# Emoji Predictor

- In this project we will be given a sentence and will output an emoji best suited for the sentiment of that sentence.
   emoji predictor example
- We will be working with custom dataset.
- **Steps**:
- 1) Working with Emoji Package
- 2) Process .csv files (custom dataset)
- 3) Transfer Learning (Glove Vector for word embeddings)
- 4) RNN/LSTM Based Model
- 5) Study in detail about Stacked LSTM architecture and implement it for better results
- 6) Prediction and testing accuracy

## 1) Working with Emoji Package

- We will need to install "emoji" package using pip

In [1]:
```
!pip install emoji
```

```
Requirement already satisfied: emoji in c:\programdata\anaconda3\lib\site-packages (0.5.4)
```

In [3]:
```
import emoji
import numpy as np
import pandas as pd
```

In [3]:
```
# We will use ".EMOJI_UNICODE" attribute to print the dictionary of this dataset in whic
h
```

```
# keys --> description of emoji || values --> actual emoji
emoji.EMOJI_UNICODE
```

```
{':1st_place_medal:': '🥇',
 ':2nd_place_medal:': '🥈',
 ':3rd_place_medal:': '🥉',
 ':AB_button_(blood_type):': '🆎',
 ':ATM_sign:': '🏧',
 ':A_button_(blood_type):': '🅰',
 ':Afghanistan:': '🇦🇫',
 ':Albania:': '🇦🇱',
 ':Algeria:': '🇩🇿',
 ':American_Samoa:': '🇦🇸',
 ':Andorra:': '🇦🇩',
 ':Angola:': '🇦🇴',
 ':Anguilla:': '🇦🇮',
 ':Antarctica:': '🇦🇶',
 ':Antigua_&_Barbuda:': '🇦🇬',
 ':Aquarius:': '♒',
 ':Argentina:': '🇦🇷',
 ':Aries:': '♈',
 ':Armenia:': '🇦🇲',
 ':Aruba:': '🇦🇼',
 ':Ascension_Island:': '🇦🇨',
 ':Australia:': '🇦🇺',
 ':Austria:': '🇦🇹',
 ':Azerbaijan:': '🇦🇿',
 ':BACK_arrow:': '🔙',
 ':B_button_(blood_type):': '🅱',
 ':Bahamas:': '🇧🇸',
 ':Bahrain:': '🇧🇭',
 ':Bangladesh:': '🇧🇩',
 ':Barbados:': '🇧🇧',
 ':Belarus:': '🇧🇾',
 ':Belgium:': '🇧🇪',
 ':Belize:': '🇧🇿',
 ':Benin:': '🇧🇯',
 ':Bermuda:': '🇧🇲',
 ':Bhutan:': '🇧🇹',
```

```
':Bolivia:': '🇧🇴',
':Bosnia_&_Herzegovina:': '🇧🇦',
':Botswana:': '🇧🇼',
':Bouvet_Island:': '🇧🇻',
':Brazil:': '🇧🇷',
':British_Indian_Ocean_Territory:': '🇮🇴',
':British_Virgin_Islands:': '🇻🇬',
':Brunei:': '🇧🇳',
':Bulgaria:': '🇧🇬',
':Burkina_Faso:': '🇧🇫',
':Burundi:': '🇧🇮',
':CL_button:': '🆑',
':COOL_button:': '🆒',
':Cambodia:': '🇰🇭',
':Cameroon:': '🇨🇲',
':Canada:': '🇨🇦',
':Canary_Islands:': '🇮🇨',
':Cancer:': '♋',
':Cape_Verde:': '🇨🇻',
':Capricorn:': '♑',
':Caribbean_Netherlands:': '🇧🇶',
':Cayman_Islands:': '🇰🇾',
':Central_African_Republic:': '🇨🇫',
':Ceuta_&_Melilla:': '🇪🇦',
':Chad:': '🇹🇩',
':Chile:': '🇨🇱',
':China:': '🇨🇳',
':Christmas_Island:': '🇨🇽',
':Christmas_tree:': '🎄',
':Clipperton_Island:': '🇨🇵',
':Cocos_(Keeling)_Islands:': '🇨🇨',
':Colombia:': '🇨🇴',
':Comoros:': '🇰🇲',
':Congo_-_Brazzaville:': '🇨🇬',
':Congo_-_Kinshasa:': '🇨🇩',
':Cook_Islands:': '🇨🇰',
':Costa_Rica:': '🇨🇷',
':Croatia:': '🇭🇷',
':Cuba:': '🇨🇺',
```

```
':Curaçao:': '🇨🇼',
':Cyprus:': '🇨🇾',
':Czechia:': '🇨🇿',
':Côte_d’Ivoire:': '🇨🇮',
':Denmark:': '🇩🇰',
':Diego_Garcia:': '🇩🇬',
':Djibouti:': '🇩🇯',
':Dominica:': '🇩🇲',
':Dominican_Republic:': '🇩🇴',
':END_arrow:': '🔚',
':Ecuador:': '🇪🇨',
':Egypt:': '🇪🇬',
':El_Salvador:': '🇸🇻',
':England:': '🏴\U000e0067\U000e0062\U000e0065\U000e006e\U000e0067\U000e007f',
':Equatorial_Guinea:': '🇬🇶',
':Eritrea:': '🇪🇷',
':Estonia:': '🇪🇪',
':Ethiopia:': '🇪🇹',
':European_Union:': '🇪🇺',
':FREE_button:': '🆓',
':Falkland_Islands:': '🇫🇰',
':Faroe_Islands:': '🇫🇴',
':Fiji:': '🇫🇯',
':Finland:': '🇫🇮',
':France:': '🇫🇷',
':French_Guiana:': '🇬🇫',
':French_Polynesia:': '🇵🇫',
':French_Southern_Territories:': '🇹🇫',
':Gabon:': '🇬🇦',
':Gambia:': '🇬🇲',
':Gemini:': '♊',
':Georgia:': '🇬🇪',
':Germany:': '🇩🇪',
':Ghana:': '🇬🇭',
':Gibraltar:': '🇬🇮',
':Greece:': '🇬🇷',
':Greenland:': '🇬🇱',
':Grenada:': '🇬🇩',
':Guadeloupe:': '🇬🇵',
```

```
':Guam:': '🇬🇺',
':Guatemala:': '🇬🇹',
':Guernsey:': '🇬🇬',
':Guinea:': '🇬🇳',
':Guinea-Bissau:': '🇬🇼',
':Guyana:': '🇬🇾',
':Haiti:': '🇭🇹',
':Heard_&_McDonald_Islands:': '🇭🇲',
