Sarcasm Detection

In [1]:
```
import tensorflow as tf
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

Dataset

Acknowledgement

Misra, Rishabh, and Prahal Arora. "Sarcasm Detection using Hybrid Neural Network." arXiv preprint arXiv:1908.07414 (2019).

Loading the Data

In [2]:
```
import pandas as pd
data = pd.read_json("/content/drive/My Drive/Colab Notebooks/Sarcasm_Headlines_Dataset.json",lines=True)
```

Dropping `article_link` from dataset, since it is not significant

In [3]:
```
del data['article_link']
```

Got length of each headline and add a column for the same.

In [4]:
```
data['column-length'] = data['headline'].apply(lambda x: len(x))
```

In [5]:
```
data['headline'][10]
```

Out[5]:
```
'airline passengers tackle man who rushes cockpit in bomb threat'
```

In [6]:
```
data.head(10)
```

Out[6]:

	headline	is_sarcastic	column-length
0	former versace store clerk sues over secret 'b...	0	78
1	the 'roseanne' revival catches up to our thorn...	0	84
2	mom starting to fear son's web series closest ...	1	79
3	boehner just wants wife to listen, not come up...	1	84
4	j.k. rowling wishes snape happy birthday in th...	0	64
5	advancing the world's women	0	27
6	the fascinating case for eating lab-grown meat	0	46
7	this ceo will send your kids to school, if you...	0	67
8	top snake handler leaves sinking huckabee camp...	1	50
9	friday's morning email: inside trump's presser...	0	59

Initializing parameter values

- Set values for max_features, maxlen, & embedding_size
- max_features: Number of words to take from tokenizer(most frequent words)
- maxlen: Maximum length of each sentence is limited to 25
- embedding_size: size of embedding vector

In [7]:
```
max_features = 10000
maxlen = 25
embedding_size = 200
```

Applied `tensorflow.keras` Tokenizer and get indices for words

- Initialized Tokenizer object with number of words as 10000
- Fit the tokenizer object on headline column
- Converted the text to sequence

In [8]:
```
tokenizer = Tokenizer() #create tokenizer object
tokenizer.fit_on_texts(data['headline']) #create word index dict

word_index_dict = tokenizer.word_index #get word index dict
vocab_size = len(word_index_dict) + 1

vocab_size
```

Out[8]: 29657

Pad sequences

- Pad each example with a maximum length
- Convert target column into numpy array

In [9]:
```
sequence = tokenizer.texts_to_sequences(data['headline'])

padding = pad_sequences(sequences=sequence, maxlen=maxlen)
```

In [10]:
```
training_size = len(data)
test_portion = 0.3
split = int(test_portion*training_size)
split
test_sequences = padding[0:split]
training_sequences = padding[split:]

labels = data['is_sarcastic']

test_labels = labels[0:split]
training_labels = labels[split:]
```

In [11]:
```
print(test_labels.shape)
print(training_labels.shape)
```

```
(8012,)
(18697,)
```

Vocab mapping

- There is no word for 0th index

In [12]:
```
tokenizer.word_index
```

Out[12]:
```
{'to': 1,
 'of': 2,
 'the': 3,
 'in': 4,
 'for': 5,
 'a': 6,
 'on': 7,
 'and': 8,
 'with': 9,
 'is': 10,
 'new': 11,
 'trump': 12,
 'man': 13,
 'from': 14,
 'at': 15,
 'about': 16,
 'you': 17,
 'this': 18,
 'by': 19,
 'after': 20,
 'up': 21,
 'out': 22,
 'be': 23,
 'how': 24,
 'as': 25,
 'it': 26,
 'that': 27,
 'not': 28,
 'are': 29,
 'your': 30,
 'his': 31,
 'what': 32,
 'he': 33,
 'all': 34,
 'just': 35,
 'who': 36,
 'has': 37,
 'will': 38,
 'more': 39,
 'one': 40,
 'into': 41,
```

