In [ ]:
```
from fastai.text import *   # Quick access to NLP functionality
```

Text example

An example of creating a language model and then transfering to a classifier.

In [ ]:
```
path = untar_data(URLs.IMDB_SAMPLE)
path
```

Out[ ]:
```
PosixPath('/home/ubuntu/.fastai/data/imdb_sample')
```

Open and view the independent and dependent variables:

In [ ]:
```
df = pd.read_csv(path/'texts.csv', header=None)
df.head()
```

Out[ ]:
```
          0                                                1         2
0     label                                             text  is_valid
1  negative  Un-bleeping-believable! Meg Ryan doesn't even ...     False
2  positive       This is a extremely well-made film. The acting...     False
3  negative      Every once in a long while a movie will come a...     False
4  positive        Name just says it all. I watched this movie wi...     False
```

Create a `DataBunch` for each of the language model and the classifier:

In [ ]:
```
data_lm = TextLMDataBunch.from_csv(path, 'texts.csv')
data_clas = TextClasDataBunch.from_csv(path, 'texts.csv', vocab=data_lm.train_ds.vo
```

We'll fine-tune the language model. fast.ai has a pre-trained English model available that we can download, we just have to specify it like this:

In [ ]:
```
moms = (0.8,0.7)
```

In [ ]:
```
learn = language_model_learner(data_lm, pretrained_model=URLs.WT103_1)
learn.unfreeze()
learn.fit_one_cycle(4, slice(1e-2), moms=moms)
```

Total time: 00:17

epoch	train_loss	valid_loss	accuracy
1	4.639660	3.914269	0.293896
2	4.283420	3.723600	0.302778
3	4.032526	3.689489	0.304384
4	3.857930	3.681090	0.304303

Save our language model's encoder:

```
In [ ]: learn.save_encoder('enc')
```

Fine tune it to create a classifier:

```
In [ ]: learn = text_classifier_learner(data_clas)
        learn.load_encoder('enc')
        learn.freeze()
        learn.fit_one_cycle(4, moms=moms)
```

Total time: 00:22

epoch	train_loss	valid_loss	accuracy
1	0.668317	0.604398	0.716418
2	0.643791	0.572027	0.701493
3	0.622935	0.562883	0.686567
4	0.614669	0.529685	0.736318

```
In [ ]: learn.unfreeze()
        learn.fit_one_cycle(8, slice(1e-5,1e-3), moms=moms)
```

Total time: 01:32

epoch	train_loss	valid_loss	accuracy
1	0.588901	0.545256	0.711443
2	0.608616	0.490764	0.781095
3	0.598989	0.572883	0.701493
4	0.570460	0.485850	0.776119
5	0.548549	0.505190	0.761194
6	0.562036	0.488297	0.771144
7	0.545467	0.481813	0.805970
8	0.547870	0.491384	0.766169