Utterance-final or in Australian English

Martin Schweinberger and Michael Haugh

Feb. 28, 2023

Preparation

In a first step, we load or activate the packages.

```
library(dplyr)
library(stringr)
library(tidyr)
library(quanteda)
library(here)
library(openxlsx)
library(knitr)
```

Step 1: Computational extraction of all potential instances of utterance-final *or*

This section extracts potential instances or candidate examples of utterance-final *or* (UF-or) from four spoken corpora:

- Australian Radio Talkback
- Griffith Corpus of Spoken Australian English
- Monash corpus
- The La Trobe Corpus of Spoken Australian English

Loading data

The corpora were downloaded and stored in a directory (folder) called `data`. To load the data, we define the paths to the files containing the transcripts (which are located in the `data` folder in the specific sub-directories for the corpora).

> **NOTE**: DO NOT EXECUTE THE CODE CHUNKS BELOW! IT IS DISPLAYED FOR TRANSPARENCY REASONS ONLY! THE DATA IS NOT MADE AVAILABLE FOR COPYRIGHT REASONS!
>
> **THE INTERACTIVE CODE (CODE THAT IS EXECUTABLE) STARTS WITH THE SECTION "INTERACTIVE CODE BELOW"**

```
fart <- list.files(here::here("data", "Australian Radio Talkback/files/Raw"), 
full.names = T)
fgri <- list.files(here::here("data", "Griffith Corpus of Spoken Australian
```

```
English/files/Raw"),
                          pattern = ".txt", full.names = T)
fmon <- list.files(here::here("data", "Monash/files/Text"),
                          pattern = ".txt", full.names = T)
flat <- list.files(here::here("data", "The La Trobe Corpus of Spoken
Australian English/files/Raw"),
                          pattern = ".txt", full.names = T)
```

We now check if we have the paths to the data by inspecting the first six paths of files in the *Australian Radio Talkback* corpus.

```
# inspect
head(fart)
```

```
## [1] "D:/Uni/UQ/BookProjects/SI_ReplicationInCorpusLinguistics/ourpaper/ijcl_ufor/data/Australian Radio Talkbac
k/files/Raw/ABCE1-raw.txt"
## [2] "D:/Uni/UQ/BookProjects/SI_ReplicationInCorpusLinguistics/ourpaper/ijcl_ufor/data/Australian Radio Talkbac
k/files/Raw/ABCE2-raw.txt"
## [3] "D:/Uni/UQ/BookProjects/SI_ReplicationInCorpusLinguistics/ourpaper/ijcl_ufor/data/Australian Radio Talkbac
k/files/Raw/ABCE3-raw.txt"
## [4] "D:/Uni/UQ/BookProjects/SI_ReplicationInCorpusLinguistics/ourpaper/ijcl_ufor/data/Australian Radio Talkbac
k/files/Raw/ABCE4-raw.txt"
## [5] "D:/Uni/UQ/BookProjects/SI_ReplicationInCorpusLinguistics/ourpaper/ijcl_ufor/data/Australian Radio Talkbac
k/files/Raw/ABCNE1-raw.txt"
## [6] "D:/Uni/UQ/BookProjects/SI_ReplicationInCorpusLinguistics/ourpaper/ijcl_ufor/data/Australian Radio Talkbac
k/files/Raw/ABCNE2-raw.txt"
```

We now proceed by loading and processing (cleaning) the data.

Load ART

We start with the content of the *Australian Radio Talkback* corpus (art).

```
# load raw content
vart <- sapply(fart, function(x){
  # read in content of the file
  x <- readLines(x)
  # remove empty rows
  x <- x[x != ""]
  })
# unlist the object containing the corpus data
arttext <- unlist(vart)
# collapse into a data frame
artdf <- data.frame(names(arttext), names(arttext),arttext) %>%
  # rename columns
  dplyr::rename(corpus = colnames(.)[1],
                file = colnames(.)[2],
                text = colnames(.)[3]) %>%
  # create new columns containing corpus, file, and speaker information as
well as a column with clean content
  dplyr::mutate(
    # extract corpus name
    corpus = stringr::str_replace_all(corpus, ".*data/(.*?)/.*", "\\1"),
    # extract file name
    file = stringr::str_replace_all(file, ".*Raw/(.*?)-raw.*", "\\1"),
    # extract speaker
```

```
    speaker = stringr::str_replace_all(text, "\\[(.*?)\\].*", "\\1"),
    # clean transcripts
    textclean = stringr::str_remove_all(text, ".*?\\]"),
    # remove superfluous white spaces
    textclean = stringr::str_squish(textclean))
# remove row names
rownames(artdf) <- NULL
# inspect
knitr::kable(head(artdf))
```

corpus	file	text	speaker	textclean
Australian Radio Talkback	ABCE1	[Presenter 1: Simon Marnie, M] Thanks for that John Hall now John Hall will be listening for the next hour 'cos Angus Stewart is here to take your calls eight-triple-three-one-thousand one-eight-hundred-eight-hundred-seven-oh-two something in the garden that's causing you problems give us a call right now and Angus can I mean y'know he is known in the trade as Mr popergation {propagation} Mr propagation. He's also known for his passion for natives and his love of o orchids am I right so far.	Presenter 1: Simon Marnie, M	Thanks for that John Hall now John Hall will be listening for the next hour 'cos Angus Stewart is here to take your calls eight-triple-three-one-thousand one-eight-hundred-eight-hundred-seven-oh-two something in the garden that's causing you problems give us a call right now and Angus can I mean y'know he is known in the trade as Mr popergation {propagation} Mr propagation. He's also known for his passion for natives and his love of o orchids am I right so far.
Australian Radio Talkback	ABCE1	[Expert 1: Angus Stewart, M] I guess yeah yeah .	Expert 1: Angus Stewart, M	I guess yeah yeah .
Australian Radio Talkback	ABCE1	[P1] He's also known for his ability to open cosposting {composting} toilets so he can tell you anything worm farm problems certainly helped us and although I'm still confused about dry ingredients we might talk about that as well but eight-triple-three-one-thousand one-eight-hundred-eight-hundred-seven-oh-two fine sunny day today top temperatures on the coast of twenty-seven inland thirty degrees Bowral enjoying twenty-seven and Katoomba twenty-five degrees currently around town on the coast it's seventeen that's four below <,> r Richmond and Bankstown are fifteen degrees Penrith sixteen Katoomba thirteen and Gosford twelve. One of the jewels in the open garden scheme crown is opening today and this is just a garden to envy how would you like <,> to have <,> a beautiful sandstone cottage nestled underneath a waterfall with a little pond and then a creek that runs through with thousands of water dragons so tame they come up and just <,> kiss you. Would you like to live there.	P1	He's also known for his ability to open cosposting {composting} toilets so he can tell you anything worm farm problems certainly helped us and although I'm still confused about dry ingredients we might talk about that as well but eight-triple-three-one-thousand one-eight-hundred-eight-hundred-seven-oh-two fine sunny day today top temperatures on the coast of twenty-seven inland thirty degrees Bowral enjoying twenty-seven and Katoomba twenty-five degrees currently around town on the coast it's seventeen that's four below <,> r Richmond and Bankstown are fifteen degrees Penrith sixteen Katoomba thirteen and Gosford twelve. One of the jewels in the open garden scheme crown is opening today and this is just a garden to envy how would you like <,> to have <,> a beautiful sandstone cottage nestled underneath a waterfall with a little pond and then a creek that runs through with thousands of water dragons so tame they come up and just <,> kiss you. Would you like to live there.
Australian Radio Talkback	ABCE1	[E1] Okay.	E1	Okay.
Australian Radio Talkback	ABCE1	[P1] Jeanne Villani does and we'll find out the secret of her open garden and give you the address so that you can go along today and tomorrow to see Waterfall Cottage which is a part of the open garden scheme all this and more because it is Saturday.	P1	Jeanne Villani does and we'll find out the secret of her open garden and give you the address so that you can go along today and tomorrow to see Waterfall Cottage which is a part of the open garden scheme all this and more because it is Saturday.
Australian Radio Talkback	ABCE1	{program advert}	{program advert}	{program advert}