':Honduras:': '🇭🇳',
':Hong_Kong_SAR_China:': '🇭🇰',
':Hungary:': '🇭🇺',
':ID_button:': '🆔',
':Iceland:': '🇮🇸',
':India:': '🇮🇳',
':Indonesia:': '🇮🇩',
':Iran:': '🇮🇷',
':Iraq:': '🇮🇶',
':Ireland:': '🇮🇪',
':Isle_of_Man:': '🇮🇲',
':Israel:': '🇮🇱',
':Italy:': '🇮🇹',
':Jamaica:': '🇯🇲',
':Japan:': '🇯🇵',
':Japanese_acceptable_button:': '㊆',
':Japanese_application_button:': '申',
':Japanese_bargain_button:': '㊊',
':Japanese_castle:': '🏯',
':Japanese_congratulations_button:': '㊗',
':Japanese_discount_button:': '割',
':Japanese_dolls:': '🎎',
':Japanese_free_of_charge_button:': '無',
':Japanese_here_button:': '🈁',
':Japanese_monthly_amount_button:': '月',
':Japanese_no_vacancy_button:': '満',
':Japanese_not_free_of_charge_button:': '有',
':Japanese_open_for_business_button:': '営',
':Japanese_passing_grade_button:': '合',
':Japanese_post_office:': '〒',
':Japanese_prohibited_button:': '禁',
```

```
':Japanese_reserved_button:': '🈯',
':Japanese_secret_button:': '㊙',
':Japanese_service_charge_button:': '🈂',
':Japanese_symbol_for_beginner:': '🔰',
':Japanese_vacancy_button:': '🈳',
':Jersey:': '🇯🇪',
':Jordan:': '🇯🇴',
':Kazakhstan:': '🇰🇿',
':Kenya:': '🇰🇪',
':Kiribati:': '🇰🇮',
':Kosovo:': '🇽🇰',
':Kuwait:': '🇰🇼',
':Kyrgyzstan:': '🇰🇬',
':Laos:': '🇱🇦',
':Latvia:': '🇱🇻',
':Lebanon:': '🇱🇧',
':Leo:': '♌',
':Lesotho:': '🇱🇸',
':Liberia:': '🇱🇷',
':Libra:': '♎',
':Libya:': '🇱🇾',
':Liechtenstein:': '🇱🇮',
':Lithuania:': '🇱🇹',
':Luxembourg:': '🇱🇺',
':Macau_SAR_China:': '🇲🇴',
':Macedonia:': '🇲🇰',
':Madagascar:': '🇲🇬',
':Malawi:': '🇲🇼',
':Malaysia:': '🇲🇾',
':Maldives:': '🇲🇻',
':Mali:': '🇲🇱',
':Malta:': '🇲🇹',
':Marshall_Islands:': '🇲🇭',
':Martinique:': '🇲🇶',
':Mauritania:': '🇲🇷',
':Mauritius:': '🇲🇺',
':Mayotte:': '🇾🇹',
':Mexico:': '🇲🇽',
':Micronesia:': '🇫🇲',
```

```
':Moldova:': '🇲🇩',
':Monaco:': '🇲🇨',
':Mongolia:': '🇲🇳',
':Montenegro:': '🇲🇪',
':Montserrat:': '🇲🇸',
':Morocco:': '🇲🇦',
':Mozambique:': '🇲🇿',
':Mrs._Claus:': '🤶',
':Mrs._Claus_dark_skin_tone:': '🤶🏿',
':Mrs._Claus_light_skin_tone:': '🤶🏻',
':Mrs._Claus_medium-dark_skin_tone:': '🤶🏾',
':Mrs._Claus_medium-light_skin_tone:': '🤶🏼',
':Mrs._Claus_medium_skin_tone:': '🤶🏽',
':Myanmar_(Burma):': '🇲🇲',
':NEW_button:': '🆕',
':NG_button:': '🆖',
':Namibia:': '🇳🇦',
':Nauru:': '🇳🇷',
':Nepal:': '🇳🇵',
':Netherlands:': '🇳🇱',
':New_Caledonia:': '🇳🇨',
':New_Zealand:': '🇳🇿',
':Nicaragua:': '🇳🇮',
':Niger:': '🇳🇪',
':Nigeria:': '🇳🇬',
':Niue:': '🇳🇺',
':Norfolk_Island:': '🇳🇫',
':North_Korea:': '🇰🇵',
':Northern_Mariana_Islands:': '🇲🇵',
':Norway:': '🇳🇴',
':OK_button:': '🆗',
':OK_hand:': '👌',
':OK_hand_dark_skin_tone:': '👌🏿',
':OK_hand_light_skin_tone:': '👌🏻',
':OK_hand_medium-dark_skin_tone:': '👌🏾',
':OK_hand_medium-light_skin_tone:': '👌🏼',
':OK_hand_medium_skin_tone:': '👌🏽',
':ON!_arrow:': '🔛',
':O_button_(blood_type):': '🅾',
```

```
':Oman:': '🇴🇲',
':Ophiuchus:': '⛎',
':P_button:': '🅿',
':Pakistan:': '🇵🇰',
':Palau:': '🇵🇼',
':Palestinian_Territories:': '🇵🇸',
':Panama:': '🇵🇦',
':Papua_New_Guinea:': '🇵🇬',
':Paraguay:': '🇵🇾',
':Peru:': '🇵🇪',
':Philippines:': '🇵🇭',
':Pisces:': '♓',
':Pitcairn_Islands:': '🇵🇳',
':Poland:': '🇵🇱',
':Portugal:': '🇵🇹',
':Puerto_Rico:': '🇵🇷',
':Qatar:': '🇶🇦',
':Romania:': '🇷🇴',
':Russia:': '🇷🇺',
':Rwanda:': '🇷🇼',
':Réunion:': '🇷🇪',
':SOON_arrow:': '🔜',
':SOS_button:': '🆘',
':Sagittarius:': '♐',
':Samoa:': '🇼🇸',
':San_Marino:': '🇸🇲',
':Santa_Claus:': '🎅',
':Santa_Claus_dark_skin_tone:': '🎅🏿',
':Santa_Claus_light_skin_tone:': '🎅🏻',
':Santa_Claus_medium-dark_skin_tone:': '🎅🏾',
':Santa_Claus_medium-light_skin_tone:': '🎅🏼',
':Santa_Claus_medium_skin_tone:': '🎅🏽',
':Saudi_Arabia:': '🇸🇦',
':Scorpio:': '♏',
':Scotland:': '🏴\U000e0067\U000e0062\U000e0073\U000e0063\U000e0074\U000e007f',
':Senegal:': '🇸🇳',
':Serbia:': '🇷🇸',
':Seychelles:': '🇸🇨',
':Sierra_Leone:': '🇸🇱',
```

```
':Singapore:': '🇸🇬',
':Sint_Maarten:': '🇸🇽',
':Slovakia:': '🇸🇰',
':Slovenia:': '🇸🇮',