```
'report': 42,
'year': 43,
'why': 44,
'have': 45,
'area': 46,
'over': 47,
'donald': 48,
'u': 49,
'day': 50,
'says': 51,
's': 52,
'can': 53,
'first': 54,
'woman': 55,
'time': 56,
'like': 57,
'her': 58,
"trump's": 59,
'old': 60,
'no': 61,
'get': 62,
'off': 63,
'an': 64,
'life': 65,
'people': 66,
'obama': 67,
'now': 68,
'house': 69,
'still': 70,
"'": 71,
'women': 72,
'make': 73,
'was': 74,
'than': 75,
'white': 76,
'back': 77,
'my': 78,
'i': 79,
'clinton': 80,
'down': 81,
'if': 82,
'5': 83,
```

```
'when': 84,
'world': 85,
'could': 86,
'we': 87,
'their': 88,
'before': 89,
'americans': 90,
'way': 91,
'do': 92,
'family': 93,
'most': 94,
'gop': 95,
'they': 96,
'study': 97,
'school': 98,
"it's": 99,
'black': 100,
'best': 101,
'years': 102,
'bill': 103,
'should': 104,
'3': 105,
'him': 106,
'would': 107,
'so': 108,
'police': 109,
'only': 110,
'watch': 111,
'american': 112,
'really': 113,
'being': 114,
'but': 115,
'last': 116,
'know': 117,
'10': 118,
"can't": 119,
'death': 120,
'home': 121,
'during': 122,
'video': 123,
'finds': 124,
'state': 125,
```

```
'or': 126,
'president': 127,
'health': 128,
'going': 129,
'say': 130,
'show': 131,
'nation': 132,
'good': 133,
'things': 134,
'hillary': 135,
"'the": 136,
'may': 137,
'2': 138,
'against': 139,
'campaign': 140,
'every': 141,
'she': 142,
'love': 143,
'mom': 144,
'need': 145,
'big': 146,
'right': 147,
'party': 148,
'gets': 149,
'000': 150,
'too': 151,
'getting': 152,
'these': 153,
'kids': 154,
'some': 155,
'parents': 156,
'work': 157,
'court': 158,
'little': 159,
'change': 160,
'take': 161,
'high': 162,
'makes': 163,
'self': 164,
'our': 165,
'calls': 166,
'john': 167,
```

```
'other': 168,
'news': 169,
'through': 170,
"doesn't": 171,
'while': 172,
"here's": 173,
'never': 174,
'child': 175,
'gay': 176,
'dead': 177,
'look': 178,
'election': 179,
'want': 180,
'own': 181,
'4': 182,
"don't": 183,
'see': 184,
'takes': 185,
'america': 186,
'7': 187,
'local': 188,
'real': 189,
'where': 190,
'next': 191,
'stop': 192,
'even': 193,
'its': 194,
"he's": 195,
'war': 196,
'college': 197,
'go': 198,
'6': 199,
"nation's": 200,
'sex': 201,
'bush': 202,
'made': 203,
'plan': 204,
'office': 205,
'again': 206,
'guy': 207,
'two': 208,
'dad': 209,
```

```
'another': 210,
'around': 211,
'dog': 212,
'got': 213,
'1': 214,
'million': 215,
'ever': 216,
'week': 217,
'baby': 218,
'debate': 219,
'thing': 220,
'them': 221,
'gun': 222,
'wants': 223,
'care': 224,
'us': 225,
'help': 226,
'much': 227,
'long': 228,
'night': 229,
'congress': 230,
'job': 231,
'finally': 232,
'north': 233,
'been': 234,
'under': 235,
"man's": 236,
'actually': 237,
'star': 238,
'national': 239,
'live': 240,
'climate': 241,
'season': 242,
'money': 243,
'couple': 244,
"won't": 245,
'8': 246,
'9': 247,
'top': 248,
'god': 249,
'anti': 250,
'media': 251,
```