Load GRI

We continue with the content of the files of the *Griffith Corpus of Spoken Australian English* (gri).

```
vgri <- sapply(fgri, function(x){
  x <- readLines(x, encoding = "UTF-8")
  x <- x[x != ""]
  x <- x[!stringr::str_detect(x, "\\|.*\\|")]
  x <- paste0(x, collapse = " ")
  x <- stringr::str_split(stringr::str_replace_all(x, "( [A-Z]:)",
"qwertz\\1"), "qwertz")
  x <- unlist(x)
  x <- stringr::str_squish(x)
})
gritext <- unlist(vgri)
# collapse into df
gridf <- data.frame(names(gritext), names(gritext),gritext) %>%
  dplyr::rename(corpus = colnames(.)[1],
                file = colnames(.)[2],
                text = colnames(.)[3]) %>%
  dplyr::mutate(corpus = stringr::str_replace_all(corpus, ".*data/(.*?)/.*",
"\\1"),
                file = stringr::str_replace_all(file, ".*Raw/(.*?)-raw.*",
"\\1"),
                speaker = stringr::str_remove_all(text, ":.*"),
                speaker = stringr::str_remove_all(speaker, "\\W.*\\W"),
                speaker = stringr::str_remove_all(speaker, "[^[:alpha:]]"),
                speaker = stringr::str_remove_all(speaker, "[a-z]"),
                textclean = stringr::str_remove_all(text, "<.*?>"),
                textclean = stringr::str_remove_all(textclean, "(.*?)"),
                textclean = stringr::str_remove(textclean, "[0-9]{0,} {0,}[A-
Z]{1,}:"),
                textclean = stringr::str_remove_all(textclean, "[^[:alpha:]'
]"),
                textclean = stringr::str_squish(textclean))
rownames(gridf) <- NULL
# inspect
knitr::kable(head(gridf))
```

corpus	file	text	speaker	textclean
Griffith Corpus of Spoken Australian English	GCSAusE01	Transcript Coversheet 1	T	Transcript Coversheet
Griffith Corpus of Spoken Australian English	GCSAusE01	S: I'm glad I saw you? I thought I lost you 2 [( ) 3	S	I'm glad I saw you I thought I lost you
Griffith Corpus of Spoken Australian English	GCSAusE01	H: [no I've been here for a whi:le, 4	H	no I've been here for a while
Griffith Corpus of Spoken Australian English	GCSAusE01	S: hm:: if ↑I couldn't bo↓rrow, (1.3) second (.) book of reading 5 (.) f[o:r 6	S	hm if I couldn't borrow second book of reading for
Griffith Corpus of Spoken Australian English	GCSAusE01	H: [fo:r commu:nicating at cro- no: for family gender and 7 sexuality= 8	H	for communicating at cro no for family gender and sexuality
Griffith Corpus of Spoken Australian English	GCSAusE01	S: =ah: that's the second e:dition?= 9	S	ah that's the second edition

Load MON

We continue with the content of the files of the *Monash corpus* (mon)

```
vmon <- sapply(fmon, function(x){
  x <- readLines(x, encoding = "UTF-8")
  x <- x[x != ""]
  x <- paste0(x, collapse = "qwertz") %>%
  stringr::str_remove_all("qwertz    ") %>%
  stringr::str_split("qwertz") %>%
  unlist() %>%
  stringr::str_squish()
  })
# unlist
montext <- unlist(vmon)
# collapse into df
mondf <- data.frame(names(montext), names(montext), montext) %>%
  dplyr::rename(corpus = colnames(.)[1],
                file = colnames(.)[2],
                text = colnames(.)[3]) %>%
  dplyr::mutate(corpus = stringr::str_replace_all(corpus, ".*data/(.*?)/.*",
"\\1"),
                file = stringr::str_replace_all(file, ".*Text/(.*?).txt",
"\\1"),
                speaker = paste0("NA"),
                textclean = stringr::str_squish(text))
rownames(mondf) <- NULL
# inspect
knitr::kable(head(mondf))
```

corpus	file	text	speaker	textclean
Monash	MEBH1M1_Sanitised-plain1	Now, REDACTED. Could you tell me your name please.	NA	Now, REDACTED. Could you tell me your name please.
Monash	MEBH1M1_Sanitised-plain2	My name's REDACTED. I'm -	NA	My name's REDACTED. I'm -
Monash	MEBH1M1_Sanitised-plain3	Okay	NA	Okay
Monash	MEBH1M1_Sanitised-plain4	I'm fifteen years old.	NA	I'm fifteen years old.
Monash	MEBH1M1_Sanitised-plain5	Fifteen?	NA	Fifteen?
Monash	MEBH1M1_Sanitised-plain6	Yes.	NA	Yes.

Load LAT

We continue with the content of the files of the *The La Trobe Corpus of Spoken Australian English* (lat)

```
vlat <- sapply(flat, function(x){
  x <- readLines(x, encoding = "UTF-8")
  x <- x[x != ""]
  })
lattext <- unlist(vlat)
# collapse into df
latdf <- data.frame(names(lattext), names(lattext),lattext) %>%
  dplyr::rename(corpus = colnames(.)[1],
```

```
                file = colnames(.)[2],
                text = colnames(.)[3]) %>%
  dplyr::mutate(corpus = stringr::str_replace_all(corpus, ".*data/(.*?)/.*",
"\\1"),
                file = stringr::str_replace_all(file, ".*Raw/(.*?)-raw.*",
"\\1"),
                speaker = stringr::str_remove_all(text, ":.*"),
                speaker = stringr::str_remove_all(speaker, "\\W.*\\W"),
                speaker = stringr::str_remove_all(speaker, "[^[:alpha:]]"),
                speaker = stringr::str_remove_all(speaker, "[a-z]"),
                textclean = stringr::str_remove_all(text, "^[A-
Z]{1,}:{0,1}"),
                textclean = stringr::str_squish(textclean))
rownames(latdf) <- NULL
# inspect
knitr::kable(head(latdf))
```

corpus	file	text	speaker	textclean
The La Trobe Corpus of Spoken Australian English	Transcrp - Beth & Daniel	TRANSCRIPTION of Beth & Daniel	D	TRANSCRIPTION of Beth & Daniel
The La Trobe Corpus of Spoken Australian English	Transcrp - Beth & Daniel	B: …no problem	B	…no problem
The La Trobe Corpus of Spoken Australian English	Transcrp - Beth & Daniel	K: Ok Daniel I might get you to just tell me what you miss most about France?	K	Ok Daniel I might get you to just tell me what you miss most about France?
The La Trobe Corpus of Spoken Australian English	Transcrp - Beth & Daniel	D: What I miss most [about France]	D	What I miss most [about France]
The La Trobe Corpus of Spoken Australian English	Transcrp - Beth & Daniel	K [yeah] since you've been away	K	[yeah] since you've been away
The La Trobe Corpus of Spoken Australian English	Transcrp - Beth & Daniel	D: Aaah partly food even if in Australia the food is not so bad	D	Aaah partly food even if in Australia the food is not so bad

Collapse data into one table

We now combine the corpora into a single data frame called *oz*.

```
oz <- rbind(artdf, gridf, mondf, latdf)
# inspect
knitr::kable(head(oz))
```

corpus	file	text	speaker	textclean
Australian Radio Talkback	ABCE1	[Presenter 1: Simon Marnie, M] Thanks for that John Hall now John Hall will be listening for the next hour 'cos Angus Stewart is here to take your calls eight-triple-three-one-thousand one-eight-hundred-eight-hundred-seven-oh-two something in the garden that's causing you problems give us a call right now and Angus can I mean y'know he is known in the trade as Mr popergation {propagation} Mr propagation. He's also known for his passion for natives and his love of o orchids am I right so far.	Presenter 1: Simon Marnie, M	Thanks for that John Hall now John Hall will be listening for the next hour 'cos Angus Stewart is here to take your calls eight-triple-three-one-thousand one-eight-hundred-eight-hundred-seven-oh-two something in the garden that's causing you problems give us a call right now and Angus can I mean y'know he is known in the trade as Mr popergation {propagation} Mr propagation. He's also known for his passion for natives and his love of o orchids am I right so far.
Australian Radio Talkback	ABCE1	[Expert 1: Angus Stewart, M] I guess yeah yeah .	Expert 1: Angus Stewart, M	I guess yeah yeah .
Australian Radio Talkback	ABCE1	[P1] He's also known for his ability to open cosposting {composting} toilets so he can tell you anything worm farm problems certainly helped us and although I'm still confused about dry ingredients we might talk about that as well but eight-triple-three-one-thousand one-eight-hundred-eight-hundred-seven-oh-two fine sunny day today top temperatures on the coast of twenty-seven inland thirty degrees Bowral enjoying twenty-seven and Katoomba twenty-five degrees currently around town on the coast it's seventeen that's four below <,> r Richmond and Bankstown are fifteen degrees Penrith sixteen Katoomba thirteen and Gosford twelve. One of the jewels in the open garden scheme crown is opening today and this is just a garden to envy how would you like <,> to have <,> a beautiful sandstone cottage nestled underneath a waterfall with a little pond and then a creek that runs through with thousands of water dragons so tame they come up and just <,> kiss you. Would you like to live there.	P1	He's also known for his ability to open cosposting {composting} toilets so he can tell you anything worm farm problems certainly helped us and although I'm still confused about dry ingredients we might talk about that as well but eight-triple-three-one-thousand one-eight-hundred-eight-hundred-seven-oh-two fine sunny day today top temperatures on the coast of twenty-seven inland thirty degrees Bowral enjoying twenty-seven and Katoomba twenty-five degrees currently around town on the coast it's seventeen that's four below <,> r Richmond and Bankstown are fifteen degrees Penrith sixteen Katoomba thirteen and Gosford twelve. One of the jewels in the open garden scheme crown is opening today and this is just a garden to envy how would you like <,> to have <,> a beautiful sandstone cottage nestled underneath a waterfall with a little pond and then a creek that runs through with thousands of water dragons so tame they come up and just <,> kiss you. Would you like to live there.
Australian Radio Talkback	ABCE1	[E1] Okay.	E1	Okay.
Australian Radio Talkback	ABCE1	[P1] Jeanne Villani does and we'll find out the secret of her open garden and give you the address so that you can go along today and tomorrow to see Waterfall Cottage which is a part of the open garden scheme all this and more because it is Saturday.	P1	Jeanne Villani does and we'll find out the secret of her open garden and give you the address so that you can go along today and tomorrow to see Waterfall Cottage which is a part of the open garden scheme all this and more because it is Saturday.
Australian Radio Talkback	ABCE1	{program advert}	{program advert}	{program advert}

