of people).

Now here’s how (26) works, all together:

(26) (repeated)

Surface

Interpretation

• $t o$ is a choice function $f$—it takes the set of people ($[dare]$) as its argument, returns a person.
• $[kimasita]$ is a function from people to propositions, so it takes the person $t o$ provides (say, $x$) and returns a proposition that $x$ left.
• ka is “$\exists f$”—it says there is some $f$ such that $x$ left, where $x = f([dare])$.

(27) $\exists f, f(people)$ came.
‘There is a way of choosing someone from the set of people such that the chosen person came.’

That’s exactly what we want (for ‘someone came’):
• True in any world $w$ in which there is a person who came.
(since we can always find a way to pick that person from the set $[dare]$).

V. Back to questions —tying up a loose end

One thing we didn’t discuss: ka has a meaning (as just proposed—roughly “$\exists f$”, to be specific). So, how come you can ask questions either with or without ka?

Answer (has got to be): In this context, ka turns out to have no effect on the meaning.

(28) dare-ga kimasita ka?
who-SUBJ came.POLITE Q
‘Who came?’

(29) dare-t o-ga kimasita ka
(syntactic proposal from Hagstrom 1998)

(C is not pronounced, but drives movement of ka)

The syntactic movement has this semantic effect:

(31) $\lambda p \exists f, p = \lambda w, f(people)$ came in w.

‘A proposition $p$ is in the set if and only if there is a choice function $f$ such that $p$ is the proposition $x$ came where $x$ is the person chosen by $f$ from the set of people.’

Characterizes a set of propositions of the form $x$ came where $x$ is a person that can be chosen from the set of people.
But recall—That’s the same set that was characterized by (17) from before (for the variant without $ka$)

(17) $\lambda p \exists x \in \text{people. } p = \lambda w . x \text{ came in } w$

Repeated from before

Note: The formulas are slightly different but the sets are exactly the same.

(17) says $x$ is drawn from the set of people (by unspecified means)

(31) says $f$ draws $x$ from the set of people.

Any proposition that satisfies (17) also satisfies (31) and vice-versa.

The point: Two mechanisms (flexible functional application vs. movement of $ka$ to C) converge on the same abstract representation (same set of propositions).

And that is why the questions with and without $ka$ mean the same thing.

Ok, we’ve done what we set out to do:

• We have achieved the subgoal, to characterize the meanings of question words and $ka$.
  1) $\text{dare \ prime yelling(x) } = \lambda x. \text{person(x)}$ [the set of people]
  2) $\text{ka \ prime \ exists}\}$ [existential quantification over choice functions]
• Along the way, we got an idea of how the meaning of questions should be characterized
  (as a set of propositions, those which are its possible answers)
• And we say how question words and $ka$ participate in these meanings
  (as well as in the meaning of $dareka$ ‘someone’-type indefinites).

VI. In support of the apparent “redundancy”, $ka$ in multiple questions

So, we saw that presence vs. absence of $ka$ didn’t seem to affect the meaning of (32–33).

(32) $\text{dare\prime-}_\text{ga kimasita ka?}$
  $\text{who}\_prime-\text{SUBJ came. POLITE Q}$
  ‘Who came?’

(33) $\text{dare\prime-}_\text{ga kimasita ?}$
  $\text{who}\_prime-\text{SUBJ came. POLITE}$
  ‘Who came?’

And we just saw why—

two different mechanisms converge on the same set of possible answers.

Do we really need two different mechanisms?

In fact, yes.

It turns out that $ka$ does have an observable effect in multiple questions.

(34) $\text{dare\prime-}_\text{ga nani\-}_\text{O kaimasita ka?}$
  $\text{who}\_prime-\text{SUBJ what}\-\text{OBJ bought. POLITE Q}$
  ‘Who bought what?’

(35) $\text{dare\prime-}_\text{ga nani\-}_\text{O kaimasita ?}$
  $\text{who}\_prime-\text{SUBJ what}\-\text{OBJ bought. POLITE}$
  ‘Who bought what?’

(34), with $ka$, can be answered with a list of pairs.

(35), without $ka$, can only be answered with a single pair.

First: What is a “list of pairs” question?

Proposal: A “list of pairs” question is actually a set of questions… and you answer each one.