```
'food': 252,
'ways': 253,
'20': 254,
'shows': 255,
'sexual': 256,
'better': 257,
'give': 258,
'shooting': 259,
'had': 260,
'teen': 261,
'face': 262,
'making': 263,
'game': 264,
'paul': 265,
'reveals': 266,
'me': 267,
'trying': 268,
'senate': 269,
'supreme': 270,
'announces': 271,
'there': 272,
'away': 273,
'men': 274,
'history': 275,
'business': 276,
'bad': 277,
'without': 278,
'students': 279,
'everyone': 280,
'attack': 281,
'end': 282,
'story': 283,
'fight': 284,
'facebook': 285,
'son': 286,
'free': 287,
'children': 288,
'enough': 289,
'tv': 290,
'law': 291,
'movie': 292,
'city': 293,
```

```
'any': 294,
'introduces': 295,
'pope': 296,
'deal': 297,
'government': 298,
'body': 299,
'part': 300,
'york': 301,
'11': 302,
'tell': 303,
'great': 304,
'film': 305,
'does': 306,
'former': 307,
'single': 308,
'entire': 309,
'friends': 310,
'fire': 311,
'call': 312,
'found': 313,
'friend': 314,
'book': 315,
'wedding': 316,
'think': 317,
'come': 318,
'republican': 319,
'must': 320,
'girl': 321,
'find': 322,
'second': 323,
'middle': 324,
'morning': 325,
'support': 326,
'same': 327,
'speech': 328,
'public': 329,
'photos': 330,
'use': 331,
'talk': 332,
'line': 333,
'car': 334,
'sanders': 335,
```

```
'name': 336,
'keep': 337,
'thinks': 338,
'run': 339,
'already': 340,
'looking': 341,
'presidential': 342,
'coming': 343,
'james': 344,
'republicans': 345,
'email': 346,
"didn't": 347,
'tax': 348,
'pretty': 349,
'case': 350,
'company': 351,
'behind': 352,
'rights': 353,
'power': 354,
'open': 355,
'future': 356,
'marriage': 357,
'between': 358,
'releases': 359,
'violence': 360,
'christmas': 361,
'security': 362,
'2016': 363,
"world's": 364,
'used': 365,
'human': 366,
'killed': 367,
'voters': 368,
'once': 369,
'control': 370,
'goes': 371,
'group': 372,
'vote': 373,
'win': 374,
'might': 375,
'democrats': 376,
'student': 377,
```

```
'full': 378,
'something': 379,
'doing': 380,
'secret': 381,
'asks': 382,
'fans': 383,
'12': 384,
'having': 385,
'team': 386,
'bernie': 387,
'department': 388,
'twitter': 389,
'room': 390,
'ban': 391,
'ad': 392,
'because': 393,
'poll': 394,
'teacher': 395,
'female': 396,
'post': 397,
'each': 398,
'wife': 399,
'inside': 400,
'ryan': 401,
'sure': 402,
'race': 403,
'claims': 404,
'music': 405,
'three': 406,
'meet': 407,
'record': 408,
'art': 409,
'forced': 410,
'boy': 411,
'15': 412,
'missing': 413,
'many': 414,
'political': 415,
'unveils': 416,
'perfect': 417,
'head': 418,
'super': 419,
```

```
'very': 420,
'photo': 421,
'judge': 422,
'running': 423,
'reports': 424,
'red': 425,
'father': 426,
'save': 427,
'class': 428,
'scientists': 429,
'month': 430,
'plans': 431,
'days': 432,
'country': 433,
'person': 434,
'living': 435,
'tells': 436,
'social': 437,
'minutes': 438,
'put': 439,
'summer': 440,
'everything': 441,
'dies': 442,
'california': 443,
'always': 444,
'until': 445,
'obamacare': 446,
'states': 447,
'here': 448,
'pay': 449,
'ready': 450,
'texas': 451,
'were': 452,
'michael': 453,
'looks': 454,
'employee': 455,
'talks': 456,
'candidate': 457,
'needs': 458,
'did': 459,
'eating': 460,
'working': 461,
```

