CS 6740/IS 6300 A3 data readme

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 $0}' | perl -MList::Util=shuffle -e 'print shuffle(<STDIN>);' | head -`echo "dummy" | awk '{print int(.2*3610)}'` > original20percent.pos.a.txt
% cat sentences.tsv | awk '{if ($(NF-1)= "-1") print $0}' | perl -MList::Util=shuffle -e 'print shuffle(<STDIN>);' | head -`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

```
% cat original-dev.ids.a.txt original-test.ids.a.txt | sort | uniq -c | sort -nr | head

1 9993   # so, nothing appears in twice in the concatenation of the "a" files.
```

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.

Version	Published	Changed By	Comment
CURRENT (v. 5)	**Oct 29, 2019 13:37**	**Lillian Lee**	Confirmed that the
v. 4	Oct 28, 2019 12:28	**Lillian Lee**	added sanity chec
v. 3	Oct 28, 2019 12:21	**Lillian Lee**	Utterly embarrassi
v. 2	Oct 25, 2019 13:16	**Lillian Lee**	add explicit diff tra
v. 1	Oct 25, 2019 13:13	**Lillian Lee**	

```
% wc -l *ids.a.txt

692 original-dev.ids.a.txt
692 original-test.ids.a.txt
1384 total
```

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 `expr 361 \* 8 + 1`','`expr 361 \* 9`' p' original.pos.ids > original.pos.dev.ids
% sed -n `expr 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 `expr 331 \* 8 + 1`','`expr 331 \* 9`' p' original.neg.ids > original.neg.dev.ids
% sed -n `expr 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 | 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*
>
> *#17*
> *I noticed that there are duplicate indices in* challenge.tsv*. For one example, there are two instances of 559_b's from* challenge.tsv*:*
>
> *559_a Unfolds with the creepy elegance and carefully calibrated precision of a Dario Argento horror film. 1 559_b Unfolds with all the creepy elegance and carefully calibrated precision of a Jim Carrey comedy film. -1 559_b Unfolds with the creepy elegance and carefully calibrated precision of a Uwe Boll horror film. -1*
>
> *I am not sure if this was intentional, or the third 559 example was meant to be encoded as something like 559_c. I first assumed there would only be pairs (a and b) of similar sentences in the challenge dataset, but the above examples show that there can be either pairs or trios of them.*
>
> ***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! 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.*
> *In general, there could be as many as 3 x_b's, one per each of the three breaker teams' data.*