':Solomon_Islands:': '🇸🇧',
':Somalia:': '🇸🇴',
':South_Africa:': '🇿🇦',
':South_Georgia_&_South_Sandwich_Islands:': '🇬🇸',
':South_Korea:': '🇰🇷',
':South_Sudan:': '🇸🇸',
':Spain:': '🇪🇸',
':Sri_Lanka:': '🇱🇰',
':St._Barthélemy:': '🇧🇱',
':St._Helena:': '🇸🇭',
':St._Kitts_&_Nevis:': '🇰🇳',
':St._Lucia:': '🇱🇨',
':St._Martin:': '🇲🇫',
':St._Pierre_&_Miquelon:': '🇵🇲',
':St._Vincent_&_Grenadines:': '🇻🇨',
':Statue_of_Liberty:': '🗽',
':Sudan:': '🇸🇩',
':Suriname:': '🇸🇷',
':Svalbard_&_Jan_Mayen:': '🇸🇯',
':Swaziland:': '🇸🇿',
':Sweden:': '🇸🇪',
':Switzerland:': '🇨🇭',
':Syria:': '🇸🇾',
':São_Tomé_&_Príncipe:': '🇸🇹',
':T-Rex:': '\U0001f996',
':TOP_arrow:': '🔝',
':Taiwan:': '🇹🇼',
':Tajikistan:': '🇹🇯',
':Tanzania:': '🇹🇿',
':Taurus:': '♉',
':Thailand:': '🇹🇭',
':Timor-Leste:': '🇹🇱',
':Togo:': '🇹🇬',
':Tokelau:': '🇹🇰',
':Tokyo_tower:': '🗼',
```

```
':Tonga:': '🇹🇴',
':Trinidad_&_Tobago:': '🇹🇹',
':Tristan_da_Cunha:': '🇹🇦',
':Tunisia:': '🇹🇳',
':Turkey:': '🇹🇷',
':Turkmenistan:': '🇹🇲',
':Turks_&_Caicos_Islands:': '🇹🇨',
':Tuvalu:': '🇹🇻',
':U.S._Outlying_Islands:': '🇺🇲',
':U.S._Virgin_Islands:': '🇻🇮',
':UP!_button:': '🆙',
':Uganda:': '🇺🇬',
':Ukraine:': '🇺🇦',
':United_Arab_Emirates:': '🇦🇪',
':United_Kingdom:': '🇬🇧',
':United_Nations:': '🇺🇳',
':United_States:': '🇺🇸',
':Uruguay:': '🇺🇾',
':Uzbekistan:': '🇺🇿',
':VS_button:': '🆚',
':Vanuatu:': '🇻🇺',
':Vatican_City:': '🇻🇦',
':Venezuela:': '🇻🇪',
':Vietnam:': '🇻🇳',
':Virgo:': '♍',
':Wales:': '🏴\U000e0067\U000e0062\U000e0077\U000e006c\U000e0073\U000e007f',
':Wallis_&_Futuna:': '🇼🇫',
':Western_Sahara:': '🇪🇭',
':Yemen:': '🇾🇪',
':Zambia:': '🇿🇲',
':Zimbabwe:': '🇿🇼',
':abacus:': '\U0001f9ee',
':adhesive_bandage:': '\U0001fa79',
':admission_tickets:': '🎟',
':adult:': '\U0001f9d1',
':adult_dark_skin_tone:': '\U0001f9d1🏿',
':adult_light_skin_tone:': '\U0001f9d1🏻',
':adult_medium-dark_skin_tone:': '\U0001f9d1🏾',
':adult_medium-light_skin_tone:': '\U0001f9d1🏼',
```

```
':adult_medium_skin_tone:': '\U0001f9d1🏽',
':aerial_tramway:': '🚡',
':airplane:': '✈',
':airplane_arrival:': '🛬',
':airplane_departure:': '🛫',
':alarm_clock:': '⏰',
':alembic:': '⚗',
':alien:': '👽',
':alien_monster:': '👾',
':ambulance:': '🚑',
':american_football:': '🏈',
':amphora:': '🏺',
':anchor:': '⚓',
':anger_symbol:': '💢',
':angry_face:': '😠',
':angry_face_with_horns:': '👿',
':anguished_face:': '😧',
':ant:': '🐜',
':antenna_bars:': '📶',
':anxious_face_with_sweat:': '😰',
':articulated_lorry:': '🚛',
':artist_palette:': '🎨',
':astonished_face:': '😲',
':atom_symbol:': '⚛',
':auto_rickshaw:': '\U0001f6fa',
':automobile:': '🚗',
':avocado:': '🥑',
':axe:': '\U0001fa93',
':baby:': '👶',
':baby_angel:': '👼',
':baby_angel_dark_skin_tone:': '👼🏿',
':baby_angel_light_skin_tone:': '👼🏻',
':baby_angel_medium-dark_skin_tone:': '👼🏾',
':baby_angel_medium-light_skin_tone:': '👼🏼',
':baby_angel_medium_skin_tone:': '👼🏽',
':baby_bottle:': '🍼',
':baby_chick:': '🐤',
':baby_dark_skin_tone:': '👶🏿',
':baby_light_skin_tone:': '👶🏻',
```

```
':baby_medium-dark_skin_tone:': '👶🏾',
':baby_medium-light_skin_tone:': '👶🏼',
':baby_medium_skin_tone:': '👶🏽',
':baby_symbol:': '🚼',
':backhand_index_pointing_down:': '👇',
':backhand_index_pointing_down_dark_skin_tone:': '👇🏿',
':backhand_index_pointing_down_light_skin_tone:': '👇🏻',
':backhand_index_pointing_down_medium-dark_skin_tone:': '👇🏾',
':backhand_index_pointing_down_medium-light_skin_tone:': '👇🏼',
':backhand_index_pointing_down_medium_skin_tone:': '👇🏽',
':backhand_index_pointing_left:': '👈',
':backhand_index_pointing_left_dark_skin_tone:': '👈🏿',
':backhand_index_pointing_left_light_skin_tone:': '👈🏻',
':backhand_index_pointing_left_medium-dark_skin_tone:': '👈🏾',
':backhand_index_pointing_left_medium-light_skin_tone:': '👈🏼',
':backhand_index_pointing_left_medium_skin_tone:': '👈🏽',
':backhand_index_pointing_right:': '👉',
':backhand_index_pointing_right_dark_skin_tone:': '👉🏿',