Who bought what?
  when answered like
  $\text{John bought coffee, Mary bought pizza, Howard bought carrots, …}$
  is interpreted as a set of questions
  \{ $\text{What did John buy?, What did Mary buy?, What did Howard buy?, …}$ \}
  each of which you answer.

The idea behind the (34) vs. (35) contrast:

In the proposed system, we have two ways of getting sets.

One is flexible functional application, which deals with sets of arguments.

The other is moving $ka$ outside the proposition (to C).

In (35) there is no $ka$, so we have only one means available (flexible functional application)—
  We can only get a set of propositions (a single pair question).

In (34), we have $ka$ so we can use both means of set construction—
  One gets a set of propositions (a question),
  the other gets a set of questions (a pair list question).

(How it works exactly requires more background than we covered in the talk—see Appendix)
Japanese is not alone—lots of languages form ‘someone’ from ‘who’ and a particle.

- **German**: wer ‘who’ irgendwer ‘someone’
- **Icelandic**: hver ‘who’ ein-hver ‘someone’
- **Latin**: quis ‘who’ ali-quis ‘someone’
- **Rumanian**: cine ‘who’ cine-va ‘someone’
- **Mod. Greek**: pios ‘who’ ka-pios ‘someone’
- **Bulgarian**: koj ‘who’ nja-koj ‘someone’
- **Serbo-Croatian**: ko ‘who’ ne-ko ‘someone’
- **Polish**: kto ‘who’ kto ‘someone’
- **Bulgarian**: koj ‘who’ nja-koj ‘someone’
- **Korean**: nwukwu ‘who’ nwukwu-nka ‘someone’
- **Basque**: nor ‘who’ nor-bait ‘someone’
- **Sinhala**: kau ‘who’ d ‘someone’
- **English**: where ‘someone’

_Bigger ideas:_ What we learned about questions in Japanese is true more generally—even in languages which are less transparent (wrt ‘someone’ ~ ‘who’). Question formation involves question particles which move syntactically, and whose meaning is roughly “∃” for a choice function.

References mentioned


**APPENDIX: How the pair-list readings and single-pair readings relate to ka.**

(35) **dare**-**ga** **nani**-**o** kaimasita ?

_Who bought what?_

Repeated from before

There is no _ka_, so we need _flexible functional application_ (stated below—specifically, we need (iv))

(36) **FLEXIBLE FUNCTIONAL APPLICATION** (taken from Rullmann & Beck 1997)

\[
\begin{align*}
&[f_a] = \\
&\text{(where } f \text{ and } a \text{ are sisters)} \\
&(i) f(a) \\
&(ii) \lambda m \exists x. [m = f(x) \land a(x)] \\
&(iii) \lambda m \exists g. [m = g(a) \land f(g)] \\
&(iv) \lambda m \exists g. [m = g(x) \land f(g) \land a(x)] \\
&\text{whichever is defined}
\end{align*}
\]

Results in a set of propositions like x bought y for every value of x in [dare] and every value of y in [nani].

Picking an answer simultaneously picks a value of x and y—it’s a single pair answer.

(34) **dare**-**ga** **nani**-**o** kaimasita **ka**?

_Who bought what?_

Repeated from before

In (34), the structure is different—_ka_ has moved from next to one of the _wh_ -words.

(39) \[
\begin{align*}
&\llangle s, t \rrangle, \llangle t \rrangle \\
&\begin{align*}
&\langle e, t \rangle \\
&\langle e, t \rangle \\
&\langle e, t \rangle
\end{align*}
\end{align*}
\]

Artificially suspending FFA, we have:

\[
\begin{align*}
&\lambda p \exists x. p = \lambda w. [\text{dare}] \text{ bought } f(\text{things}) \text{ in } w. \\
&\text{In the spirit of FFA, we want to end up with one of these for each member of [dare] This requires allowing } \lambda \text{-abstraction to also be "flexible" (see Hagstrom 1998).}
\end{align*}
\]

\[
\begin{align*}
&\llangle e, t \rrangle, \llangle e, t \rrangle, \llangle e, t \rrangle \\
&\begin{align*}
&\langle e, t \rangle \\
&\langle e, t \rangle \\
&\langle e, t \rangle
\end{align*}
\end{align*}
\]

In the end we get something characterizable as:

(42) \[
\begin{align*}
&\lambda Q \exists x. p = \lambda w. x \text{ bought } f(\text{things}) \text{ in } w.
\end{align*}
\]