Extract UF-or

In a next step, we extract utterances with utterance final *or*. We determine this by checking if a string (utterance) ends with the sequence *or* but we allow for another words to come after the or if it has up to three chacraters (e.g., "… or uhm?").

```
ufor <- oz %>%
  dplyr::mutate(ufor = ifelse(stringr::str_detect(textclean, " or
{0,}.{0,3}$"), 1, 0)) %>%
  dplyr::filter(ufor == 1)
# inspect
knitr::kable(head(ufor$textclean))
```

x
In the paper bag and then what you sprinkle the paper bag over the moss or.
Playing piano or um.
Oil for decking oil oil for painting decks or.
Look um we're looking at buying um a fairly old house about thirty years old and it's fibro lined. Um <,> uh I'm not too sure of the distan the difference between asbestos and fibro I want to know are there any uh concerns for health reasons or.
Does it devalue a property ih like if ih ih um when you resell or.
Are the ceilings {break} battened or.

Next, we want to extract concordances (keywords-in-context) of potential hits (utterance-final *or*). The context should be two utterances preceding the utterance with utterance-final *or* and two utterances following the instance of utterance-final *or*.

```
inds = which(stringr::str_detect(oz$textclean, " or {0,}.{0,3}$"))
# We use lapply() to get all rows for all indices, result is a list
rows <- lapply(inds, function(x) (x-2):(x+2))
# With unlist() you get all relevant rows
ufors <- oz[unlist(rows),]
# insepct
knitr::kable(head(ufors, 10))
```

	corpus	file	text	speaker	textclean
132	Australian Radio Talkback	ABCE1	[C5] Yeah I would someone told me to grow moss . On the rock and then put the seed on the moss and I just <,> I've grown dehn well I've had Dendrobiums growing for years and I've never gr seen anything that w could recognise as a seed that was all.	C5	Yeah I would someone told me to grow moss . On the rock and then put the seed on the moss and I just <,> I've grown dehn well I've had Dendrobiums growing for years and I've never gr seen anything that w could recognise as a seed that was all.
133	Australian Radio Talkback	ABCE1	[E1] Yes yeah well that's uh probably 'cos they're so microscopic you you actually they're they're like specks of dust so you really uh if you get one of the seed pods and uh split it open you'll find it contains tens of thousands of of individual seeds so it's it's really a matter of catching the pod uh as it's ripening and um y you can actually put it into a paper bag and and you collect all that seed.	E1	Yes yeah well that's uh probably 'cos they're so microscopic you you actually they're they're like specks of dust so you really uh if you get one of the seed pods and uh split it open you'll find it contains tens of thousands of of individual seeds so it's it's really a matter of catching the pod uh as it's ripening and um y you can actually put it into a paper bag and and you collect all that seed.
134	Australian Radio Talkback	ABCE1	[P1] In the paper bag and then what you sprinkle the paper bag over the moss or.	P1	In the paper bag and then what you sprinkle the paper bag over the moss or.
135	Australian Radio Talkback	ABCE1	[E1] Yeah well what uh what you could do is get a seed tray put some uh sphagnum moss in that and then uh put the seed much as like is done with ferns spores where uh yeah you get a seed tray and then put a sheet of glass over the top to to uh keep the humih humidity up while the seeds are germinating. And um yeah that that will give you generally a a higher success rate.	E1	Yeah well what uh what you could do is get a seed tray put some uh sphagnum moss in that and then uh put the seed much as like is done with ferns spores where uh yeah you get a seed tray and then put a sheet of glass over the top to to uh keep the humih humidity up while the seeds are germinating. And um yeah that that will give you generally a a higher success rate.
136	Australian Radio Talkback	ABCE1	[C5] Well that sounds excellent thank you very much and just as a little aside.	C5	Well that sounds excellent thank you very much and just as a little aside.
382	Australian Radio Talkback	ABCE1	[P1] Now uh let's just <,> hear a little bit no it's not there anymore let's see whether uh Beth wants to take Lynne's place do you Beth.	P1	Now uh let's just <,> hear a little bit no it's not there anymore let's see whether uh Beth wants to take Lynne's place do you Beth.
383	Australian Radio Talkback	ABCE1	[Caller 10: Beth, F] I do indeed thank you so much .	Caller 10: Beth, F	I do indeed thank you so much .
384	Australian Radio Talkback	ABCE1	[P1] Playing piano or um.	P1	Playing piano or um.
385	Australian Radio Talkback	ABCE1	[C10] Definitely not no no.	C10	Definitely not no no.
386	Australian Radio	ABCE1	[P1] You've got a cycad question.	P1	You've got a cycad question.

We now generate a table with the instances of utterance-final *or* and the preceding as well as subsequent utterances and save the data to out computer for the manual annotation of the functions of utterance-final *or*.

```
# label instances
nhits <- sapply(rows, function(x){ length(x) })
nints <- 1:length(rows)
labs <- rep(paste0("instance ", nints), each = nhits)
# label context
contlabs <- rep(c("pre2", "pre1", "hit", "post1", "post2"), length(rows))
# add to kwics
ufors <- ufors %>%
  dplyr::mutate(hit = labs,
                context = contlabs) %>%
  dplyr::select(-speaker, -textclean) %>%
  dplyr::relocate(corpus, file, hit, context, text)
```

The data frame now contains five lines for each instance: *pre2*, *pre1*, *hit*, *post1*, and *post2*. The instance of utterance-final *or* is shown in the row labeled as *hit*. The table below shows the first 10 lines of the data frame (i.e., 2 instances of utterance-final *or* plus two utterances before the instance, labelled *pre2* and *pre1*, and two utterances after the instance of utterance-final *or*, labelled *post1* and *post2*).

```
# inspect
knitr::kable(head(ufors, 10))
```

	corpus	file	hit	context	text
132	Australian Radio Talkback	ABCE1	instance 1	pre2	[C5] Yeah I would someone told me to grow moss . On the rock and then put the seed on the moss and I just <,> I've grown dehn well I've had Dendrobiums growing for years and I've never gr seen anything that w could recognise as a seed that was all.
133	Australian Radio Talkback	ABCE1	instance 1	pre1	[E1] Yes yeah well that's uh probably 'cos they're so microscopic you you actually they're they're like specks of dust so you really uh if you get one of the seed pods and uh split it open you'll find it contains tens of thousands of of individual seeds so it's it's really a matter of catching the pod uh as it's ripening and um y you can actually put it into a paper bag and and you collect all that seed.
134	Australian Radio Talkback	ABCE1	instance 1	hit	[P1] In the paper bag and then what you sprinkle the paper bag over the moss or.
135	Australian Radio Talkback	ABCE1	instance 1	post1	[E1] Yeah well what uh what you could do is get a seed tray put some uh sphagnum moss in that and then uh put the seed much as like is done with ferns spores where uh yeah you get a seed tray and then put a sheet of glass over the top to to uh keep the humih humidity up while the seeds are germinating. And um yeah that that will give you generally a a higher success rate.
136	Australian Radio Talkback	ABCE1	instance 1	post2	[C5] Well that sounds excellent thank you very much and just as a little aside.
382	Australian Radio Talkback	ABCE1	instance 2	pre2	[P1] Now uh let's just <,> hear a little bit no it's not there anymore let's see whether uh Beth wants to take Lynne's place do you Beth.
383	Australian Radio Talkback	ABCE1	instance 2	pre1	[Caller 10: Beth, F] I do indeed thank you so much .
384	Australian Radio Talkback	ABCE1	instance 2	hit	[P1] Playing piano or um.
385	Australian Radio Talkback	ABCE1	instance 2	post1	[C10] Definitely not no no.
386	Australian Radio Talkback	ABCE1	instance 2	post2	[P1] You've got a cycad question.

Save data to disc for manual annotation

We now save the data so that we can annotate and code the data manually in a spreadsheet software (MS Excel).

```
# save
write.xlsx(ufors, here::here("tables", "step1_ufor_complete.xlsx"), sheetName
= "Sheet1",
           colNames = TRUE,
           rowNames = TRUE,
           append = FALSE)
```

Step 2: Manual annotation

This section details the annotation scheme used to manually annotate the instances of UF-or in a spreadsheet software (MS Excel).