':backhand_index_pointing_right_light_skin_tone:': '👉🏻',
':backhand_index_pointing_right_medium-dark_skin_tone:': '👉🏾',
':backhand_index_pointing_right_medium-light_skin_tone:': '👉🏼',
':backhand_index_pointing_right_medium_skin_tone:': '👉🏽',
':backhand_index_pointing_up:': '👆',
':backhand_index_pointing_up_dark_skin_tone:': '👆🏿',
':backhand_index_pointing_up_light_skin_tone:': '👆🏻',
':backhand_index_pointing_up_medium-dark_skin_tone:': '👆🏾',
':backhand_index_pointing_up_medium-light_skin_tone:': '👆🏼',
':backhand_index_pointing_up_medium_skin_tone:': '👆🏽',
':bacon:': '🥓',
':badger:': '\U0001f9a1',
':badminton:': '🏸',
':bagel:': '\U0001f96f',
':baggage_claim:': '🛄',
':baguette_bread:': '🥖',
':balance_scale:': '⚖',
':bald:': '\U0001f9b2',
':bald_man:': '👨\u200d\U0001f9b2',
':bald_woman:': '👩\u200d\U0001f9b2',
':ballet_shoes:': '\U0001fa70',
```

```
':balloon:': '🎈',
':ballot_box_with_ballot:': '🗳',
':ballot_box_with_check:': '☑',
':banana:': '🍌',
':banjo:': '\U0001fa95',
':bank:': '🏦',
':bar_chart:': '📊',
':barber_pole:': '💈',
':baseball:': '⚾',
':basket:': '\U0001f9fa',
':basketball:': '🏀',
':bat:': '🦇',
':bathtub:': '🛁',
':battery:': '🔋',
':beach_with_umbrella:': '🏖',
':beaming_face_with_smiling_eyes:': '😁',
':bear_face:': '🐻',
':bearded_person:': '\U0001f9d4',
':bearded_person_dark_skin_tone:': '\U0001f9d4🏿',
':bearded_person_light_skin_tone:': '\U0001f9d4🏻',
':bearded_person_medium-dark_skin_tone:': '\U0001f9d4🏾',
':bearded_person_medium-light_skin_tone:': '\U0001f9d4🏼',
':bearded_person_medium_skin_tone:': '\U0001f9d4🏽',
':beating_heart:': '💓',
':bed:': '🛏',
':beer_mug:': '🍺',
':bell:': '🔔',
':bell_with_slash:': '🔕',
':bellhop_bell:': '🛎',
':bento_box:': '🍱',
':beverage_box:': '\U0001f9c3',
':bicycle:': '🚲',
':bikini:': '👙',
':billed_cap:': '\U0001f9e2',
':biohazard:': '☣',
':bird:': '🐦',
':birthday_cake:': '🎂',
':black_circle:': '●',
':black_flag:': '🏴',
```

```
':black_heart:': '🖤',
':black_large_square:': '⬛',
':black_medium-small_square:': '◾',
':black_medium_square:': '◼',
':black_nib:': '✒',
':black_small_square:': '▪',
':black_square_button:': '🔲',
':blond-haired_man:': '👱\u200d♂',
':blond-haired_man_dark_skin_tone:': '👱🏿\u200d♂',
':blond-haired_man_light_skin_tone:': '👱🏻\u200d♂',
':blond-haired_man_medium-dark_skin_tone:': '👱🏾\u200d♂',
':blond-haired_man_medium-light_skin_tone:': '👱🏼\u200d♂',
':blond-haired_man_medium_skin_tone:': '👱🏽\u200d♂',
':blond-haired_person:': '👱',
':blond-haired_person_dark_skin_tone:': '👱🏿',
':blond-haired_person_light_skin_tone:': '👱🏻',
':blond-haired_person_medium-dark_skin_tone:': '👱🏾',
':blond-haired_person_medium-light_skin_tone:': '👱🏼',
':blond-haired_person_medium_skin_tone:': '👱🏽',
':blond-haired_woman:': '👱\u200d♀',
':blond-haired_woman_dark_skin_tone:': '👱🏿\u200d♀',
':blond-haired_woman_light_skin_tone:': '👱🏻\u200d♀',
':blond-haired_woman_medium-dark_skin_tone:': '👱🏾\u200d♀',
':blond-haired_woman_medium-light_skin_tone:': '👱🏼\u200d♀',
':blond-haired_woman_medium_skin_tone:': '👱🏽\u200d♀',
':blossom:': '🌼',
':blowfish:': '🐡',
':blue_book:': '📘',
':blue_circle:': '🔵',
':blue_heart:': '💙',
':blue_square:': '\U0001f7e6',
':boar:': '🐗',
':bomb:': '💣',
':bone:': '\U0001f9b4',
':bookmark:': '🔖',
':bookmark_tabs:': '📑',
':books:': '📚',
':bottle_with_popping_cork:': '🍾',
':bouquet:': '💐',
```

```
':bow_and_arrow:': '🏹',
':bowl_with_spoon:': '\U0001f963',
':bowling:': '🎳',
':boxing_glove:': '🥊',
':boy:': '👦',
':boy_dark_skin_tone:': '👦🏿',
':boy_light_skin_tone:': '👦🏻',
':boy_medium-dark_skin_tone:': '👦🏾',
':boy_medium-light_skin_tone:': '👦🏽',
':boy_medium_skin_tone:': '👦🏽',
':brain:': '\U0001f9e0',
':bread:': '🍞',
':breast-feeding:': '\U0001f931',
':breast-feeding_dark_skin_tone:': '\U0001f931🏿',
':breast-feeding_light_skin_tone:': '\U0001f931🏻',
':breast-feeding_medium-dark_skin_tone:': '\U0001f931🏾',
