

# Data Visualization for 2023 SIGMORPHON-UniMorph

## Contents

<b>Exploratory plots</b>	<b>3</b>
Raw error counts for each algorithm and each split set. . . . .	3
Proportions of errors for each algorithm and each split set. . . . .	4

<b>Proportion when we divide by total number of forms for each split for evaluation.</b>	<b>6</b>
--	----------

<b>Combining test and dev sets</b>	<b>8</b>
------------------------------------	----------

Set figures write path and formats, set caching as default

```
knitr::opts_chunk$set(fig.path = "figs/fra-", dev = c('pdf', 'png'), cache=T)
```

Load libraries

```
library(tidyr)
library(ggplot2)
library(magrittr)
```

```
##
## Attaching package: 'magrittr'

## The following object is masked from 'package:tidyr':
##
##   extract
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
library(stringr)
```

Read in data files.

```
dat.forms <- read.csv("data/fra_data_set_form_counts.csv")
dat.errors <- read.csv("data/fra_error_distribution.csv")
```

Make data set “long” form, with a column for error type and change error type to factor and set level ordering

```
dat.err.long <- dat.errors %>%
  pivot_longer(!Data.set, names_to = "error_type", values_to = "count") %>%
  mutate(error_type = factor(error_type, levels = rev(c("Old.lemma", "Old.Rule", "Free.Var", "Over.Reg")))
```

Add in more information about the data sets based on their abbreviated name.

```
dat.err.long %<>%
  mutate(
    algorithm = if_else(str_starts(Data.set, "ru"), "rule", "neural"),
    split = if_else(str_ends(Data.set, "d"), "dev", "test"),
    split_set = case_when(
      str_detect(Data.set, "-s-") ~ "Seg (by lemma)",
      str_detect(Data.set, "-m-") ~ "Seg-Minimal (by form)",
      str_detect(Data.set, "-o-") ~ "Original Split"
    )
  )

# Mutate to factors

dat.err.long %<>%
  mutate(
    algorithm = factor(algorithm, levels = c("rule", "neural"))
  )
```

Also compute proportion of total number of errors that is some particular error type.

```
tot.errors.by.data.set <- dat.err.long %>%
  group_by(Data.set) %>%
  summarize(tot_error_data_set = sum(count))

dat.err.long %<>%
  left_join(tot.errors.by.data.set, by = "Data.set") %>%
  mutate(prop_out_of_errors_for_data_set = count/tot_error_data_set)
```

Finally, add in total number of forms in dev/test set to scale number of errors on dev/test set.

First, we need to convert data frame for dat.forms to long form.

```
dat.forms.long <- dat.forms %>%
  mutate(split = case_when(Data.set == "fra.trn" ~ "trn",
                           Data.set == "fra.dev" ~ "dev",
                           Data.set == "fra.tst" ~ "test")
  ) %>%
  pivot_longer(cols = c("Original", "Segmentations", "Segmentations.Minimal"),
               names_to = "split_set_raw",
               values_to = "split_size"
  )

# Let's also rename split_set names to be consistent with dat.err.long

dat.forms.long %<>%
  mutate(split_set = case_when(split_set_raw == "Original" ~ "Original Split",
                               split_set_raw == "Segmentations" ~ "Seg (by lemma)",
                               split_set_raw == "Segmentations.Minimal" ~ "Seg-Minimal (by form)"
  )
  ) %>%
  dplyr::select(split_set, split, split_size)
```

OK, now we can join the two data frames to compute percent of error by total forms in test/dev sets.

```
dat.err.long %<>%
  left_join(dat.forms.long, by = c("split", "split_set")) %>%
  mutate(prop_forms_in_eval_split = count/split_size)
```