Manual annotation focused on:

1. action format (Question or Assertion)
2. question type (polar, alternative, Q-word), and
3. identification of false positives (FP)

UF-or data annotation scheme

The annotation scheme used to code individual instances of utterance-final *or* is provided below. Each instance was inspected and annotated with regard to the categories shown below.

Action format	Question type	Response polarity	Response elaboration	Response alignment
Question [Q]	Information-seeking question [Q]	Yes [Y]	Explicit (yes/no, direct repeat) [E]	Type-conforming (yes/no; A or B) [TC]
Assertion [A]	Polar question [P]	No [N]	Not explicit [NE]	Non-type-conforming [NTC]
	Alternative question [A]	Yes-No [Y-N]		
	False positive (i.e. not a question) [FP]	B-answer [B]		
		No Answer [NoA]		

INTERACTIVE CODE BELOW

THE CODE CHUNKS BELOW ARE INTERACTIVE (EXECUTABLE) WHICH ALLOWS YOU TO INSPECT THE DATA AND PROBE IT IN GREATER DETAIL.

Data Exploration and Analysis

We now load the manually annotated data and check what the data looks like.

```
ufor_step2 <- openxlsx::read.xlsx(here::here("tables",
"step2_ufors_qa_annotated.xlsx"), sheet = 1)
# inspect
```

```
ufor_step2 %>%
  dplyr::filter(corpus == "The La Trobe Corpus of Spoken Australian English")
%>%
  # show first 10 rows
  head(10) %>%
  # show results as a table
  knitr::kable()
```

X1	corpus	file	hit	context	text	action.format	question.type
22687	The La Trobe Corpus of Spoken Australian English	Transcrp - Beth & Daniel	instance 81	pre2	'cos of to avoid trouble or to avoid um you know like conflict or something (?)	NA	NA
22688	The La Trobe Corpus of Spoken Australian English	Transcrp - Beth & Daniel	instance 81	pre1	K: mmm	NA	NA
22689	The La Trobe Corpus of Spoken Australian English	Transcrp - Beth & Daniel	instance 81	hit	B: but I'm not sure I don't know if that's um general or not	A	FP
22690	The La Trobe Corpus of Spoken Australian English	Transcrp - Beth & Daniel	instance 81	post1	D: I think you you should have and you must have an opinion you should have your own opinion which is crazy um I tends to	NA	NA
2269	The La	Transcr	instanc	post2	express	NA	NA

X1	corpus	file	hit	context	text	action.format	question.type
1	Trobe Corpus of Spoken Australian English	p - Beth & Daniel	e 81		myself yeah and perhaps in a way it's easier for me for example in English in English country 'cos I can do it and		
22756	The La Trobe Corpus of Spoken Australian English	Transcrp - Beth & Daniel	instance 82	pre2	um but it's not necessarily always bad to be opinionated you could use it when it's not not being so negative	NA	NA
22757	The La Trobe Corpus of Spoken Australian English	Transcrp - Beth & Daniel	instance 82	pre1	D: but it does it mean having a strong opinion on everything or have just opinion on everything which is different .. no?	NA	NA
22758	The La Trobe Corpus of Spoken Australian English	Transcrp - Beth & Daniel	instance 82	hit	B: ha.. having a strong opinion or =	Q	FP

X1	corpus	file	hit	context	text	action.format	question.type
22759	The La Trobe Corpus of Spoken Australian English	Transcrp - Beth & Daniel	instance 82	post1	D: = yeah [fighting]	NA	NA
22760	The La Trobe Corpus of Spoken Australian English	Transcrp - Beth & Daniel	instance 82	post2	B: [or just an opin..]	NA	NA

The table has the following columns:

- X1: an identifier value allowing us to unambiguously identifying every row in the data.

- corpus: the name of the corpus in which the potential instance occurred
- file: the file in which the potential instance occurred
- hit: the number of the potential instance

- context: specification of whether the text shows previous utterances (pre2 and pre1), the instance itself (hit), or subsequent utterances (post1 and post2)

- text: the utterance preceding, containing, or following a potential instance of UF-or

- action.format: Question or Assertion
- question.type: Polar, Alternative, Q-word

Most of the cells are empty and do not contain any annotation information (these are all cells containing *NA* which stands for *not applicable*).

We continue by cleaning the data, for example, by replacing `NA` and renaming columns and variable levels to be easier to understand

```
ufor_step2_clean <- ufor_step2 %>%
  dplyr::group_by(hit) %>%
  tidyr::fill(action.format, .direction = "updown") %>%
  tidyr::fill(question.type, .direction = "updown") %>%
  # rename
  dplyr::rename(`Action Format` = action.format,
                `Question Type` = question.type) %>%
  # renaming levels
  dplyr::mutate(`Action Format` = factor(`Action Format`,
                                         levels = c("Q", "A"),
```

```
                                        labels = c("Question (interrogative)",
"Assertion (declarative)")),
                `Question Type`  = factor(`Question Type`,
                                        levels = c("P", "A", "Q", "FP"),
                                        labels = c("Polar question",
"Alternative question", "Q-word question", "False positive"))) %>%
  # remove grouping
  dplyr::ungroup()
# inspect
head(ufor_step2_clean, 5)
```

```
## # A tibble: 5 × 8
##   X1    corpus                    file  hit        context text  Actio…¹
Quest…²
##   <chr> <chr>                     <chr> <chr>      <chr>   <chr> <fct>
<fct>
## 1 132   Australian Radio Talkback ABCE1 instance 1 pre2    [C5]… Questi…
Polar …
## 2 133   Australian Radio Talkback ABCE1 instance 1 pre1    [E1]… Questi…
Polar …
## 3 134   Australian Radio Talkback ABCE1 instance 1 hit     [P1]… Questi…
Polar …
## 4 135   Australian Radio Talkback ABCE1 instance 1 post1   [E1]… Questi…
Polar …
## 5 136   Australian Radio Talkback ABCE1 instance 1 post2   [C5]… Questi…
Polar …
## # … with abbreviated variable names ¹`Action Format`, ²`Question Type`
```

We now create a first overview table showing how many instances there are per Action Format.

```
ufor_step2_clean %>%
    dplyr::filter(context == "hit") %>%
  dplyr::mutate(`Question Type` = ifelse(`Question Type` == "False positive",
"False positive", "N")) %>%
  dplyr::group_by(`Action Format`,`Question Type`) %>%
  dplyr::summarise(N = n()) %>%
  tidyr::spread(`Question Type`, N) %>%
  dplyr::mutate(N = N + `False positive`) %>%
  dplyr::relocate(`False positive`, .after = N) %>%
  replace(is.na(.), 0)
```

```
## `summarise()` has grouped output by 'Action Format'. You can override
using the
## `.groups` argument.
```

```
## # A tibble: 2 × 3
## # Groups:   Action Format [2]
##   `Action Format`              N `False positive`
##   <fct>                    <int>            <int>
## 1 Question (interrogative)    73               10
## 2 Assertion (declarative)      0               25
```

Candidate UF-or information seeking questions (N=63: 73-10)

```
ufor_step2_clean %>%
  dplyr::filter(context == "hit") %>%
  dplyr::filter(`Question Type` != "False positive") %>%
  dplyr::group_by(`Question Type`) %>%
  dplyr::summarise(N = n()) %>%
  dplyr::add_row(`Question Type` = "Total",
                 N = sum(.$N))
```

```
## # A tibble: 4 × 2
##   `Question Type`          N
##   <chr>                <int>
## 1 Polar question          55
## 2 Alternative question     7
## 3 Q-word question          1
## 4 Total                   63
```

False positives

All false positives combined.

```
ufor_step2_clean %>%
  dplyr::filter(context == "hit") %>%
  dplyr::filter(`Question Type` == "False positive") %>%
  dplyr::group_by(`Question Type`) %>%
  dplyr::summarise(N = n())
```

```
## # A tibble: 1 × 2
##   `Question Type`     N
##   <fct>           <int>
## 1 False positive     35
```

1. Assertions (N=25)

As we were interested in UF-or questions, assertions were by definition false positives (although an interesting phenomenon in its own right).