':breast-feeding_medium-light_skin_tone:': '\U0001f931🏽',
':breast-feeding_medium_skin_tone:': '\U0001f931🏽',
':brick:': '\U0001f9f1',
':bride_with_veil:': '👰',
':bride_with_veil_dark_skin_tone:': '👰🏿',
':bride_with_veil_light_skin_tone:': '👰🏻',
':bride_with_veil_medium-dark_skin_tone:': '👰🏾',
':bride_with_veil_medium-light_skin_tone:': '👰🏽',
':bride_with_veil_medium_skin_tone:': '👰🏽',
':bridge_at_night:': '🌉',
':briefcase:': '💼',
':briefs:': '\U0001fa72',
':bright_button:': '🔆',
':broccoli:': '\U0001f966',
':broken_heart:': '💔',
':broom:': '\U0001f9f9',
':brown_circle:': '\U0001f7e4',
':brown_heart:': '\U0001f90e',
':brown_square:': '\U0001f7eb',
':bug:': '🐛',
':building_construction:': '🏗',
':bullet_train:': '🚅',
':burrito:': '🌯',
```

```
':bus:': '🚌',
':bus_stop:': '🚏',
':bust_in_silhouette:': '👤',
':busts_in_silhouette:': '👥',
':butter:': '\U0001f9c8',
':butterfly:': '🦋',
':cactus:': '🌵',
':calendar:': '📆',
':call_me_hand:': '🤙',
':call_me_hand_dark_skin_tone:': '🤙🏿',
':call_me_hand_light_skin_tone:': '🤙🏻',
':call_me_hand_medium-dark_skin_tone:': '🤙🏾',
':call_me_hand_medium-light_skin_tone:': '🤙🏼',
':call_me_hand_medium_skin_tone:': '🤙🏽',
':camel:': '🐫',
':camera:': '📷',
':camera_with_flash:': '📸',
':camping:': '🏕',
':candle:': '🕯',
':candy:': '🍬',
':canned_food:': '\U0001f96b',
':canoe:': '🛶',
':card_file_box:': '🗃',
':card_index:': '📇',
':card_index_dividers:': '🗂',
':carousel_horse:': '🎠',
':carp_streamer:': '🎏',
':carrot:': '🥕',
':castle:': '🏰',
':cat:': '🐈',
':cat_face:': '😺',
':cat_face_with_tears_of_joy:': '😹',
':cat_face_with_wry_smile:': '😼',
':chains:': '⛓',
':chair:': '\U0001fa91',
':chart_decreasing:': '📉',
':chart_increasing:': '📈',
':chart_increasing_with_yen:': '💹',
':cheese_wedge:': '🧀',
```

```
':chequered_flag:': '🏁',
':cherries:': '🍒',
':cherry_blossom:': '🌸',
':chess_pawn:': '♟',
':chestnut:': '🌰',
':chicken:': '🐔',
':child:': '\U0001f9d2',
':child_dark_skin_tone:': '\U0001f9d2🏿',
':child_light_skin_tone:': '\U0001f9d2🏻',
':child_medium-dark_skin_tone:': '\U0001f9d2🏾',
':child_medium-light_skin_tone:': '\U0001f9d2🏼',
':child_medium_skin_tone:': '\U0001f9d2🏽',
':children_crossing:': '🚸',
':chipmunk:': '🐿',
':chocolate_bar:': '🍫',
':chopsticks:': '\U0001f962',
':church:': '⛪',
':cigarette:': '🚬',
':cinema:': '🎦',
':circled_M:': 'Ⓜ',
':circus_tent:': '🎪',
':cityscape:': '🏙',
':cityscape_at_dusk:': '🌆',
':clamp:': '🗜',
':clapper_board:': '🎬',
':clapping_hands:': '👏',
':clapping_hands_dark_skin_tone:': '👏🏿',
':clapping_hands_light_skin_tone:': '👏🏻',
':clapping_hands_medium-dark_skin_tone:': '👏🏾',
':clapping_hands_medium-light_skin_tone:': '👏🏼',
':clapping_hands_medium_skin_tone:': '👏🏽',
':classical_building:': '🏛',
':clinking_beer_mugs:': '🍻',
':clinking_glasses:': '🥂',
':clipboard:': '📋',
':clockwise_vertical_arrows:': '🔃',
':closed_book:': '📕',
':closed_mailbox_with_lowered_flag:': '📪',
':closed_mailbox_with_raised_flag:': '📫',
```

```
':closed_umbrella:': '☂',
':cloud:': '☁',
':cloud_with_lightning:': '🌩',
':cloud_with_lightning_and_rain:': '⛈',
':cloud_with_rain:': '🌧',
':cloud_with_snow:': '🌨',
':clown_face:': '🤡',
':club_suit:': '♣',
':clutch_bag:': '👝',
':coat:': '\U0001f9e5',
':cocktail_glass:': '🍸',
':coconut:': '\U0001f965',
':coffin:': '⚰',
':cold_face:': '\U0001f976',
':collision:': '💥',
':comet:': '☄',
':compass:': '\U0001f9ed',
':computer_disk:': '💽',
':computer_mouse:': '🖱',
':confetti_ball:': '🎊',
':confounded_face:': '😖',
':confused_face:': '😕',
':construction:': '🚧',
':construction_worker:': '👷',
':construction_worker_dark_skin_tone:': '👷🏿',
':construction_worker_light_skin_tone:': '👷🏻',
':construction_worker_medium-dark_skin_tone:': '👷🏾',
':construction_worker_medium-light_skin_tone:': '👷🏼',
':construction_worker_medium_skin_tone:': '👷🏽',
':control_knobs:': '🎛',
':convenience_store:': '🏪',
':cooked_rice:': '🍚',
':cookie:': '🍪',
':cooking:': '🍳',