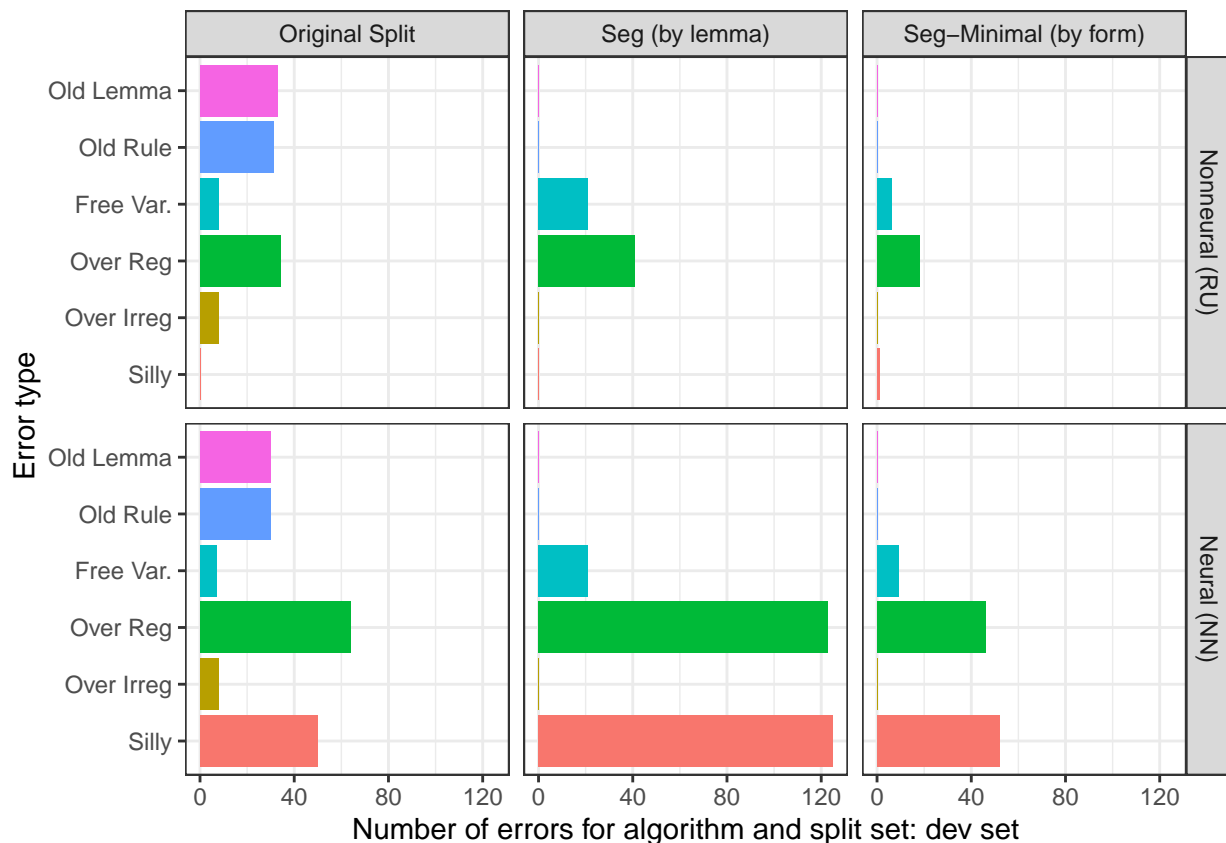
## Exploratory plots

Raw error counts for each algorithm and each split set.

Dev set:

```
new_labels_algo <- c("rule" = "Nonneural (RU)", "neural" = "Neural (NN)")
new_labels_errors <- c("Old.lemma" = "Old Lemma", "Old.Rule" = "Old Rule", "Free.Var" = "Free Var.", "Over.Reg" = "Over Reg", "Over.Irreg" = "Over Irreg", "Silly" = "Silly")
```