For example *Suggestion/advice marked with UF-or* (N=1)

ART: COME3 (instance 13):

```
ufor_step2_clean %>%
  dplyr::filter(hit == "instance 13") %>%
  dplyr::filter(context == "hit"| context == "post1") %>%
  dplyr::mutate(text = stringr::str_remove_all(text, ".*>. ")) %>%
  dplyr::select(text)
```

```
## # A tibble: 2 × 1
##   text
##   <chr>
## 1 If the vomiting is still there on Monday morning it certainly is worth a
trip…
```

```
## 2 [C12] I can't have Bonox because it's two high in salt for the
Meniere's.
```

2. Interrogatives (N=10)

a. FP due to question being request for permission rather than request for information (N=1)

ART: COME3 (instance 16)

```
ufor_step2_clean %>%
  dplyr::filter(hit == "instance 16") %>%
  dplyr::filter(context != "post2") %>%
  dplyr::select(text)
```

```
## # A tibble: 4 × 1
##   text
##   <chr>
## 1 [C14] Yes. And I was just gunna ask about um how he and I are
compatible.
## 2 [P2] I've only got a minute to the news I don't think I can add all that
up.
## 3 [C14] Oh okay <P2 but I could do> well anything <P2 I could> about him
or.
## 4 [P2] Yes I I can j I can do that for you okay uh five and nine's
fourteen's f…
```

b. FP due to instance being utterance-medial rather than utterance-final (N=3)

MCE: MECG1M1 (instance 54)

```
ufor_step2_clean %>%
  dplyr::filter(hit == "instance 54") %>%
  dplyr::select(text)
```

```
## # A tibble: 5 × 1
##   text
##   <chr>
## 1 Well, say you've done something wrong, you
## 2 Yeap.
## 3 And you have been punished by your uh parents or
## 4 Yeap.
## 5 by the school.
```

LTCE: Beth & Daniel (instance 82)

```
ufor_step2_clean %>%
  dplyr::filter(hit == "instance 82") %>%
  dplyr::select(text)
```

```
## # A tibble: 5 × 1
##   text
##   <chr>
## 1 "      um but it’s not necessarily always bad to be opinionated you
```

```
could use…
## 2 "D:    but it does it mean having a strong opinion on everything or have
just…
## 3 "B:    ha.. having a strong opinion or ="
## 4 "D:                              = yeah [fighting]"
## 5 "B:                                   [or just an opin..]"
```

MCE: MESJ3F1 (instance 73):

```
ufor_step2_clean %>%
  dplyr::filter(hit == "instance 73") %>%
  dplyr::select(text)

## # A tibble: 5 × 1
##   text
##   <chr>
## 1 Um, we've talked about, oh, have you ever been in hospital yourself?
## 2 As in like, can you just
## 3 Have you ever had an accident or
## 4 (?)
## 5 or have you been in hospital or you thought you were going to
```

c. FP created through script (N=6), i.e.:

- *or not* (instance 42, 76, 92)
- *or no* (instance 78)
- *or so* (instance 22)
- *or two* (instance 41)

or not

```
ufor_step2_clean %>%
  dplyr::filter(`Question Type` == "False positive",
                context == "hit") %>%
  dplyr::filter(hit == "instance 42" | hit == "instance 76" | hit ==
"instance 92") %>%
  dplyr::select(text)

## # A tibble: 3 × 1
##   text
##   <chr>
## 1 J: [do you] wa↑nt breakfast or 157. no↓t= 158.
## 2 yeh I know so you wanna see whether they do or not
## 3 Ken:  That's the same as um … tolerant isn't it? or not
```

or no

```
ufor_step2_clean %>%
  dplyr::filter(`Question Type` == "False positive",
                context == "hit") %>%
```

```
  dplyr::filter(hit == "instance 78") %>%
  dplyr::select(text)
```

```
## # A tibble: 1 × 1
##   text
##   <chr>
## 1 RB: I think the intonation was bad. Can I say it again? or no?
```

or so

```
ufor_step2_clean %>%
  dplyr::filter(`Question Type` == "False positive",
                context == "hit") %>%
  dplyr::filter(hit == "instance 22") %>%
  dplyr::mutate(text = stringr::str_remove_all(text, ".*away. ")) %>%
  dplyr::select(text)
```

```
## # A tibble: 1 × 1
##   text
##   <chr>
## 1 Um do you see it going up much in the future or s just plugging away as
it ha…
```

or two

```
ufor_step2_clean %>%
  dplyr::filter(`Question Type` == "False positive",
                context == "hit") %>%
  dplyr::filter(hit == "instance 41") %>%
  dplyr::mutate(text = stringr::str_remove_all(text, ".* 19 ")) %>%
  dplyr::select(text)
```

```
## # A tibble: 1 × 1
##   text
##   <chr>
## 1 how many weeks after: hers is yours? >one week? 20 or two 21 (1.0) 22
```

False negative (N=1)

(identified by Haugh 2011 in GCSAusE, but not extracted through script) (GCSAusE011)

T: =so is it? (.) is it easy? o:r [like what]
B: [it's ha:]rd,

Step 3: Manual annotation

This section focuses on polar interrogatives.

Responses to UF-or polar interrogatives were manually annotated by analyst (bottom-up [data-driven] and top-down [previous studies] annotation schema) (n=55)

Manual annotation focused on:

1. response **polarity** (confirming [Y], disconfirming [N], (dis)confirming [Y-N], non-answers [NA])

2. response **alignment** (type-conforming [TC], non-type-conforming [NTC])

3. response **elaboration** (elaboration [E], no elaboration [NE])

We now load the manually annotated data and check what the data looks like.

```
ufor_step3 <- openxlsx::read.xlsx(here::here("tables",
"step3_ufors_qp_annotated.xlsx"), sheet = 1)
# inspect
ufor_step3 %>%
  dplyr::filter(corpus == "The La Trobe Corpus of Spoken Australian English")
%>%
  # show first 10 rows
  head(10) %>%
  # show results as a table
  knitr::kable()
```

X1	corpus	file	hit	context	text	action.format	question.type	response.polarity	response.elaboration	response.alignment
2302 7	The La Trobe Corpus of Spoken Australian English	Transcrp - Heather & Marie	instance 83	hit	think about a certain situation and you really .. hadn’t .. come to a conclusion would you say “may be maybe yes mayb	Q	P	NA	NA	NA

X1	corpus	file	hit	context	text	action.format	question.type	response.polarity	response.elaboration	response.alignment
					e no" or…					
23028	The La Trobe Corpus of Spoken Australian English	Transcrip - Heather & Marie	instance 83	post1	Marie: yeah well you have to ask me a question when I can answer yes or no well I have to answer yes or no so that I answer	NA	NA	YN	E	NTC
23029	The La Trobe Corpus of Spoken Australian English	Transcrip - Heather & Marie	instance 83	post2	"peut-être bien que oui, peut-être bien que non" =	NA	NA	NA	NA	NA
23026	The La Trobe	Transcrip - Hea	instance 83	pre1	Heather: it when you	NA	NA	NA	NA	NA

X1	corpus	file	hit	context	text	action.format	question.type	response.polarity	response.elaboration	response.alignment
	Corpus of Spoken Australian English	ther & Marie			weren't sure .. about something about your it's .. I sit here well what did you					
23025	The La Trobe Corpus of Spoken Australian English	Transcrip - Heather & Marie	instance 83	pre 2	Marie: [yeah it's]	NA	NA	NA	NA	NA
25769	The La Trobe Corpus of Spoken Australian English	Transcrip - Natalie & Ken	instance 96	hit	Kerry: What makes you say that the Europeans would be … like what ex..	Q	P	NA	NA	NA

X1	corpus	file	hit	context	text	action.format	question.type	response.polarity	response.elaboration	response.alignment
					..is that through experience? or ..					
25770	The La Trobe Corpus of Spoken Australian English	Transcrip - Natalie & Ken	instance 96	post1	Natalie: [um]	NA	NA	NoA	NE	NTC
25771	The La Trobe Corpus of Spoken Australian English	Transcrip - Natalie & Ken	instance 96	post2	Kerry: [do] you know someone or it's just a general impression you have?	NA	NA	NA	NA	NA
25768	The La Trobe Corpus of Spoken Aust	Transcrip - Natalie & Ken	instance 96	pre1	Natalie: but um ...	NA	NA	NA	NA	NA

X1	corpus	file	hit	context	text	action.format	question.type	response.polarity	response.elaboration	response.alignment
	ralian English									
25767	The La Trobe Corpus of Spoken Australian English	Transcrip - Natalie & Ken	instance 96	pre2	Ken: = yeah	NA	NA	NA	NA	NA

The table has the following columns:

- X1: an identifier value allowing us to unambiguously identifying every row in the data.
- corpus: the name of the corpus in which the potential instance occurred
- file: the file in which the potential instance occurred
- hit: the number of the potential instance
- context: specification of whether the text shows previous utterances (pre2 and pre1), the instance itself (hit), or subsequent utterances (post1 and post2)
- text: the utterance preceding, containing, or following a potential instance of UF-or
- action.format: Question or Assertion
- question.type: Polar, Alternative, Q-word
- response.polarity: Polarity of the response (post1) -confirming [Y], disconfirming [N], (dis)confirming [Y-N], non-answers [NA]

- response.elaboration: Elaboration of the response (post1) - elaboration [E], no elaboration [NE]

- response.alignment: Alignment of the response (post1) - type-conforming [TC], non-type-conforming [NTC]

Most of the cells are empty and do not contain any annotation information (these are all cells containing *NA* which stands for *not applicable*).

