On the verge of `collapse`

These changes are related to the `collapse` argument for `unnest_tokens()`. What does this argument do? Let's say we have some text in a dataframe, with some metadata attached to each row.

```
library(tidyverse)
library(tidytext)

d <- tibble(
  txt = c(
    "Because I could not stop for Death -",
    "He kindly stopped for me -",
    "The Carriage held but just Ourselves –",
    "And Immortality."
  ),
  meta = c("a", "a", "b", "a")
)

d
```

```
## # A tibble: 4 x 2
##   txt                                    meta
##
## 1 Because I could not stop for Death -   a
## 2 He kindly stopped for me -             a
## 3 The Carriage held but just Ourselves – b
## 4 And Immortality.                       a
```

We can use `unnest_tokens()` to tokenize to words in a pretty straightforward manner.

```
d %>% unnest_tokens(token, txt)
```

```
## # A tibble: 20 x 2
##    meta  token
##
##  1 a     because
##  2 a     i
##  3 a     could
##  4 a     not
##  5 a     stop
##  6 a     for
##  7 a     death
##  8 a     he
##  9 a     kindly
## 10 a     stopped
## 11 a     for
## 12 a     me
## 13 b     the
## 14 b     carriage
```

```
## 15 b     held
## 16 b     but
## 17 b     just
## 18 b     ourselves
## 19 a     and
## 20 a     immortality
```

What should happen if we want to tokenize to something like bigrams (a set of two words), though? Should we include bigrams that cross row boundaries, such as “death he”? The `collapse` argument is intended to control this. Its original implementation was not entirely consistent, though, and sometimes surprised users. The new `collapse` argument can take two kinds of options:

- `NULL`, which means no collapsing across rows
- A character vector of variables to collapse text across

The new behavior also never combines rows that are not adjacent to each other, even if they share a `collapse` variable.

The default is `collapse = NULL`. Notice that bigrams are not created that span across rows (no “death he”).

```
d %>% unnest_tokens(token, txt, token = "ngrams", n = 2) ## default:
collapse = NULL
```

```
## # A tibble: 16 x 2
##    meta  token
##
##  1 a     because i
##  2 a     i could
##  3 a     could not
##  4 a     not stop
##  5 a     stop for
##  6 a     for death
##  7 a     he kindly
##  8 a     kindly stopped
##  9 a     stopped for
## 10 a     for me
## 11 b     the carriage
## 12 b     carriage held
## 13 b     held but
## 14 b     but just
## 15 b     just ourselves
## 16 a     and immortality
```

You can specify collapsing variables. This has only one, but you can use multiple. This approach *does* create a bigram “death he” but does not collapse together the 2nd “a” line and the last one, because they are not adjacent.

```
d %>% unnest_tokens(token, txt, token = "ngrams", n = 2, collapse =
"meta")
```

```
## # A tibble: 17 x 2
```

```
##    meta  token
##
##  1 a     because i
##  2 a     i could
##  3 a     could not
##  4 a     not stop
##  5 a     stop for
##  6 a     for death
##  7 a     death he
##  8 a     he kindly
##  9 a     kindly stopped
## 10 a     stopped for
## 11 a     for me
## 12 b     the carriage
## 13 b     carriage held
## 14 b     held but
## 15 b     but just
## 16 b     just ourselves
## 17 a     and immortality
```

What about grouped data?

Before this recent update, `unnest_tokens()` did not handle grouped data consistently or well. Now, groups are another way to specify which variables should be used collapsing rows.

```
d %>%
  group_by(meta) %>%
  unnest_tokens(token, txt, token = "ngrams", n = 2)
```

```
## # A tibble: 17 x 2
## # Groups:   meta [2]
##    meta  token
##
##  1 a     because i
##  2 a     i could
##  3 a     could not
##  4 a     not stop
##  5 a     stop for
##  6 a     for death
##  7 a     death he
##  8 a     he kindly
##  9 a     kindly stopped
## 10 a     stopped for
## 11 a     for me
## 12 b     the carriage
## 13 b     carriage held
## 14 b     held but
## 15 b     but just
## 16 b     just ourselves
## 17 a     and immortality
```

But you *cannot* use both!

```
d %>%
  group_by(meta) %>%
  unnest_tokens(token, txt, token = "ngrams", n = 2, collapse = "meta")

## Error: Use the `collapse` argument or grouped data, but not both.
```

I've been reluctant to dig into this, because I know it is disruptive to folks to have a breaking change. However, after seeing the new flexibility, there is a lot in favor of moving forward with this more consistent and correct behavior. For example, take a look at the dataset of Jane Austen's six published, completed novels. We have information about line, chapter, and book.

```
library(janeaustenr)

original_books <- austen_books() %>%
  group_by(book) %>%
  mutate(
    linenumber = row_number(),
    chapter = cumsum(str_detect(
      text,
      regex("^chapter [\\divxlc]",
        ignore_case = TRUE
      )
    ))
  ) %>%
  ungroup()

original_books

## # A tibble: 73,422 x 4
##    text                    book                linenumber chapter
##
##  1 "SENSE AND SENSIBILITY" Sense & Sensibility          1       0
##  2 ""                      Sense & Sensibility          2       0
##  3 "by Jane Austen"        Sense & Sensibility          3       0
##  4 ""                      Sense & Sensibility          4       0
##  5 "(1811)"                Sense & Sensibility          5       0
##  6 ""                      Sense & Sensibility          6       0
##  7 ""                      Sense & Sensibility          7       0
##  8 ""                      Sense & Sensibility          8       0
##  9 ""                      Sense & Sensibility          9       0
## 10 "CHAPTER 1"             Sense & Sensibility         10       1
## # … with 73,412 more rows
```

We can tokenize with `collapse = NULL`, which will not combine text across rows across lines. This may be appropriate for some text analysis tasks.

```
original_books %>%
  unnest_tokens(token, text, token = "ngrams", n = 2)

## # A tibble: 675,025 x 4
##    book                linenumber chapter token
##
##  1 Sense & Sensibility          1       0 sense and
```

```
##  2 Sense & Sensibility          1       0 and sensibility
##  3 Sense & Sensibility          2       0
##  4 Sense & Sensibility          3       0 by jane
##  5 Sense & Sensibility          3       0 jane austen
##  6 Sense & Sensibility          4       0
##  7 Sense & Sensibility          5       0
##  8 Sense & Sensibility          6       0
##  9 Sense & Sensibility          7       0
## 10 Sense & Sensibility          8       0
## # … with 675,015 more rows
```

Alternatively, we can tokenize using `collapse = c("book", "chapter")`. Notice that we have **more bigrams** this way, because we have combined text across rows to find more bigrams, but only within chapters. We could have used `group_by(book, chapter)` instead.

```
original_books %>%
  unnest_tokens(token, text,
    token = "ngrams", n = 2,
    collapse = c("book", "chapter")
  )
```

```
## # A tibble: 724,780 x 3
##    book                chapter token
##
##  1 Sense & Sensibility       0 sense and
##  2 Sense & Sensibility       0 and sensibility
##  3 Sense & Sensibility       0 sensibility by
##  4 Sense & Sensibility       0 by jane
##  5 Sense & Sensibility       0 jane austen
##  6 Sense & Sensibility       0 austen 1811
##  7 Sense & Sensibility       1 chapter 1
##  8 Sense & Sensibility       1 1 the
##  9 Sense & Sensibility       1 the family
## 10 Sense & Sensibility       1 family of
## # … with 724,770 more rows
```