```
'water': 462,
'list': 463,
'justice': 464,
'secretary': 465,
'shot': 466,
'hot': 467,
'warns': 468,
'times': 469,
'comes': 470,
'past': 471,
'admits': 472,
'set': 473,
'start': 474,
'taking': 475,
'wall': 476,
'heart': 477,
'ceo': 478,
'ex': 479,
'thought': 480,
"'i": 481,
'lives': 482,
'age': 483,
'left': 484,
'mike': 485,
'mother': 486,
'town': 487,
'gives': 488,
'30': 489,
'let': 490,
'cruz': 491,
"women's": 492,
'kim': 493,
'russia': 494,
'idea': 495,
'drug': 496,
'chief': 497,
'phone': 498,
"you're": 499,
'cancer': 500,
'george': 501,
'crisis': 502,
'service': 503,
```

```
'biden': 504,
'wins': 505,
'hours': 506,
"i'm": 507,
'letter': 508,
'wrong': 509,
'tips': 510,
'meeting': 511,
'south': 512,
'korea': 513,
'lost': 514,
'breaking': 515,
'daughter': 516,
'air': 517,
'50': 518,
'probably': 519,
'young': 520,
'fbi': 521,
'street': 522,
'dream': 523,
'percent': 524,
'yet': 525,
'education': 526,
'isis': 527,
'romney': 528,
'word': 529,
'thousands': 530,
'restaurant': 531,
'small': 532,
'nuclear': 533,
'fucking': 534,
'kill': 535,
'today': 536,
'believe': 537,
'king': 538,
'tweets': 539,
'together': 540,
'half': 541,
'someone': 542,
'ted': 543,
'hard': 544,
'questions': 545,
```

```
'military': 546,
'march': 547,
"she's": 548,
'few': 549,
'administration': 550,
'owner': 551,
'feel': 552,
'cat': 553,
'leaves': 554,
'fan': 555,
'internet': 556,
'officials': 557,
'third': 558,
'talking': 559,
'nothing': 560,
'director': 561,
'federal': 562,
'sleep': 563,
'chris': 564,
'rock': 565,
'place': 566,
"what's": 567,
'washington': 568,
'guide': 569,
'online': 570,
'attacks': 571,
'muslim': 572,
'earth': 573,
'giving': 574,
'move': 575,
'lot': 576,
'florida': 577,
'ask': 578,
'iran': 579,
'latest': 580,
'series': 581,
'holiday': 582,
'congressman': 583,
'community': 584,
'abortion': 585,
'well': 586,
'order': 587,
```

```
'buy': 588,
'personal': 589,
'less': 590,
'months': 591,
'majority': 592,
'birthday': 593,
'hour': 594,
't': 595,
'prison': 596,
'2015': 597,
'democratic': 598,
'outside': 599,
'problem': 600,
'leave': 601,
'assault': 602,
'those': 603,
'shit': 604,
'travel': 605,
'hollywood': 606,
'wearing': 607,
'beautiful': 608,
'girlfriend': 609,
"isn't": 610,
'ice': 611,
'reason': 612,
'bar': 613,
'francis': 614,
'told': 615,
'different': 616,
'favorite': 617,
'issues': 618,
'cover': 619,
'rules': 620,
'rise': 621,
'happy': 622,
'fox': 623,
'fun': 624,
'special': 625,
'mark': 626,
'system': 627,
'read': 628,
'watching': 629,
```

```
'reasons': 630,
'girls': 631,
'straight': 632,
'play': 633,
"america's": 634,
'al': 635,
'celebrates': 636,
"obama's": 637,
'minute': 638,
'thinking': 639,
'hate': 640,
'excited': 641,
'relationship': 642,
'trip': 643,
'hit': 644,
'response': 645,
'huffpost': 646,
'knows': 647,
'russian': 648,
'immigration': 649,
'protest': 650,
'scott': 651,
'following': 652,
'100': 653,
'using': 654,
'offers': 655,
'front': 656,
'message': 657,
'trailer': 658,
'stars': 659,
'leaders': 660,
'visit': 661,
'stephen': 662,
'hair': 663,
'huge': 664,
'box': 665,
'gift': 666,
'david': 667,
'union': 668,
'kind': 669,
'kid': 670,
'since': 671,
```