We continue by cleaning the data, for example, by replacing `NA` and renaming columns and variable levels to be easier to understand

```
ufor_step3_clean <- ufor_step3 %>%
  dplyr::mutate(question.type = str_squish(question.type)) %>%
  dplyr::group_by(hit) %>%
  tidyr::fill(action.format, .direction = "updown") %>%
  tidyr::fill(question.type, .direction = "updown") %>%
  tidyr::fill(response.polarity, .direction = "updown") %>%
  tidyr::fill(response.alignment, .direction = "updown") %>%
  tidyr::fill(response.elaboration, .direction = "updown") %>%
  # rename
  dplyr::rename(`Action Format` = action.format,
                `Question Type` = question.type,
                `Response Polarity` = response.polarity,
                `Response Alignment` = response.alignment,
                `Response Elaboration` = response.elaboration) %>%
  # renaming levels
  dplyr::mutate(`Action Format` = factor(`Action Format`,
                                         levels = c("Q", "A"),
                                         labels = c("Question (interrogative)",
"Assertion (declarative)")),
                `Question Type`  = factor(`Question Type`,
                                         levels = c("P", "A", "Q", "FP"),
                                         labels = c("Polar question",
"Alternative question", "Q-word question", "False positive")),
                `Response Polarity`  = factor(`Response Polarity`,
                                         levels = c("Y", "N", "YN", "B",
"NoA"),
                                         labels = c("Confirming [Y]",
"Disconfirming [N]", "(Dis)confirming [Y-N]", "B-answer [B]", "Non-answers
[NA]")),
                `Response Alignment`  = factor(`Response Alignment`,
                                         levels = c("TC", "NTC"),
                                         labels = c("Type-Conforming [TC]",
"Non-Type-Conforming [NTC]")),
                `Response Elaboration`  = factor(`Response Elaboration`,
                                         levels = c("E", "NE"),
                                         labels = c("Elaboration [E]", "No
elaboration [NE]"))) %>%
  # remove grouping
  dplyr::ungroup()
```

```
# inspect
head(ufor_step3_clean, 5)
```

```
## # A tibble: 5 × 11
##   X1    corpus file  hit   context text  Actio…¹ Quest…² Respo…³ Respo…⁴
Respo…⁵
##   <chr> <chr>  <chr> <chr> <chr>   <chr> <fct>   <fct>   <fct>   <fct>
<fct>
## 1 134   Austr… ABCE1 inst… hit     [P1]… Questi… Polar … Confir… Elabor…
Type-C…
## 2 135   Austr… ABCE1 inst… post1   [E1]… Questi… Polar … Confir… Elabor…
Type-C…
## 3 136   Austr… ABCE1 inst… post2   [C5]… Questi… Polar … Confir… Elabor…
Type-C…
## 4 133   Austr… ABCE1 inst… pre1    [E1]… Questi… Polar … Confir… Elabor…
Type-C…
## 5 132   Austr… ABCE1 inst… pre2    [C5]… Questi… Polar … Confir… Elabor…
Type-C…
## # … with abbreviated variable names ¹`Action Format`, ²`Question Type`,
## #   ³`Response Polarity`, ⁴`Response Elaboration`, ⁵`Response Alignment`
```

We now generate an overview tables.

Response Polarity

```
ufor_step3_clean %>%
    dplyr::filter(context == "hit") %>%
  dplyr::mutate(`Response Polarity` = dplyr::case_when(`Response Polarity` ==
"B-answer [B]" ~ "Non-polar",
                                                       `Response Polarity` ==
"Non-answers [NA]" ~ "Non-polar",
                                                       T ~ `Response
Polarity`)) %>%
  dplyr::group_by(`Response Polarity`) %>%
  dplyr::summarise(N = n()) %>%
  dplyr::ungroup() %>%
  dplyr::mutate(Total = sum(N)) %>%
  dplyr::rowwise() %>%
  dplyr::mutate(Percent = round(N/Total*100, 1)) %>%
  dplyr::select(-Total) %>%
  dplyr::add_row(`Response Polarity` = "Total",
                 N = sum(.$N),
                 Percent = sum(.$Percent)) %>%
  knitr::kable()
```

Response Polarity	N	Percent
(Dis)confirming [Y-N]	4	7.3
Confirming [Y]	25	45.5
Disconfirming [N]	19	34.5
Non-polar	7	12.7

Response Polarity	N	Percent
Total	55	100.0

```
ufor_step3_clean %>%
    dplyr::filter(context == "hit") %>%
  dplyr::mutate(`Response Polarity` = dplyr::case_when(`Response Polarity` ==
"B-answer [B]" ~ "B-answer [B]",
                                                       `Response Polarity` ==
"Non-answers [NA]" ~ "Non-answers [NA]",
                                                       T ~ "Polar")) %>%
  dplyr::group_by(`Response Polarity`) %>%
  dplyr::summarise(N = n()) %>%
  dplyr::ungroup() %>%
  dplyr::mutate(Total = sum(N)) %>%
  dplyr::rowwise() %>%
  dplyr::mutate(Percent = round(N/Total*100, 1)) %>%
  dplyr::select(-Total) %>%
  dplyr::add_row(`Response Polarity` = "Total",
                 N = sum(.$N),
                 Percent = sum(.$Percent)) %>%
  knitr::kable()
```

Response Polarity	N	Percent
B-answer [B]	2	3.6
Non-answers [NA]	5	9.1
Polar	48	87.3
Total	55	100.0

```
ufor_step3_clean %>%
    dplyr::filter(context == "hit") %>%
  dplyr::mutate(`Response Polarity` = dplyr::case_when(`Response Polarity` ==
"B-answer [B]" ~ "Non-polar",
                                                       `Response Polarity` ==
"Non-answers [NA]" ~ "Non-polar",
                                                       T ~ "Polar")) %>%
  dplyr::group_by(`Response Polarity`) %>%
  dplyr::summarise(N = n()) %>%
  dplyr::arrange(-N) %>%
  dplyr::ungroup() %>%
  dplyr::mutate(Total = sum(N)) %>%
  dplyr::rowwise() %>%
  dplyr::mutate(Percent = round(N/Total*100, 1)) %>%
  dplyr::select(-Total) %>%
  dplyr::add_row(`Response Polarity` = "Total",
                 N = sum(.$N),
                 Percent = sum(.$Percent)) %>%
  knitr::kable()
```

Response Polarity	N	Percent
Polar	48	87.3

Response Polarity	N	Percent
Non-polar	7	12.7
Total	55	100.0

Elaboration

```
ufor_step3_clean %>%
    dplyr::filter(context == "hit") %>%
  dplyr::mutate(`Response Polarity` = dplyr::case_when(`Response Polarity` ==
"B-answer [B]" ~ "Non-polar",
                                                       `Response Polarity` ==
"Non-answers [NA]" ~ "Non-polar",
                                                       T ~ "Polar")) %>%
  dplyr::filter(`Response Polarity` == "Polar") %>%
  dplyr::group_by(`Response Elaboration`) %>%
  dplyr::summarise(N = n()) %>%
  dplyr::arrange(-N) %>%
  dplyr::ungroup() %>%
  dplyr::mutate(Total = sum(N)) %>%
  dplyr::rowwise() %>%
  dplyr::mutate(Percent = round(N/Total*100, 1)) %>%
  dplyr::select(-Total) %>%
  dplyr::add_row(`Response Elaboration` = "Total",
                 N = sum(.$N),
                 Percent = sum(.$Percent)) %>%
  knitr::kable()
```

Response Elaboration	N	Percent
Elaboration [E]	36	75
No elaboration [NE]	12	25
Total	48	100

Response Alignment

```
ufor_step3_clean %>%
  dplyr::filter(context == "hit") %>%
  group_by(`Response Alignment`) %>%
  dplyr::summarise(N = n()) %>%
  dplyr::arrange(-N) %>%
  dplyr::ungroup() %>%
  dplyr::mutate(Total = sum(N)) %>%
  dplyr::rowwise() %>%
  dplyr::mutate(Percent = round(N/Total*100, 1)) %>%
  dplyr::select(-Total) %>%
  dplyr::add_row(`Response Alignment` = "Total",
                 N = sum(.$N),
                 Percent = sum(.$Percent)) %>%
  knitr::kable()
```

Response Alignment	N	Percent

Response Alignment	N	Percent
Type-Conforming [TC]	32	58.2
Non-Type-Conforming [NTC]	23	41.8
Total	55	100.0

```
ufor_step3_clean %>%
  dplyr::filter(context == "hit",
                `Response Alignment` == "Type-Conforming [TC]") %>%
  dplyr::mutate(`Response Polarity` = dplyr::case_when(`Response Polarity` ==
"Confirming [Y]" ~ "Confirming [Y]",
                                                       `Response Polarity` ==
"Disconfirming [N]" ~ "Disconfirming [N]",
                                                       TRUE ~ "other")) %>%
  group_by(`Response Alignment`, `Response Polarity`) %>%
  dplyr::summarise(Frequency = n()) %>%
  dplyr::arrange(-Frequency) %>%
  tidyr::spread(`Response Alignment`, Frequency) %>%
  replace(is.na(.), 0) %>%
  knitr::kable()
```