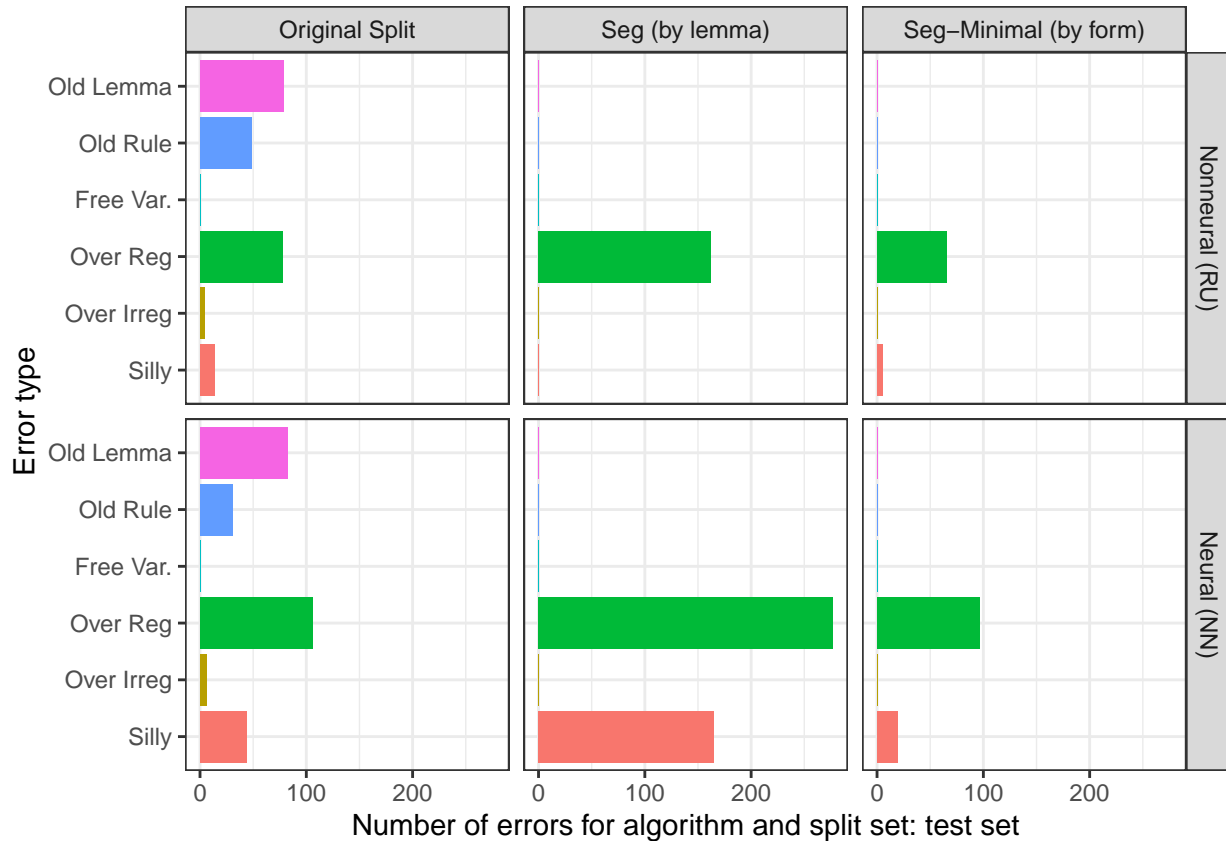
```
dat.err.long %>%
  filter(split == "dev") %>%
  ggplot(aes(y = count, x = error_type, fill = error_type)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_x_discrete(name = "Error type", labels = new_labels_errors) +
  scale_fill_discrete(type = "Okabe-Ito") +
  scale_y_continuous(name = "Number of errors for algorithm and split set: dev set") +
  facet_grid(algorithm~split_set, labeller = labeller(algorithm = new_labels_algo)) + coord_flip() + gu
```



Test set:

```
dat.err.long %>%
  filter(split == "test") %>%
  ggplot(aes(y = count, x = error_type, fill = error_type)) +
```