':copyright:': '©',
':couch_and_lamp:': '🛋',
':counterclockwise_arrows_button:': '🔄',
':couple_with_heart:': '💑',
':couple_with_heart_man_man:': '👨\u200d❤\u200d👨',
```

```
':couple_with_heart_woman_man:': '👩\u200d❤\u200d👨',
':couple_with_heart_woman_woman:': '👩\u200d❤\u200d👩',
':cow:': '🐄',
':cow_face:': '🐮',
':cowboy_hat_face:': '🤠',
':crab:': '🦀',
':crayon:': '🖍',
':credit_card:': '💳',
':crescent_moon:': '🌙',
':cricket:': '\U0001f997',
':cricket_game:': '🏏',
':crocodile:': '🐊',
':croissant:': '🥐',
':cross_mark:': '❌',
':cross_mark_button:': '❎',
':crossed_fingers:': '🤞',
':crossed_fingers_dark_skin_tone:': '🤞🏿',
':crossed_fingers_light_skin_tone:': '🤞🏻',
':crossed_fingers_medium-dark_skin_tone:': '🤞🏾',
':crossed_fingers_medium-light_skin_tone:': '🤞🏼',
':crossed_fingers_medium_skin_tone:': '🤞🏽',
':crossed_flags:': '🎌',
':crossed_swords:': '⚔',
':crown:': '👑',
':crying_cat_face:': '😿',
':crying_face:': '😢',
':crystal_ball:': '🔮',
':cucumber:': '🥒',
':cupcake:': '\U0001f9c1',
':cup_with_straw:': '\U0001f964',
':curling_stone:': '\U0001f94c',
':curly_hair:': '\U0001f9b1',
':curly-haired_man:': '👨\u200d\U0001f9b1',
':curly-haired_woman:': '👩\u200d\U0001f9b1',
':curly_loop:': '➰',
':currency_exchange:': '💱',
':curry_rice:': '🍛',
':custard:': '🍮',
':customs:': '🛃',
```

```
':cut_of_meat:': '\U0001f969',
':cyclone:': '🌀',
':dagger:': '🗡',
':dango:': '🍡',
':dashing_away:': '💨',
':deaf_person:': '\U0001f9cf',
':deciduous_tree:': '🌳',
':deer:': '🦌',
':delivery_truck:': '🚚',
':department_store:': '🏬',
':derelict_house:': '🏚',
':desert:': '🏜',
':desert_island:': '🏝',
':desktop_computer:': '🖥',
':detective:': '🕵',
':detective_dark_skin_tone:': '🕵🏿',
':detective_light_skin_tone:': '🕵🏻',
':detective_medium-dark_skin_tone:': '🕵🏾',
':detective_medium-light_skin_tone:': '🕵🏼',
':detective_medium_skin_tone:': '🕵🏽',
':diamond_suit:': '♦',
':diamond_with_a_dot:': '💠',
':dim_button:': '🔅',
':direct_hit:': '🎯',
':disappointed_face:': '😞',
':diving_mask:': '\U0001f93f',
':diya_lamp:': '\U0001fa94',
':dizzy:': '💫',
':dizzy_face:': '😵',
':dna:': '\U0001f9ec',
':dog:': '🐕',
':dog_face:': '🐶',
':dollar_banknote:': '💵',
':dolphin:': '🐬',
':door:': '🚪',
':dotted_six-pointed_star:': '🔯',
':double_curly_loop:': '➿',
':double_exclamation_mark:': '‼',
':doughnut:': '🍩',
```

```
':dove:': '🕊',
':down-left_arrow:': '↙',
':down-right_arrow:': '↘',
':down_arrow:': '⬇',
':downcast_face_with_sweat:': '😓',
':downwards_button:': '🔽',
':dragon:': '🐉',
':dragon_face:': '🐲',
':dress:': '👗',
':drooling_face:': '🤤',
':drop_of_blood:': '\U0001fa78',
':droplet:': '💧',
':drum:': '🥁',
':duck:': '🦆',
':dumpling:': '\U0001f95f',
':dvd:': '📀',
':e-mail:': '📧',
':eagle:': '🦅',
':ear:': '👂',
':ear_dark_skin_tone:': '👂🏿',
':ear_light_skin_tone:': '👂🏻',
':ear_medium-dark_skin_tone:': '👂🏾',
':ear_medium-light_skin_tone:': '👂🏼',
':ear_medium_skin_tone:': '👂🏽',
':ear_of_corn:': '🌽',
':ear_with_hearing_aid:': '\U0001f9bb',
':egg:': '🥚',
':eggplant:': '🍆',
':eight-pointed_star:': '✴',
':eight-spoked_asterisk:': '✳',
':eight-thirty:': '🕣',
':eight_o’clock:': '🕗',
':eject_button:': '⏏',
':electric_plug:': '🔌',
':elephant:': '🐘',
':eleven-thirty:': '🕦',
':eleven_o’clock:': '🕚',
':elf:': '\U0001f9dd',
':elf_dark_skin_tone:': '\U0001f9dd🏿',
```

```
':elf_light_skin_tone:': '\U0001f9dd🏻',
':elf_medium-dark_skin_tone:': '\U0001f9dd🏾',
':elf_medium-light_skin_tone:': '\U0001f9dd🏼',
':elf_medium_skin_tone:': '\U0001f9dd🏽',
':envelope:': '✉',
':envelope_with_arrow:': '📩',
':euro_banknote:': '💶',
':evergreen_tree:': '🌲',
':ewe:': '🐑',
':exclamation_mark:': '❗',
':exclamation_question_mark:': '⁉',
':exploding_head:': '\U0001f92f',
':expressionless_face:': '😑',
':eye:': '👁',
':eye_in_speech_bubble:': '👁\u200d🗨',
':eyes:': '👀',
':face_blowing_a_kiss:': '😘',
':face_savoring_food:': '😋',
':face_screaming_in_fear:': '😱',