Response Polarity	Type-Conforming [TC]
Confirming [Y]	18
Disconfirming [N]	14

```
ufor_step3_clean %>%
  dplyr::filter(context == "hit",
                `Response Alignment` == "Non-Type-Conforming [NTC]") %>%
  dplyr::mutate(`Response Polarity` = dplyr::case_when(`Response Polarity` ==
"Confirming [Y]" ~ "Confirming [Y]",
                                                       `Response Polarity` ==
"Disconfirming [N]" ~ "Disconfirming [N]",
                                                       `Response Polarity` ==
"(Dis)confirming [Y-N]" ~ "(Dis)confirming [Y-N]",
                                                       TRUE ~ "Non-polar"))
%>%
  group_by(`Response Alignment`, `Response Polarity`) %>%
  dplyr::summarise(Frequency = n()) %>%
  dplyr::arrange(-Frequency) %>%
  tidyr::spread(`Response Alignment`, Frequency) %>%
  replace(is.na(.), 0) %>%
  dplyr::ungroup() %>%
  dplyr::mutate(Total = sum(`Non-Type-Conforming [NTC]`)) %>%
  dplyr::rowwise() %>%
  dplyr::mutate(Percent = round(`Non-Type-Conforming [NTC]`/Total*100, 1))
%>%
  dplyr::select(-Total) %>%
  dplyr::add_row(`Response Polarity` = "Total",
                 `Non-Type-Conforming [NTC]` = sum(.$`Non-Type-Conforming
[NTC]`),
                 Percent = sum(.$Percent)) %>%
  knitr::kable()
```

Response Polarity	Non-Type-Conforming [NTC]	Percent
(Dis)confirming [Y-N]	4	17.4
Confirming [Y]	7	30.4
Disconfirming [N]	5	21.7
Non-polar	7	30.4
Total	23	99.9

```
ufor_step3_clean %>%
  dplyr::filter(context == "hit",
                `Response Alignment` == "Non-Type-Conforming [NTC]") %>%
  dplyr::mutate(`Response Polarity` = dplyr::case_when(`Response Polarity` ==
"Confirming [Y]" ~ "Polar",
                                                       `Response Polarity` ==
"Disconfirming [N]" ~ "Polar",
                                                       `Response Polarity` ==
"(Dis)confirming [Y-N]" ~ "Polar",
                                                       TRUE ~ "Non-polar"))
%>%
  group_by(`Response Alignment`, `Response Polarity`) %>%
  dplyr::summarise(Frequency = n()) %>%
  dplyr::arrange(-Frequency) %>%
  tidyr::spread(`Response Alignment`, Frequency) %>%
  replace(is.na(.), 0) %>%
  dplyr::ungroup() %>%
  dplyr::mutate(Total = sum(`Non-Type-Conforming [NTC]`)) %>%
  dplyr::rowwise() %>%
  dplyr::mutate(Percent = round(`Non-Type-Conforming [NTC]`/Total*100, 1))
%>%
  dplyr::select(-Total) %>%
  dplyr::add_row(`Response Polarity` = "Total",
                 `Non-Type-Conforming [NTC]` = sum(.$`Non-Type-Conforming
[NTC]`),
                 Percent = sum(.$Percent)) %>%
  knitr::kable()
```

Response Polarity	Non-Type-Conforming [NTC]	Percent
Non-polar	7	30.4
Polar	16	69.6
Total	23	100.0

Step 4: Computational-interpretive analysis

Computationally analysed responses to UF-or information-seeking polar questions [Q-P] using pivot tables and manual close-reading (N=55)

1. UF-or information seeking questions are invariably responded as polar questions

2. UF-or makes elaboration a relevant next

1. UF-or information seeking questions are invariably responded as polar questions

Hypothesis: UF-or information-seeking questions are invariably responded to as polar questions (i.e. p or not?) rather than alternative questions (i.e. p or q?) (cf. Haugh 2011)

a. Distributional evidence

48/55 responses are confirming/disconfirming/(dis)confirming (i.e. respond to as polar Q) (87.3%)

2/55 responses are q responses (i.e. respond to as alternative Q) (3.6%)

4/55 responses are non-answer responses (i.e. equivocal as to whether treating it as polar Q) (7.3%)

b. Interpretive evidence

Alternative responses and non-answers are occasioned by teasing or repair (N=6).

Alternative answers are used as vehicles for teasing or repair (N=2):

1. (MCE: MEBH2FB, instance 50)

```
ufor_step2_clean %>%
  dplyr::filter(hit == "instance 50") %>%
  dplyr::filter(context != "pre2") %>%
  dplyr::select(text)
```

```
## # A tibble: 4 × 1
##   text
##   <chr>
## 1 BH2F: you can't smoke in here
## 2 BH2M: says who? what're you gonna do kick me out or
## 3 BH2F: I'm gonna jump on ya
## 4 BH2F: nah I'm gonna jump on you
```

q response to deliver counter-tease information-seeking question as vehicle for teasing challenge

2. (MCE: MECG2M1, instance 61)

```
ufor_step2_clean %>%
  dplyr::filter(hit == "instance 61") %>%
  dplyr::filter(context != "pre2") %>%
  dplyr::select(text)
```

```
## # A tibble: 4 × 1
##   text
##   <chr>
## 1 Mmm. Finally, how would you describe Melbourne to a visitor from
interstate?
## 2 Um, I think it's great. Um, what do you mean, like tell them where to go
```

```
or?
## 3 Anything.
## 4 Ah it's, I'd tell them that it's a really great place to stay because ah
you …
```

q response (confirming) post-first insert expansion of prior information-seeking (Q-word) question

Non-answer responses are used as vehicles for teases, responses to teases or repair (N=4)

1. (ART: NAT2, instance 27)

```
ufor_step2_clean %>%
  dplyr::filter(hit == "instance 27") %>%
  dplyr::filter(context != "pre2",
                context != "pre1") %>%
  dplyr::select(text)
```

```
## # A tibble: 3 × 1
##   text
##   <chr>
## 1 [C6] <laughs> <P2 laughs> What is what I wanted to ask Tim <,> and uh
relucta…
## 2 [P1] It's called good writing I think <laughs>.
## 3 [C6] <laughs> Yes indeed but <E1 oh> <inaudible> Tim feel about um Dylan
Thom…
```

non-answer response teasing Q-producer Q-producer pursues response to her question

(2) (ART: ABCE2, instance 3)

```
ufor_step2_clean %>%
  dplyr::filter(hit == "instance 3") %>%
  dplyr::filter(context != "pre2") %>%
  dplyr::select(text)
```

```
## # A tibble: 4 × 1
##   text
##   <chr>
## 1 [E2] It's a that's a <inaudible> <E1 inaudible>.
## 2 [E1] Oil for decking oil <P1 yeah> oil for painting decks <P1 yes> or.
## 3 [P1] Okay <E1 laughs>.
## 4 [C4] Okay then uh you've answered the question thanks very much Woodies.
```

non-answer response in response to tease stand-alone ‘or’ to jokingly bait recipient

(3) ART: COMNE1 (instance 18)

```
ufor_step2_clean %>%
  dplyr::filter(hit == "instance 18") %>%
  dplyr::select(text)
```

```
## # A tibble: 5 × 1
##   text
```

```
##   <chr>
## 1 [P2] What would that be like.
## 2 [P1] Oh well if you like stage shows it'd be fantastic I'm sure.
## 3 [P2] Yeah ih be ah be able to draw something outta that is it Simba was
that …
## 4 [P1] <laughs>.
## 5 [P2] Simba in that one.
```

non-answer response information-seeking question as vehicle for tease

(4) ART: COME4 (instance 14)

```
ufor_step2_clean %>%
  dplyr::filter(hit == "instance 14") %>%
  dplyr::filter(context != "pre2") %>%
  dplyr::select(text)
```

```
## # A tibble: 4 × 1
##   text
##   <chr>
## 1 [E1] So irrespective of what else is going on in your life you've got
sore ar…
## 2 [C15] Um once you spoke about a j a person with jaw pain and and I think
you …
## 3 [E1] They had T M J arthritis but Cathy.
## 4 [C15] Yeah.
```

non-answer response repair of terms of prior Q

2. UF-or makes elaboration a relevant next

UF-or makes elaboration a relevant next (cf. Drake 2015)

(n=49) NB. Alternative and non-answer responses (n=6) removed from count of (non)elaboration

a. Distributional evidence

(Dis)confirmation only (N=11) (22.4%) confirmation only (N=7) disconfirmation only (N=4)

(Dis)confirmation + elaboration (N=38) (77.6%) confirmation + elaboration (N=18) (dis)confirmation + elaboration (N=4) disconfirmation + elaboration (N=16)

b. Interpretive evidence: Deviant cases

non-production of elaboration treated as accountable absence

MCE: MEBH1MB (instance 47) [deviant case]

```
ufor_step2_clean %>%
  dplyr::filter(hit == "instance 47") %>%
  dplyr::select(text)
```

```
## # A tibble: 5 × 1
##   text
##   <chr>
## 1 all right I suppose
## 2 that’s good
## 3 tired . still tired from last night or
## 4 yeah
## 5 had a tiring week this week?
```

treats minimal confirmation as requiring elaboration

c. Interpretive evidence: Borderline cases

Bare confirmation/disconfirmation functions as “go ahead” response

ART: COMNE3 (instance 21)

```
ufor_step2_clean %>%
  dplyr::filter(hit == "instance 21") %>%
  dplyr::filter(context != "pre2",
                context != "pre1") %>%
  dplyr::select(text)
```

```
## # A tibble: 3 × 1
##   text
##   <chr>
## 1 [E1] Yeah look there you're you're right there're very few but look
these day…
## 2 [C8] Yeah yeah.
## 3 [E1] Yeah guy named Nigel Plowman and he operates <C8 ah yeah> his
company's …
```