```
'moment': 672,
'china': 673,
'chinese': 674,
'birth': 675,
'non': 676,
'cop': 677,
'store': 678,
'lessons': 679,
'late': 680,
'hope': 681,
'accused': 682,
'taylor': 683,
'date': 684,
'career': 685,
'interview': 686,
'himself': 687,
'politics': 688,
'weekend': 689,
'called': 690,
'early': 691,
'victims': 692,
'least': 693,
'bring': 694,
'senator': 695,
'whole': 696,
'tom': 697,
'conversation': 698,
'adorable': 699,
'waiting': 700,
'jimmy': 701,
'break': 702,
'sports': 703,
'syria': 704,
'powerful': 705,
'drunk': 706,
'c': 707,
'point': 708,
'united': 709,
'leader': 710,
'anything': 711,
'become': 712,
'investigation': 713,
```

```
'opens': 714,
'learned': 715,
'words': 716,
'millions': 717,
'k': 718,
'die': 719,
'fashion': 720,
'cops': 721,
"they're": 722,
'reality': 723,
'billion': 724,
'fall': 725,
'key': 726,
'true': 727,
'host': 728,
'returns': 729,
'joe': 730,
'totally': 731,
'syrian': 732,
'killing': 733,
'massive': 734,
'40': 735,
'almost': 736,
'turn': 737,
'breaks': 738,
'driving': 739,
'mass': 740,
'global': 741,
'dating': 742,
'far': 743,
'policy': 744,
'schools': 745,
'stand': 746,
'trans': 747,
'dinner': 748,
'oil': 749,
'apple': 750,
'un': 751,
'awards': 752,
'queer': 753,
'worried': 754,
'kills': 755,
```

```
'iraq': 756,
'low': 757,
'song': 758,
'dance': 759,
'turns': 760,
'puts': 761,
'spends': 762,
'stage': 763,
'sign': 764,
'candidates': 765,
'j': 766,
'vows': 767,
'risk': 768,
'bus': 769,
'names': 770,
'final': 771,
'planned': 772,
'feels': 773,
'anniversary': 774,
'lgbt': 775,
'signs': 776,
'jr': 777,
'murder': 778,
'seen': 779,
'prince': 780,
'reportedly': 781,
'hits': 782,
'light': 783,
'sick': 784,
'adds': 785,
'crash': 786,
'd': 787,
'worst': 788,
'surprise': 789,
'hands': 790,
'near': 791,
'transgender': 792,
'weird': 793,
'nfl': 794,
'return': 795,
'moving': 796,
"there's": 797,
```

```
'pence': 798,
'mind': 799,
'center': 800,
'decision': 801,
'longer': 802,
'workers': 803,
'advice': 804,
'worth': 805,
'eat': 806,
'struggling': 807,
'discover': 808,
'oscar': 809,
'across': 810,
'style': 811,
'kardashian': 812,
'employees': 813,
'test': 814,
'13': 815,
'cut': 816,
'keeps': 817,
'band': 818,
'industry': 819,
'experience': 820,
'side': 821,
'coffee': 822,
'check': 823,
'2014': 824,
'number': 825,
'rubio': 826,
'brings': 827,
'door': 828,
'lead': 829,
'five': 830,
'completely': 831,
'hoping': 832,
'hand': 833,
'university': 834,
'2017': 835,
'official': 836,
'starting': 837,
'lose': 838,
'whether': 839,
```

```
'force': 840,
'paris': 841,
'weight': 842,
'road': 843,
'space': 844,
'west': 845,
'audience': 846,
'important': 847,
'steve': 848,
'playing': 849,
'reform': 850,
'cool': 851,
'fighting': 852,
'suspect': 853,
'given': 854,
'defense': 855,
'program': 856,
'artist': 857,
'nyc': 858,
'williams': 859,
'role': 860,
'building': 861,
'michelle': 862,
'peace': 863,
'carolina': 864,
'remember': 865,
'chicago': 866,
'act': 867,
'pro': 868,
'possible': 869,
'apartment': 870,
'governor': 871,
'iowa': 872,
'executive': 873,
'success': 874,
'data': 875,
'chance': 876,
'ferguson': 877,
'amazon': 878,
'biggest': 879,
'protesters': 880,
'suicide': 881,
```