':face_vomiting:': '\U0001f92e',
':face_with_hand_over_mouth:': '\U0001f92d',
':face_with_head-bandage:': '🤕',
':face_with_medical_mask:': '😷',
':face_with_monocle:': '\U0001f9d0',
':face_with_open_mouth:': '😮',
':face_with_raised_eyebrow:': '\U0001f928',
':face_with_rolling_eyes:': '🙄',
':face_with_steam_from_nose:': '😤',
':face_with_symbols_on_mouth:': '\U0001f92c',
':face_with_tears_of_joy:': '😂',
':face_with_thermometer:': '🤒',
':face_with_tongue:': '😛',
':face_without_mouth:': '😶',
':factory:': '🏭',
':fairy:': '\U0001f9da',
':fairy_dark_skin_tone:': '\U0001f9da🏾',
':fairy_light_skin_tone:': '\U0001f9da🏻',
':fairy_medium-dark_skin_tone:': '\U0001f9da🏾',
':fairy_medium-light_skin_tone:': '\U0001f9da🏼',
```

```
':fairy_medium_skin_tone:': '\U0001f9da▣',
':falafel:': '\U0001f9c6',
':fallen_leaf:': '🍂',
':family:': '👪',
':family_man_boy:': '👨\u200d👦',
':family_man_boy_boy:': '👨\u200d👦\u200d👦',
':family_man_girl:': '👨\u200d👧',
':family_man_girl_boy:': '👨\u200d👧\u200d👦',
':family_man_girl_girl:': '👨\u200d👧\u200d👧',
':family_man_man_boy:': '👨\u200d👨\u200d👦',
':family_man_man_boy_boy:': '👨\u200d👨\u200d👦\u200d👦',
':family_man_man_girl:': '👨\u200d👨\u200d👧',
':family_man_man_girl_boy:': '👨\u200d👨\u200d👧\u200d👦',
':family_man_man_girl_girl:': '👨\u200d👨\u200d👧\u200d👧',
':family_man_woman_boy:': '👨\u200d👩\u200d👦',
':family_man_woman_boy_boy:': '👨\u200d👩\u200d👦\u200d👦',
':family_man_woman_girl:': '👨\u200d👩\u200d👧',
':family_man_woman_girl_boy:': '👨\u200d👩\u200d👧\u200d👦',
':family_man_woman_girl_girl:': '👨\u200d👩\u200d👧\u200d👧',
':family_woman_boy:': '👩\u200d👦',
':family_woman_boy_boy:': '👩\u200d👦\u200d👦',
':family_woman_girl:': '👩\u200d👧',
':family_woman_girl_boy:': '👩\u200d👧\u200d👦',
':family_woman_girl_girl:': '👩\u200d👧\u200d👧',
':family_woman_woman_boy:': '👩\u200d👩\u200d👦',
':family_woman_woman_boy_boy:': '👩\u200d👩\u200d👦\u200d👦',
':family_woman_woman_girl:': '👩\u200d👩\u200d👧',
':family_woman_woman_girl_boy:': '👩\u200d👩\u200d👧\u200d👦',
':family_woman_woman_girl_girl:': '👩\u200d👩\u200d👧\u200d👧',
':fast-forward_button:': '⏩',
':fast_down_button:': '⏬',
':fast_reverse_button:': '⏪',
':fast_up_button:': '⏫',
':fax_machine:': '📠',
':fearful_face:': '😨',
':female_sign:': '♀',
':ferris_wheel:': '🎡',
':ferry:': '⛴',
':field_hockey:': '🏑',
```

```
':file_cabinet:': '🗄',
':file_folder:': '📁',
':film_frames:': '🎞',
':film_projector:': '📽',
':fire:': '🔥',
':fire_extinguisher:': '\U0001f9ef',
':firecracker:': '\U0001f9e8',
':fire_engine:': '🚒',
':fireworks:': '🎆',
':first_quarter_moon:': '🌓',
':first_quarter_moon_face:': '🌛',
':fish:': '🐟',
':fish_cake_with_swirl:': '🍥',
':fishing_pole:': '🎣',
':five-thirty:': '🕠',
':five_o’clock:': '🕔',
':flag_in_hole:': '⛳',
':flamingo:': '\U0001f9a9',
':flashlight:': '🔦',
':flat_shoe:': '\U0001f97f',
':fleur-de-lis:': '⚜',
':flexed_biceps:': '💪',
':flexed_biceps_dark_skin_tone:': '💪🏿',
':flexed_biceps_light_skin_tone:': '💪🏻',
':flexed_biceps_medium-dark_skin_tone:': '💪🏾',
':flexed_biceps_medium-light_skin_tone:': '💪🏼',
':flexed_biceps_medium_skin_tone:': '💪🏽',
':floppy_disk:': '💾',
':flower_playing_cards:': '🎴',
':flushed_face:': '😳',
':flying_disc:': '\U0001f94f',
':flying_saucer:': '\U0001f6f8',
':fog:': '🌫',
':foggy:': '🌁',
':folded_hands:': '🙏',
':folded_hands_dark_skin_tone:': '🙏🏿',
':folded_hands_light_skin_tone:': '🙏🏻',
':folded_hands_medium-dark_skin_tone:': '🙏🏾',
':folded_hands_medium-light_skin_tone:': '🙏🏼',
```

```
':folded_hands_medium_skin_tone:': '🙏🏽',
':foot:': '\U0001f9b6',
':footprints:': '👣',
':fork_and_knife:': '🍴',
':fork_and_knife_with_plate:': '🍽',
':fortune_cookie:': '\U0001f960',
':fountain:': '⛲',
':fountain_pen:': '🖋',
':four-thirty:': '🕟',
':four_leaf_clover:': '🍀',
':four_o’clock:': '🕓',
':fox_face:': '🦊',
':framed_picture:': '🖼',
':french_fries:': '🍟',
':fried_shrimp:': '🍤',
':frog_face:': '🐸',