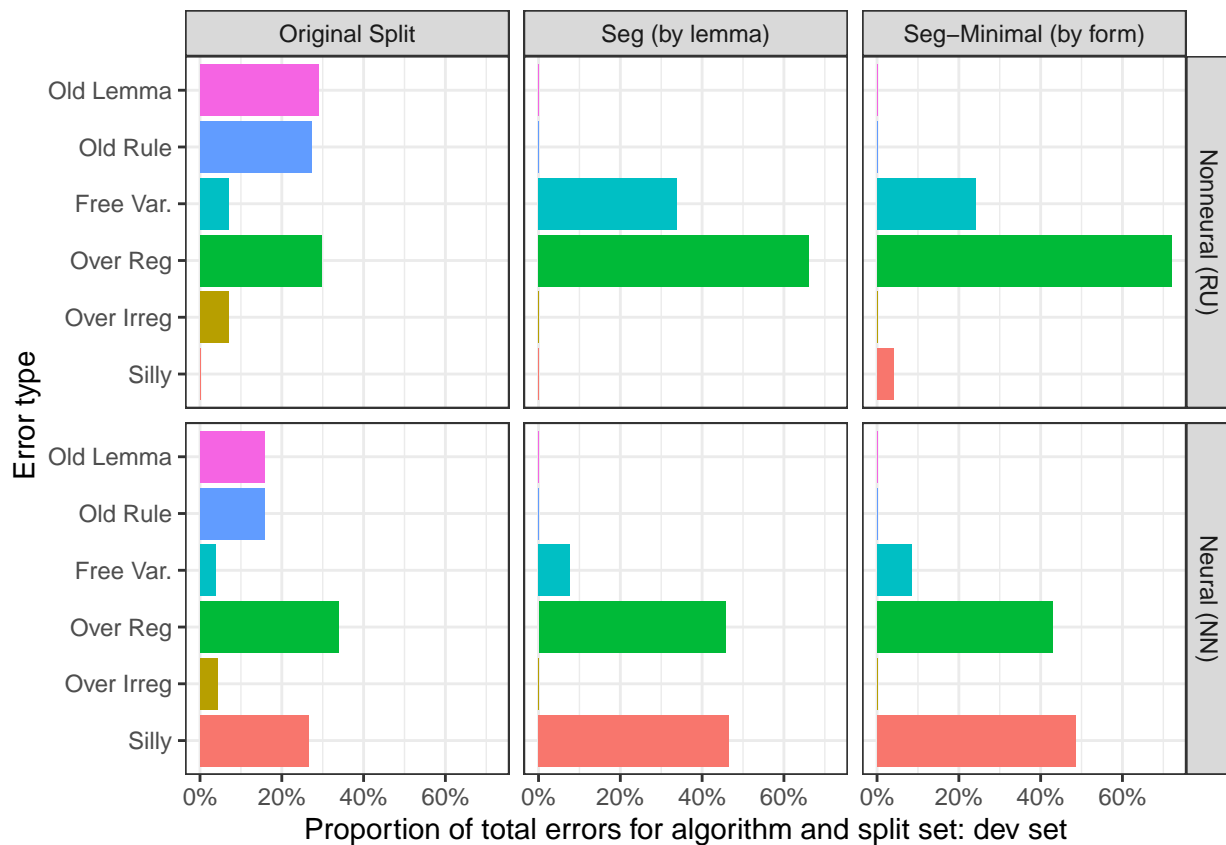
```
geom_bar(stat = "identity", position = "dodge") +
scale_x_discrete(name = "Error type", labels = new_labels_errors) +
scale_fill_discrete(type = "Okabe-Ito") +
scale_y_continuous(name = "Number of errors for algorithm and split set: test set") +
facet_grid(algorithm~split_set, labeller = labeller(algorithm = new_labels_algo)) + coord_flip() + gu
```



Proportions of errors for each algorithm and each split set.