```
'hall': 882,
'abuse': 883,
'which': 884,
'clearly': 885,
'major': 886,
'push': 887,
'hurricane': 888,
'moore': 889,
'allegations': 890,
'halloween': 891,
'oscars': 892,
'homeless': 893,
'israel': 894,
'general': 895,
'mental': 896,
'coworker': 897,
'moms': 898,
'board': 899,
'close': 900,
'magazine': 901,
'question': 902,
'ben': 903,
'hear': 904,
'demands': 905,
'fear': 906,
'wishes': 907,
'opening': 908,
'members': 909,
'celebrate': 910,
'supporters': 911,
'google': 912,
'football': 913,
'voice': 914,
'easy': 915,
'teens': 916,
'card': 917,
'kerry': 918,
'wait': 919,
'try': 920,
'throws': 921,
'tour': 922,
'pregnant': 923,
```

```
'pizza': 924,
'dying': 925,
'press': 926,
'chicken': 927,
'urges': 928,
'reveal': 929,
'simple': 930,
'green': 931,
'economy': 932,
'problems': 933,
'culture': 934,
'lgbtq': 935,
'asking': 936,
'ebola': 937,
'robert': 938,
'learn': 939,
'performance': 940,
'album': 941,
'church': 942,
'begins': 943,
'officer': 944,
'shop': 945,
'poor': 946,
'uses': 947,
'plane': 948,
'families': 949,
'harassment': 950,
'picture': 951,
'jobs': 952,
'fails': 953,
'sean': 954,
'voter': 955,
'beauty': 956,
'demand': 957,
'doctor': 958,
"we're": 959,
'spot': 960,
'shares': 961,
'leads': 962,
'hilarious': 963,
'suggests': 964,
'rally': 965,
```

```
'results': 966,
'ideas': 967,
'18': 968,
'jenner': 969,
'arrested': 970,
'male': 971,
'fuck': 972,
'leaving': 973,
'address': 974,
'rest': 975,
'receives': 976,
'amid': 977,
'epa': 978,
'deadly': 979,
'netflix': 980,
'desperate': 981,
'planet': 982,
'cnn': 983,
'marijuana': 984,
'quietly': 985,
'action': 986,
'website': 987,
'pick': 988,
'explains': 989,
'table': 990,
'energy': 991,
'users': 992,
'feeling': 993,
'sales': 994,
'colbert': 995,
'apparently': 996,
"let's": 997,
'amazing': 998,
'went': 999,
'budget': 1000,
...}
```

Set number of words

- Since the above 0th index doesn't have a word, add 1 to the length of the vocabulary

In [13]:
```
num_words = len(tokenizer.word_index) + 1
print(num_words)
```

```
29657
```

Loaded Glove Word Embeddings

In [14]:
```
glove_file = '/content/drive/My Drive/Colab Notebooks/glove.6B.100d.txt'
```

In [15]:
```
import numpy as np

embeddings_index = {}
f = open(glove_file)
for line in f:
    values = line.split()
    word = values[0]
    coefs = np.asarray(values[1:], dtype='float32')
    embeddings_index[word] = coefs
f.close()
```

Created embedding matrix

In [16]:
```
import numpy as np

EMBEDDING_FILE = '/content/drive/My Drive/Colab Notebooks/glove.6B.200d.txt'

embeddings = {}
for o in open(EMBEDDING_FILE):
    word = o.split(" ")[0]
    # print(word)
    embd = o.split(" ")[1:]
    embd = np.asarray(embd, dtype='float32')
    # print(embd)
    embeddings[word] = embd

