There are two splits of the original data, and one split of the challenge data.

Students with the following initials: AR, EM, HL, and JA should use Lillian’s split.

Students with the following initials: JC, JL, VS should use Tianze’s split.

File listing:

**Lillian’s split:** the way you can tell you have the new (as of Oct 28) split: there should NOT be the ID 473 twice in original-dev.ids.a.txt. (Thanks very much Hannah for noticing this!!!)

sentences.tsv  
original-dev.ids.a.txt (redid on Oct 28)  
original-test.ids.a.txt (redid on Oct 28)

**Tianze’s split:**

original.train.ids  
original.dev.ids  
original.test.ids

**Challenge split** (see notes at the end of this page)

challenge.tsv  
challenge.train.id-prefixes.txt

Lillian’s split of the original data was created as follows.

% cat sentences.tsv | awk '{if ($(NF-1)="+1") print $0}' | perl -MList::Util=shuffle -e 'print shuffle(<STDIN>);' | head -e echo "dummy" | awk '{print int(2*3610)}' > original20percent.pos.a.txt  
% cat sentences.tsv | awk '{if ($(NF-1)="-1") print $0}' | perl -MList::Util=shuffle -e 'print shuffle(<STDIN>);' | head -e echo "dummy" | awk '{print int(2*3310)}' > original20percent.neg.a.txt  
% tail -331 original20percent.neg.a.txt | awk '{print $1}' > original-dev.ids.a.txt  
% tail -361 original20percent.pos.a.txt | awk '{print $1}' >> original-dev.ids.a.txt  
% head -331 original20percent.neg.a.txt | awk '{print $1}' > original-test.ids.a.txt  
% head -361 original20percent.pos.a.txt | awk '{print $1}' >> original-test.ids.a.txt

Thus, you have files that specify the sentence ids for the sentences belonging to the development and test set, respectively; the training set consists of the sentence IDs that aren't in either original-dev.ids.a.txt or original-test.ids.a.txt.

#### Sanity checks

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current Version (v. 5 Oct 29, 2019 13:37)</td>
<td>Lillian Lee: Confirmed that there can be repeated IDs, but should not be repeated sentences, in the challenge data</td>
<td></td>
</tr>
<tr>
<td>v. 4 Oct 28, 2019 12:28</td>
<td>Lillian Lee: added sanity check commands</td>
<td></td>
</tr>
<tr>
<td>v. 3 Oct 28, 2019 12:21</td>
<td>Lillian Lee: Utterly embarrassing: I had ‘=’ instead of ‘==’ in my command-line stuff, causing every item to be assigned label +1 (say), instead of looking for items that had +1.</td>
<td></td>
</tr>
<tr>
<td>v. 2 Oct 25, 2019 13:16</td>
<td>Lillian Lee: add explicit diff tracking</td>
<td></td>
</tr>
<tr>
<td>v. 1 Oct 25, 2019 13:13</td>
<td>Lillian Lee</td>
<td></td>
</tr>
</tbody>
</table>

Here is a link to a page where you can view “diffs” between any two versions: use the “compare selected versions” feature to highlight precisely what text was added or deleted.
Tianze’s split of the original data was created as follows.

```
% tail -n +2 sentences.tsv.txt | cut -f 1,3 | shuf | grep ”+1” | cut -f 1 > original.pos.ids
% tail -n +2 sentences.tsv.txt | cut -f 1,3 | shuf | grep ”-1” | cut -f 1 > original.neg.ids
% sed -n ’1,’`expr 361 \* 8`’ p’ original.pos.ids > original.pos.train.ids
% sed -n ’expre 361 \* 8 + 1’,’expr 361 \* 9’ p’ original.pos.ids > original.pos.dev.ids
% sed -n ’expre 361 \* 9 + 1’,’expr 361 \* 10’ p’ original.pos.ids > original.pos.test.ids
% sed -n ’1,’`expr 331 \* 8`’ p’ original.neg.ids > original.neg.train.ids
% sed -n ’expre 331 \* 8 + 1’,’expr 331 \* 9’ p’ original.neg.ids > original.neg.dev.ids
% sed -n ’expre 331 \* 9 + 1’,’expr 331 \* 10’ p’ original.neg.ids > original.neg.test.ids
% for split in train dev test; do (cat original.pos.${split}.ids original.neg.${split}.ids > original.${split}.ids) done
```

#### Sanity check after generation:

```
% cat original.train.ids original.dev.ids original.test.ids | sort | uniq | wc -l
% cat original.train.ids original.dev.ids original.test.ids | sort | uniq | wc -l
```

#### Both gave 6920.

The challenge data split is as follows. This is not what we talked about in class, due to some imbalance in Team4_breaker_test.tsv and the fact that 10% of the data being training could be too small to allow interesting variation in fine-tuning-set size.

```
% cat Team{1,2,3}_breaker_test.tsv
# Then some manual editing (including removing:
# 673_a This quirky, snarky contemporary fairy tale could have been a family blockbuster. -1
# 673_a This quirky, snarky contemporary fairy tale could have been a family blockbuster. 1
# )
# to yield challenge.tsv
```

```
% cut -f1 challenge.tsv | cut -f1 -d”_” | sort | uniq | perl -MList::Util=shuffle -e ”print shuffle(<STDIN>);” | head -50 > challenge.train.id-prefixes.txt
```
The first entry in challenge.train.id-prefixes.txt is "850", so, the following two sentences from challenge.tsv should be in the small challenge training set:

850_a It's basically the videogame version of Top Gun... on steroids!
1
850_b It's basically the videogame version of Top Gun... -1

Note that there may be "repeated" IDs, as posted about in CampusWire:

Q:

duplicate indices in challenge.tsv

A: This was a design choice, but good to check!
Note that the actual sentences for the two 559_b's are different, although both are "challenges" to the same 559_a. So you will want all three 559s to be in the same split, counting as three different examples as a design choice, but good to check!
In general, there could be as many as 3 x_b's, one per each of the three breaker teams' data.