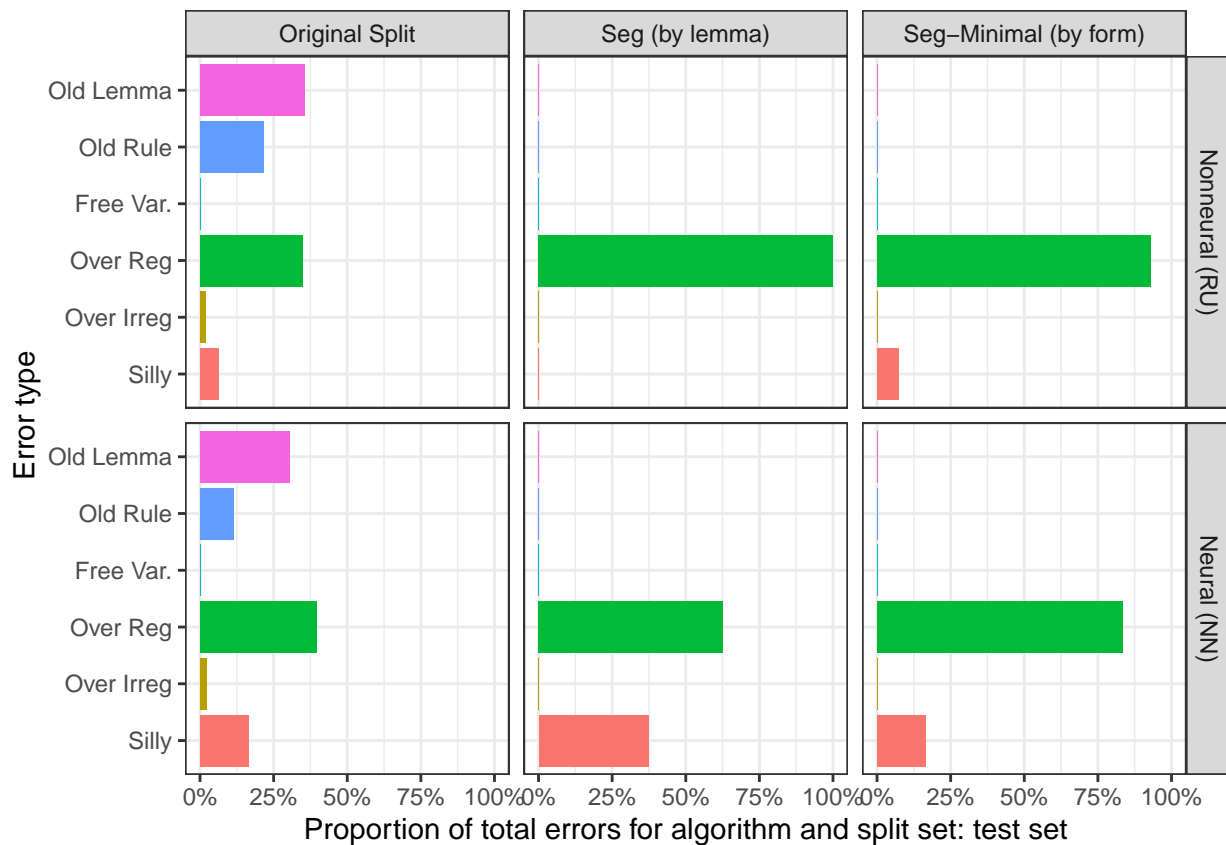
Dev set

```
dat.err.long %>%
  filter(split == "dev") %>%
  ggplot(aes(y = prop_out_of_errors_for_data_set, x = error_type, fill = error_type)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_x_discrete(name = "Error type", labels = new_labels_errors) +
  scale_fill_discrete(type = "Okabe-Ito") +
  scale_y_continuous(name = "Proportion of total errors for algorithm and split set: dev set", labels =
  facet_grid(algorithm~split_set, labeller = labeller(algorithm = new_labels_algo)) + coord_flip() + gu
```