MCE: MEBH2FB (instance 49)

```
ufor_step2_clean %>%
  dplyr::filter(hit == "instance 49") %>%
  dplyr::select(text)
```

```
## # A tibble: 5 × 1
##   text
##   <chr>
## 1 BH2M: where’re they goin’
## 2 Sound of car engine revving excessively
## 3 BH2M: are we gettin' in the cars or?
## 4 BH2F: yeah
## 5 BH2M: yeah..let’s go
```

MCE: MESJ3F1 (instance 72)

```
ufor_step2_clean %>%
  dplyr::filter(hit == "instance 72") %>%
  dplyr::filter(context != "pre2") %>%
  dplyr::select(text)
```

```
## # A tibble: 4 × 1
##   text
##   <chr>
## 1 What sort of things do you do together?
## 2 Out of school or
## 3 No
## 4 Well at school we just sort of sit down and talk about lots of things um
we u…
```

Bare confirmation occasioned by noticing of matter external to ongoing sequence

GCS: AusE32 (instance 44)

```
ufor_step2_clean %>%
  dplyr::filter(hit == "instance 44") %>%
  dplyr::filter(context != "pre2") %>%
  dplyr::select(text)
```

```
## # A tibble: 4 × 1
##   text
##   <chr>
## 1 S: I'll cut it 95 (1.7) 96
## 2 J: So yeah:: ya gonna drop into the u:m (1.8) you gonna drop into 97
the: ten…
## 3 S: Yeah man 100 (.) 101
## 4 J: Check [pic]that out[pic] 102 (.) 103
```

96 J: So yeah:: ya gonna drop into the u:m (1.8) you gonna drop into 97 the: tent embassy:: or:? 98 (1.4) 99 S: Yeah man 100 (.) 101 J: Check ↑that out↑

Bare disconfirmation for emphatic (repeated) rejection

emphatic rejection treats question as inapposite

ART: COMNE4 (instance 23)

```
ufor_step2_clean %>%
  dplyr::filter(hit == "instance 23") %>%
  dplyr::filter(context != "pre2",
                context != "pre1") %>%
  dplyr::select(text)
```

```
## # A tibble: 3 × 1
##   text
##   <chr>
## 1 [E2] Uh uh I think I might've had one of those at one stage is it um uh
is it…
## 2 [P1] No no no.
## 3 [E1] No <E2 hand chopper yes I've> no you hand one.
```

ART: ABCE1 (instance 2)

```
#ufor_step2_clean$hit[str_detect(ufor_step2_clean$text, "hear a little bit")]
ufor_step2_clean %>%
```

```
  dplyr::filter(hit == "instance 2") %>%
  dplyr::filter(context != "post2") %>%
  dplyr::select(text)
```

```
## # A tibble: 4 × 1
##   text
##   <chr>
## 1 [P1] Now uh let's just <,> hear a little bit no it's not there anymore
let's …
## 2 [Caller 10: Beth, F] I do indeed thank you so much <laughs>.
## 3 [P1] Playing piano or um.
## 4 [C10] Definitely not no no.
```

Further exploration of the data

This section provides pre-written code snippets that allow researchers to further explore the data.

Inspecting specific instances

There are overall 98 instances of potential instances of UF-or. You can access any of these (including the respective coding) if you modify the number following the sequence `"instance "` in the code below.

```
ufor_step2_clean %>%
  dplyr::filter(hit == "instance 2")
```

```
## # A tibble: 5 × 8
##   X1    corpus                    file  hit        context text  Actio…¹
Quest…²
##   <chr> <chr>                     <chr> <chr>      <chr>   <chr> <fct>
<fct>
## 1 382   Australian Radio Talkback ABCE1 instance 2 pre2    [P1]… Questi…
Polar …
## 2 383   Australian Radio Talkback ABCE1 instance 2 pre1    [Cal… Questi…
Polar …
## 3 384   Australian Radio Talkback ABCE1 instance 2 hit     [P1]… Questi…
Polar …
## 4 385   Australian Radio Talkback ABCE1 instance 2 post1   [C10… Questi…
Polar …
## 5 386   Australian Radio Talkback ABCE1 instance 2 post2   [P1]… Questi…
Polar …
## # … with abbreviated variable names ¹`Action Format`, ²`Question Type`
```

If you want to inspect the instances and coding in the data set loaded for step 3, you simply need to change `ufor_step2_clean` to `ufor_step3_clean`.

```
ufor_step3_clean %>%
  dplyr::filter(hit == "instance 2")
```

```
## # A tibble: 5 × 11
##   X1    corpus file  hit   context text  Actio…¹ Quest…² Respo…³ Respo…⁴
Respo…⁵
##   <chr> <chr>  <chr> <chr> <chr>   <chr> <fct>   <fct>   <fct>   <fct>
<fct>
## 1 384   Austr… ABCE1 inst… hit     [P1]… Questi… Polar … Discon… No ela…
Type-C…
## 2 385   Austr… ABCE1 inst… post1   [C10… Questi… Polar … Discon… No ela…
Type-C…
## 3 386   Austr… ABCE1 inst… post2   [P1]… Questi… Polar … Discon… No ela…
Type-C…
## 4 383   Austr… ABCE1 inst… pre1    [Cal… Questi… Polar … Discon… No ela…
Type-C…
## 5 382   Austr… ABCE1 inst… pre2    [P1]… Questi… Polar … Discon… No ela…
Type-C…
## # … with abbreviated variable names ¹`Action Format`, ²`Question Type`,
## #   ³`Response Polarity`, ⁴`Response Elaboration`, ⁵`Response Alignment`
```

Tabulation can be done by filtering the row containing the instance of UF-or and then grouping and summarizing based on what you want to tabulate.

For example, if you want to inspect the number of false positives among question types, you could use the command below.

```
ufor_step2_clean %>%
  dplyr::filter(context == "hit") %>%
  group_by(`Question Type`) %>%
  summarise(Frequency = n())
```

```
## # A tibble: 4 × 2
##   `Question Type`      Frequency
##   <fct>                    <int>
## 1 Polar question              55
## 2 Alternative question         7
## 3 Q-word question              1
## 4 False positive              35
```

Or, if you want to inspect the number of question types across , you could use the command below.

```
ufor_step2_clean %>%
  dplyr::filter(context == "hit") %>%
  group_by(`Question Type`, `Action Format`) %>%
  summarise(Frequency = n())
```

```
## `summarise()` has grouped output by 'Question Type'. You can override
using the
## `.groups` argument.

## # A tibble: 5 × 3
## # Groups:   Question Type [4]
##   `Question Type`      `Action Format`          Frequency
##   <fct>                <fct>                        <int>
```

```
## 1 Polar question       Question (interrogative)        55
## 2 Alternative question Question (interrogative)         7
## 3 Q-word question      Question (interrogative)         1
## 4 False positive       Question (interrogative)        10
## 5 False positive       Assertion (declarative)         25
```

Outro

```
sessionInfo()
```

```
## R version 4.2.2 (2022-10-31 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19045)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=German_Germany.utf8  LC_CTYPE=German_Germany.utf8
## [3] LC_MONETARY=German_Germany.utf8 LC_NUMERIC=C
## [5] LC_TIME=German_Germany.utf8
##
## attached base packages:
## [1] stats     graphics  grDevices datasets  utils     methods   base
##
## other attached packages:
## [1] knitr_1.42       openxlsx_4.2.5.2 here_1.0.1       quanteda_3.2.4
## [5] tidyr_1.3.0      stringr_1.5.0    dplyr_1.1.0
##
## loaded via a namespace (and not attached):
##  [1] zip_2.2.2          Rcpp_1.0.10        pillar_1.8.1
compiler_4.2.2
##  [5] tools_4.2.2        stopwords_2.3      digest_0.6.31
evaluate_0.20
##  [9] lifecycle_1.0.3    tibble_3.2.0       lattice_0.20-45
pkgconfig_2.0.3
## [13] rlang_1.0.6        Matrix_1.5-1       fastmatch_1.1-3    cli_3.6.0
## [17] rstudioapi_0.14    yaml_2.3.7         xfun_0.37
fastmap_1.1.1
## [21] withr_2.5.0        generics_0.1.3     vctrs_0.5.2
rprojroot_2.0.3
## [25] grid_4.2.2         tidyselect_1.2.0   glue_1.6.2         R6_2.5.1
## [29] fansi_1.0.4        rmarkdown_2.20     purrr_1.0.1
magrittr_2.0.3
## [33] htmltools_0.5.4    renv_0.16.0        utf8_1.2.3
stringi_1.7.12
## [37] RcppParallel_5.1.7
```