Data Preprocessing Module

Use `data_preprocess(erc_raw_data)` to preprocess the given data into csv files (no return).

In [ ]:
```
def data_preprocess(ercdata_raw, ercdata_raw_null,
                    root_path=erc_root_folder,
                    training_portion=0.85,
                    validation_portion=0.05,
                    testing_portion=0.10):
    root_path = root_path.rstrip('/')

    # make sure that they are summing up to 1.
    assert training_portion + validation_portion + testing_portion == 1.0

    # open up writers.
    dpre_train_writer = open(root_path + '/dpre_training_data.csv', 'w')
    dpre_valid_writer = open(root_path + '/dpre_validation_data.csv', 'w')
    dpre_test_writer = open(root_path + '/dpre_testing_data.csv', 'w')

    dpre_train_csv_writer = csv.writer(dpre_train_writer, delimiter=',')
    dpre_valid_csv_writer = csv.writer(dpre_valid_writer, delimiter=',')
    dpre_test_csv_writer = csv.writer(dpre_test_writer, delimiter=',')

    # write the header.
    dpre_header = ['text', 'label']
    dpre_train_csv_writer.writerow(dpre_header)
    dpre_valid_csv_writer.writerow(dpre_header)
    dpre_test_csv_writer.writerow(dpre_header)

    conflicts_count = 0

    processed_rows = []

    for i in range(len(ercdata_raw)):
        # detect if it is conflict.
        curr_label = ercdata_raw['finalized_label(is_referring)'].iloc[i]
        if curr_label == '[conflict occurring!]': # conflict label processed by annotat
            conflicts_count += 1
            continue

        # detect if it is null.
        curr_isnull = ercdata_raw_null['finalized_label(is_referring)'].iloc[i]
        if curr_isnull:
            continue

        # extract the data if it passes all validation tests.
        curr_data = normalize_v2(ercdata_raw['text'].iloc[i].lower(), ercdata_raw['targ
        curr_label = 1 if curr_label == 'True' else 0

        # buildup the data row.
        processed_rows.append([curr_data, curr_label])

    # shuffle the raw data.
    random.shuffle(processed_rows)

    len_rows = len(processed_rows)
    dpre_train_size = int(len_rows * training_portion)
    dpre_valid_size = dpre_train_size + int(len_rows * validation_portion)
    dpre_test_size = dpre_valid_size + int(len_rows * testing_portion)
```