Test set

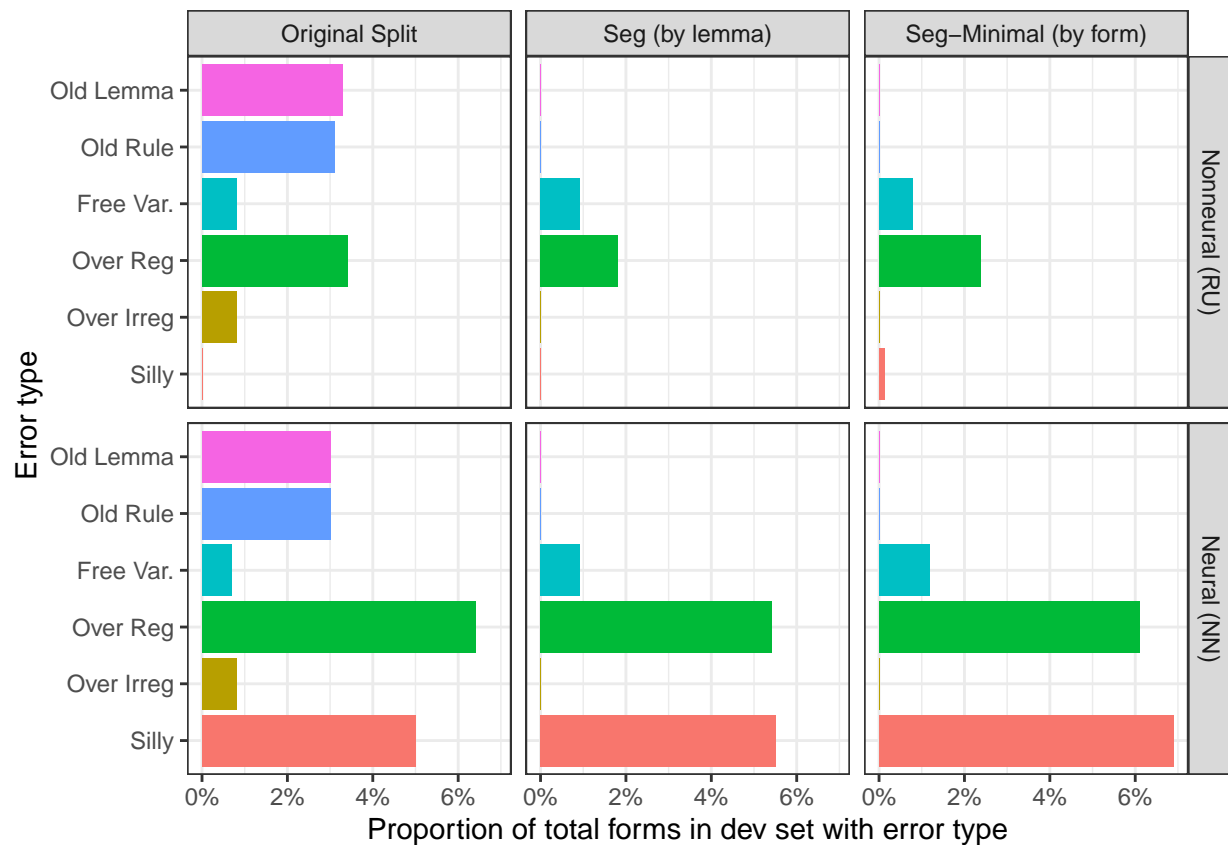
```
dat.err.long %>%
  filter(split == "test") %>%
  ggplot(aes(y = prop_out_of_errors_for_data_set, x = error_type, fill = error_type)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_x_discrete(name = "Error type", labels = new_labels_errors) +
  scale_fill_discrete(type = "Okabe-Ito") +
  scale_y_continuous(name = "Proportion of total errors for algorithm and split set: test set", labels = new_labels_errors) +
  facet_grid(algorithm~split_set, labeller = labeller(algorithm = new_labels_algo)) + coord_flip() + guides(fill = "none")
```



Proportion when we divide by total number of forms for each split for evaluation.

