Contents

1 Overview

Overall, 39 valid sets of responses were collected out of 48.

In the first run there were 6 out of 9 participants who just provided English answers.

In the second and third run I made the following changes:

- Exclude US-based turks.
- Specify in the title and the first page that the task involves translation.
- Tell the participant to stop/skip the HIT if she only speaks English.

In the second run, 9/9 participants provided valid answers.

In the third run (30 HITs), two participants still filled in "English" as their native language for whatever reason. Another participant claimed "Hindi" to be the native language but still answered in English regardless.

The breakdown of languages in all the responses is as follows:

- 7 Hindi
- 11 Spanish
- 4 Portuguese

- 1 Greek
- 1 German
- 1 Finnish
- 1 Bengali
- 1 Arabic
- 1 Tamil
- 2 Yoruba
- 1 Tagalog
- 1 Swedish
- 2 Macedonian
- 1 Russian
- 1 Romanian
- 1 Urdu

2 Issues about the sentence frame

In general, the "sentence frame" is usually fixed, similar to "... of the dots are ..." in English, for example:

- Spanish: "... de los puntos son ..."
- Portuguese: "dos pontos são ..."

However, there are still some discrepancies regarding the sentence frame:

2.1 The presence/absence of the preposition

The preposition "of", which was incorporated as a part of the sentence frame in English, would cause ungrammaticality for some sentences in those languages if fixed into the sentence frame.

- For example, for the sentence "all of the points are white", the grammatical formulation in Spanish would be "todos los puntos son blancos" ("all the points are white"). "todos de* los puntos son blancos" would not be grammatical.

- The same problem exists for Portuguese: "todos **os** pontos são brancos." is grammatical, instead of "todos dos* pontos são brancos.".
- Some participants provided alternative formulations which incorporate "de", for example "el total de los puntos son negros". However that would be a less natural expression.
- For German a contrast would be "Ein paar **der** Punkte sind weiß" vs. "Alle Punkte sind weiß".

2.2 The difference in the plurality of the word "is/are"

In Spanish when there is "none" of something, the singular form of the verb "ser" is used. For example, "None of the dots are white" would be "Ninguno de los puntos **es** blanco" instead of "Ninguno de los puntos son* blanco". This is also a part of the sentence frame in the English expression.

2.3 Inflection of words

Sometimes inflection applies to some of the words. An example in Romanian:

- @cateva@ dintre puncte sunt albe.
- @toate@ punctele sunt albe.

Apparently the word "dots" is inflected differently in the two scenarios. Also the previously mentioned problem with the preposition exists with "dintre".

2.4 Different word orders

Sometimes multiple word orders seem to be available. For example, a Tamil-speaking respondent provided multiple solutions in a sentence:

- ''@sila@ vellai pulligal ullana', 'vellai pulligal @sila@ ullana'
- '@athigam@ pulligal karuppu niram kondavai', 'karuppu niram konda pulligale @athigam@', 'karuppu pulligaley @jaasthi@'

2.5 Different transcription systems

The two Macedonian respondents used Latin script and Cyril script respectively for their answers.

2.6 Considerably different sentence structures

One Yoruba respondent seems to have offered considerably different sentence structures for almost each of the response. Input from Yoruba speakers might be needed to explain it.

- 1 @kekere@ pele ni o je funfun
- 2 @gbogbo@ pele je dudu
- 3 laarin awon pele @diye ni oku ti gbogbo e@ ma je dudu
- 4 laarin gbogbo pele @abo@ je funfun
- 5 @ni awon pele ti owa abo je@ funfun
- 6 @won po@ laarin pele ti o je funfun
- 7 @diye ni o ku@ ti gbogbo pele fi je dudu
- 8 @kosi@ pele kan ti o je funfun.
- 9 @Awon@ pele kan je dudu
- 10 @meji@ laarin pele ni o je funfun

The second respondent was more regular but still there was some variability. Maybe she intentionally tried approximate the original English structure more closely.

- 11 @die@ lara awon aami naa ni won je dudu.
- 12 @gbogbo@ awon aami naa ni won je dudu.
- 13 @o fere je gbogbo@ awon aami naa ni won je funfun.
- 14 @idaji@ ninu awon aami naa ni won je dudu., @ilaji@ ninu awon aami naa ni won je dudu., @idaji@ ninu awon aami wonyi ni won je dudu.
- 15 @ko to idaji@ ninu awon aami naa ti won je dudu.

- 16 @opolopo@ awon aami naa ni won je dudu.
- 17 @o fere je@ gbogbo awon aami naa ni won je dudu
- 18 @okankan@ ninu awon aami naa ko je funfun., @ko si@ eyiti o je funfun ninu won aami naa.
- 19 @awon kan@ lara awon aami naa je dudu.
- 20 @meji@ ninu awon aami naa ni won je dudu., @meji@ ninu awon aami wonyi ni won je dudu.

2.7 Summary

In general, it seems that it might be hard to pin down a single sentence frame with only the quantifiers left out for the participants to fill in, since not only the quantifiers themselves, but also the other words and the sentence structure might change depending on the situation.

3 UTF-8 Issue

There seems to be something weird going on with the way the experiment interacts with MTurk. For non-Latin-script languages, the .csv file obtained from MTurk directly contains the raw bytes for the UTF-8 characters instead of the actual characters, which makes interpreting them in R extremely difficult. I don't understand why R doesn't recognize them as valid UTF-8 characters and display/print them as such. Eventually I had to employ a tedious workaround which is not sustainable. I will have to seek to change the way the experiment data is sent to uploaded in order to more easily interact with UTF-8 responses in the future.

An example: `[u@\\u03b4\\u03cd\\u03bf@ \\u03b1\\u03c0\\u03cc \\u03c4\\u03b9\\u03c2 \\u03c4\\u03b5\\u03bb\\u03b5\\u03af\\u03b5\\u03c2 \\u03b5\\u03af\\u03bd\\u03b1\\u03b9 \\u03bb\\u03b5\\u03cd\\u03ba\\u03b5\\u03c2.]` (Actually there is only one backslash at each byte, but R display automatically escapes it with another slash...)

The current workaround:

1. Write the raw bytes to an external file with `writeLines`, in order to get rid of the double backslash used for character escape in R.

2. Copy the contents in the external file manually into R, surrounded by double quotes (Yeah, I had to **manually** copy the contents, since just directly reading them back with `readLines` would just show the original raw bytes, while manually enclosing them in double quotes works. I have no idea why.
3. Store the correctly rendered UTF-8 characters into some variable/external file.

The same issue also occurs with langauges with characters outside of ASCII, e.g. the German ßis displayed as //xdf and had to be manually substituted.

4 Responses

The responses are listed in a separate file (I couldn't get LaTeX to compile if I include them in the document. TeX keeps complaining `Package inputenc Error: Unicode char (U+627) (inputenc) not set up for use with LaTeX.`). Each 10 responses constitutes a group (from the same participant). They are sorted according to the following (alphabetical) order of the quantifier (Sorry, I should probably sort it according to a natural order the next time. I didn't realize it until it was done.):

- a few
- all
- almost all
- half
- less than half
- many
- most
- none
- some
- two