# create a weight matrix for words in training docs
embedding_matrix = np.zeros((num_words, 200))

for word, i in tokenizer.word_index.items():
        embedding_vector = embeddings.get(word)
        if embedding_vector is not None:
                embedding_matrix[i] = embedding_vector
```

Defined model

- Used Sequential model instance and then add Embedding layer, Bidirectional(LSTM) layer, flatten it, then dense and dropout layers as required. -In the end added a final dense layer with sigmoid activation for binary classification.

In [17]:
```
import keras
from keras.models import Sequential
from keras.layers import Dense,Embedding,LSTM,Dropout,Bidirectional,Flatten
embedding_dim = embedding_size
#Defining Neural Network
model = Sequential()
#Non-trainable embeddidng layer
model.add(Embedding(num_words, embedding_dim, weights=[embedding_matrix], input_length=maxlen))
#LSTM
model.add(Bidirectional(LSTM(units=128 ,dropout = 0.5,return_sequences=True)))
model.add(Flatten())
model.add(Dense(1, activation='sigmoid'))
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 25, 200)           5931400
_________________________________________________________________
bidirectional (Bidirectional (None, 25, 256)           336896
_________________________________________________________________
flatten (Flatten)            (None, 6400)              0
_________________________________________________________________
dense (Dense)                (None, 1)                 6401
=================================================================
Total params: 6,274,697
Trainable params: 6,274,697
Non-trainable params: 0
_________________________________________________________________
```

In [18]:
```
print('training_sentences : ',test_sequences .shape)
print('testing_sentences : ',training_sequences .shape)
print('training_labels : ',training_labels.shape)
print('testing_labels : ',test_labels.shape)
```

```
training_sentences :  (8012, 25)
testing_sentences :  (18697, 25)
training_labels :  (18697,)
testing_labels :  (8012,)
```

Compiled the model

In [19]:
```
model.compile(optimizer=keras.optimizers.Adam(lr = 0.01), loss='binary_crossentropy', metrics=['acc'])
```

Fit the model

In [20]:
```
training_sequences = np.array(training_sequences)
training_labels = np.array(training_labels)
test_sequences = np.array(test_sequences)
test_labels = np.array(test_labels)
```

In [21]:
```
history = model.fit(x=training_sequences, y=training_labels, validation_data=(test_sequences, test_labels), epochs = 10, verbose = 1)
```

```
Epoch 1/10
585/585 [==============================] - 45s 61ms/step - loss: 0.4761 - acc: 0.7718 - val_loss: 0.3444 - val_acc:
0.8611
Epoch 2/10
585/585 [==============================] - 35s 61ms/step - loss: 0.1335 - acc: 0.9485 - val_loss: 0.4734 - val_acc:
0.8633
Epoch 3/10
585/585 [==============================] - 35s 60ms/step - loss: 0.0498 - acc: 0.9821 - val_loss: 0.6164 - val_acc:
0.8533
Epoch 4/10
585/585 [==============================] - 35s 60ms/step - loss: 0.0284 - acc: 0.9892 - val_loss: 0.8717 - val_acc:
0.8561
Epoch 5/10
585/585 [==============================] - 35s 60ms/step - loss: 0.0240 - acc: 0.9923 - val_loss: 1.0170 - val_acc:
0.8512
Epoch 6/10
585/585 [==============================] - 35s 60ms/step - loss: 0.0281 - acc: 0.9916 - val_loss: 1.3145 - val_acc:
0.8441
Epoch 7/10
585/585 [==============================] - 35s 60ms/step - loss: 0.0278 - acc: 0.9920 - val_loss: 1.5442 - val_acc:
0.8391
Epoch 8/10
585/585 [==============================] - 35s 60ms/step - loss: 0.0368 - acc: 0.9897 - val_loss: 1.7721 - val_acc:
0.8456
Epoch 9/10
585/585 [==============================] - 35s 60ms/step - loss: 0.0335 - acc: 0.9909 - val_loss: 1.9952 - val_acc:
0.8455
Epoch 10/10
585/585 [==============================] - 35s 60ms/step - loss: 0.0413 - acc: 0.9917 - val_loss: 2.2087 - val_acc:
0.8399
```

Test accuracy

In [23]:
```
print("Test-Accuracy:", np.mean(history.history["val_acc"]))
```

```
Test-Accuracy: 0.8499251067638397
```

Accuracy of the model

In [24]:
```
scores = model.evaluate(test_sequences, test_labels, verbose=0)
print("Accuracy: %.5f%%" % (scores[1]*100))
```

```
Accuracy: 83.98652%
```

In [ ]:
```

```