Dev

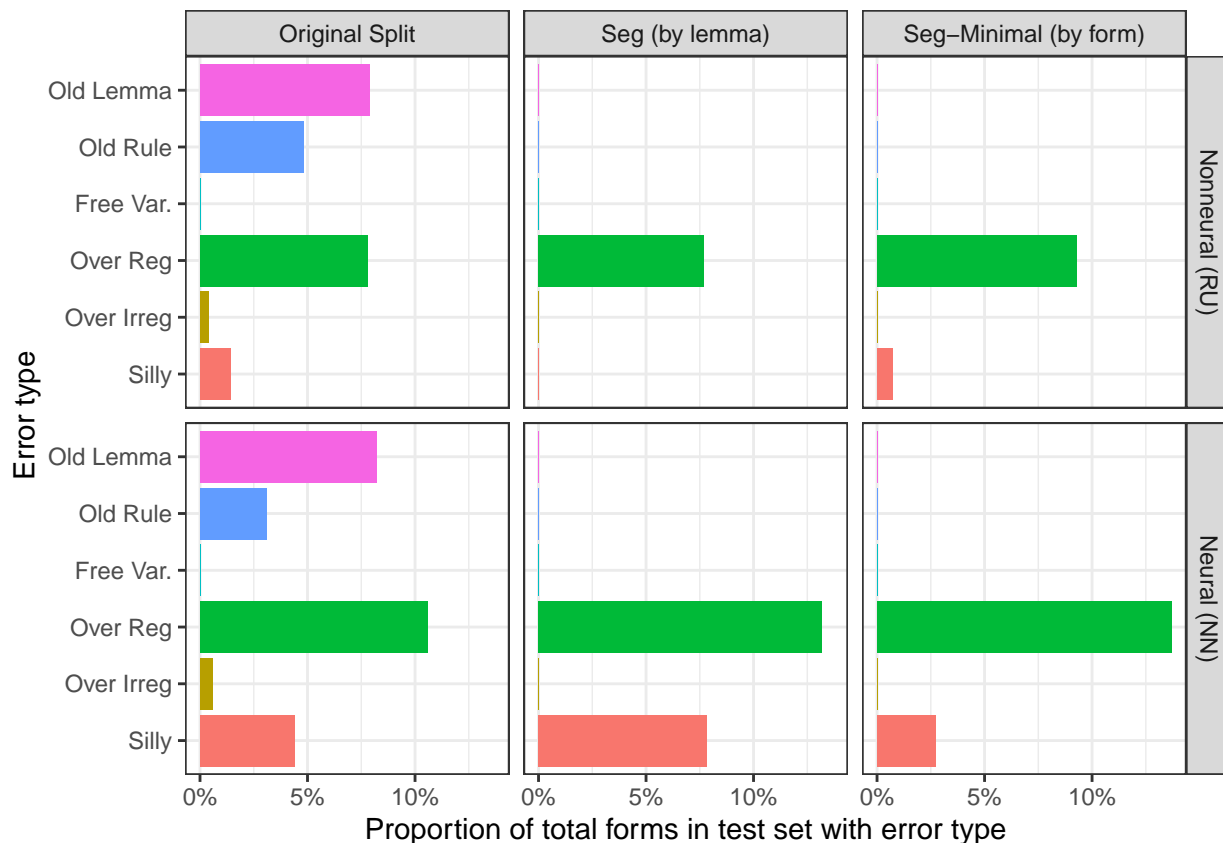
```
dat.err.long %>%
  filter(split == "dev") %>%
  ggplot(aes(y = prop_forms_in_eval_split, x = error_type, fill = error_type)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_x_discrete(name = "Error type", labels = new_labels_errors) +
  scale_fill_discrete(type = "Okabe-Ito") +
  scale_y_continuous(name = "Proportion of total forms in dev set with error type", labels = scales::percent) +
  facet_grid(algorithm~split_set, labeller = labeller(algorithm = new_labels_algo)) + coord_flip() + guides(fill = "error_type")
```



```

dat.err.long %>%
  filter(split == "test") %>%
  ggplot(aes(y = prop_forms_in_eval_split, x = error_type, fill = error_type)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_x_discrete(name = "Error type", labels = new_labels_errors) +
  scale_fill_discrete(type = "Okabe-Ito") +
  scale_y_continuous(name = "Proportion of total forms in test set with error type", labels = scales::p_c(0, 2, 4, 6)) +
  facet_grid(algorithm~split_set, labeller = labeller(algorithm = new_labels_algo)) + coord_flip() + guides(fill = "error_type")

```



## Combining test and dev sets

First, combine test and dev set split size calculations.

```
dat.forms.long %>%
  filter(split != "tn") %>%
  group_by(split_set) %>%
  summarize(eval_split_size = sum(split_size)) -> dat.forms.long.combo
```

Now combine error counts. For now let's just compute breakdown of error types among errors.

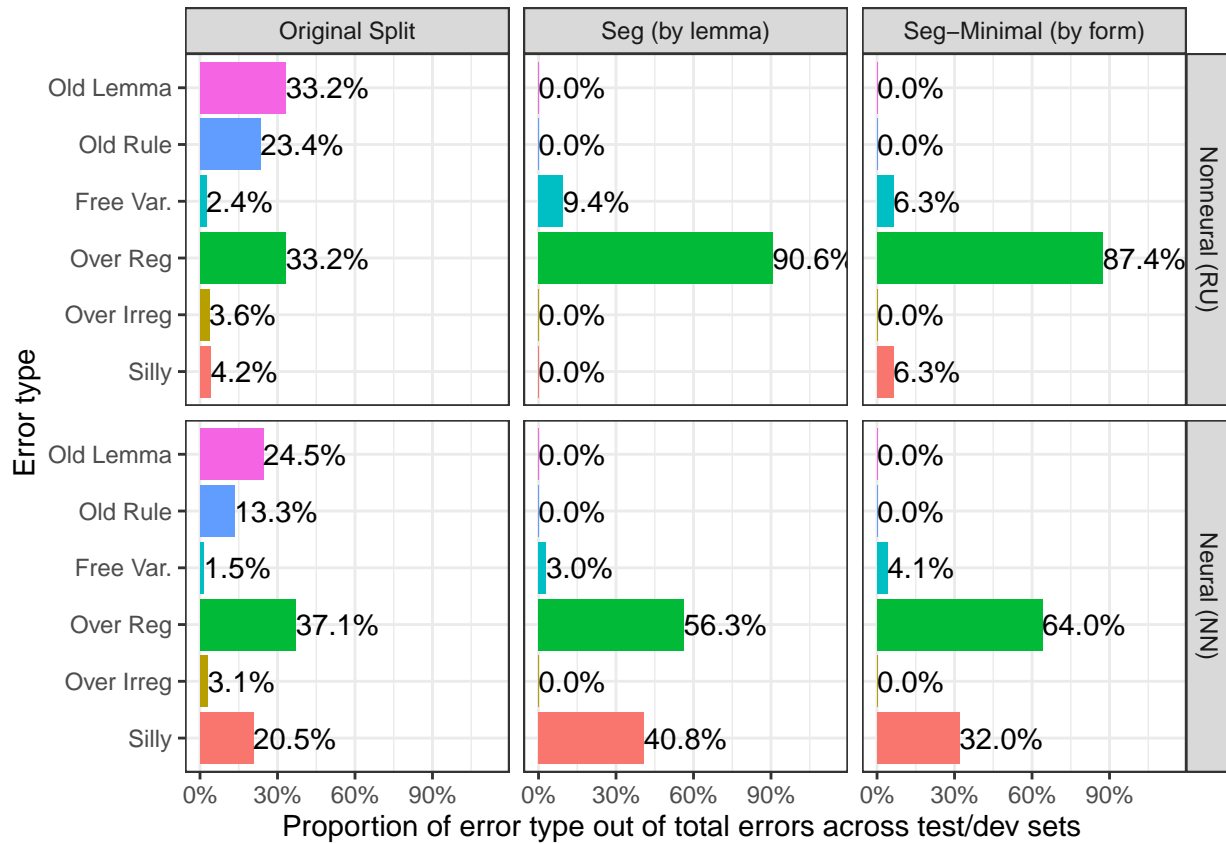
```
dat.err.long %>%
  group_by(split_set, algorithm, error_type) %>%
  summarize(eval_count = sum(count),
            tot_error_split_set = sum(tot_error_data_set)
            ) %>%
  mutate(prop_out_of_errors = eval_count/tot_error_split_set) -> dat.err.long.combo
```

## `summarise()` has grouped output by 'split\_set', 'algorithm'. You can override  
## using the `.groups` argument.

```
dat.err.long.combo %>%
  ggplot(aes(y = prop_out_of_errors, x = error_type, fill = error_type)) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_text(aes(label=scales::percent(prop_out_of_errors, accuracy = 0.1)), hjust = 0)+
  scale_x_discrete(name = "Error type", labels = new_labels_errors) +
  scale_fill_discrete(type = "Okabe-Ito") +
  scale_y_continuous(name = "Proportion of error type out of total errors across test/dev sets ", label=
```



```
facet_grid(algorithm~split_set, labeller = labeller(algorithm = new_labels_algo)) + coord_flip() + gu
```



```
dat.err.long.combo %>%
  ggplot(aes(y = prop_out_of_errors, x = error_type, fill = error_type)) +
  geom_bar(stat = "identity", position = "dodge") +
  # geom_text(aes(label=scales::percent(prop_out_of_errors, accuracy = 0.1)), hjust = -0.06)+
  scale_x_discrete(name = "Error type", labels = new_labels_errors) +
  scale_fill_discrete(type = "Okabe-Ito") +
  scale_y_continuous(name = "Proportion of error type out of total errors across test/dev sets ", labels = new_labels_errors) +
  facet_grid(algorithm~split_set, labeller = labeller(algorithm = new_labels_algo)) + coord_flip() + gu
```