':front-facing_baby_chick:': '🐥',
':frowning_face:': '☹',
':frowning_face_with_open_mouth:': '😦',
':fuel_pump:': '⛽',
':full_moon:': '🌕',
':full_moon_face:': '🌝',
':funeral_urn:': '⚱',
':game_die:': '🎲',
':garlic:': '\U0001f9c4',
':gear:': '⚙',
':gem_stone:': '💎',
':genie:': '\U0001f9de',
':ghost:': '👻',
':giraffe:': '\U0001f992',
':girl:': '👧',
':girl_dark_skin_tone:': '👧🏿',
':girl_light_skin_tone:': '👧🏻',
':girl_medium-dark_skin_tone:': '👧🏾',
':girl_medium-light_skin_tone:': '👧🏼',
':girl_medium_skin_tone:': '👧🏽',
':glass_of_milk:': '🥛',
':glasses:': '👓',
':globe_showing_Americas:': '🌎',
```

```
':globe_showing_Asia-Australia:': '🌏',
':globe_showing_Europe-Africa:': '🌍',
':globe_with_meridians:': '🌐',
':gloves:': '\U0001f9e4',
':glowing_star:': '🌟',
':goal_net:': '🥅',
':goat:': '🐐',
':goblin:': '👺',
':goggles:': '\U0001f97d',
':gorilla:': '🦍',
':graduation_cap:': '🎓',
':grapes:': '🍇',
':green_apple:': '🍏',
':green_book:': '📗',
':green_circle:': '\U0001f7e2',
':green_heart:': '💚',
':green_salad:': '🥗',
':green_square:': '\U0001f7e9',
':grimacing_face:': '😬',
':grinning_cat_face:': '😺',
':grinning_cat_face_with_smiling_eyes:': '😸',
':grinning_face:': '😀',
':grinning_face_with_big_eyes:': '😃',
':grinning_face_with_smiling_eyes:': '😄',
':grinning_face_with_sweat:': '😅',
':grinning_squinting_face:': '😆',
':growing_heart:': '💗',
':guard:': '💂',
':guard_dark_skin_tone:': '💂🏿',
':guard_light_skin_tone:': '💂🏻',
':guard_medium-dark_skin_tone:': '💂🏾',
':guard_medium-light_skin_tone:': '💂🏼',
':guard_medium_skin_tone:': '💂🏽',
':guide_dog:': '\U0001f9ae',
':guitar:': '🎸',
':hamburger:': '🍔',
':hammer:': '🔨',
':hammer_and_pick:': '⚒',
':hammer_and_wrench:': '🛠',
```

```
':hamster_face:': '🐹',
':hand_with_fingers_splayed:': '🖐',
':hand_with_fingers_splayed_dark_skin_tone:': '🖐🏿',
':hand_with_fingers_splayed_light_skin_tone:': '🖐🏻',
':hand_with_fingers_splayed_medium-dark_skin_tone:': '🖐🏾',
':hand_with_fingers_splayed_medium-light_skin_tone:': '🖐🏼',
':hand_with_fingers_splayed_medium_skin_tone:': '🖐🏽',
':handbag:': '👜',
':handshake:': '🤝',
':hatching_chick:': '🐣',
':headphone:': '🎧',
':hear-no-evil_monkey:': '🙉',
':heart_decoration:': '💟',
':heart_suit:': '♥',
':heart_with_arrow:': '💘',
':heart_with_ribbon:': '💝',
':heavy_check_mark:': '✔',
':heavy_division_sign:': '➗',
':heavy_dollar_sign:': '💲',
':heavy_heart_exclamation:': '❣',
':heavy_large_circle:': '⭕',
':heavy_minus_sign:': '➖',
':heavy_multiplication_x:': '✖',
':heavy_plus_sign:': '➕',
':hedgehog:': '\U0001f994',
':helicopter:': '🚁',
':herb:': '🌿',
':hibiscus:': '🌺',
...}
```

In [4]:
```
# To print these emojis we will use .emojize() method and giving emoji description as in
put, example:
print(emoji.emojize(":angry_face_with_horns:"))
```

👿

In [4]:
```
# For our project we will create a dictionary of 5 emojis that our dataset contains
```

```
emoji_dictionary = {
    "0": "\u2764\uFE0F",
    "1": ":baseball:",
    "2": ":grinning_face_with_big_eyes:",
    "3": ":disappointed_face:",
    "4": ":fork_and_knife:"
}

for e in emoji_dictionary.values():
    print(emoji.emojize(e))
```

❤
⚾
😄
😞
🍴

## 2) Processing a Custom Dataset

- We will process custom emoji dataset with 132 sentences and an emoji associated with each one of the sentences

In [6]:
```
data_train = pd.read_csv("Datasets/train_emoji.csv", header=None)
print(data_train.head(5))
```

```
                              0  1   2     3
0          never talk to me again  3 NaN   NaN
1  I am proud of your achievements  2 NaN   NaN
2   It is the worst day in my life  3 NaN   NaN
3                Miss you so much  0 NaN   [0]
4                    food is life  4 NaN   NaN
```

In [8]:
```
# As we can see we have to use only columns 1 & 2 in our dataset
```

```
X_train = data_train.values[:,0]
Y_train = data_train.values[:,1]

print(X_train.shape, Y_train.shape)
print(X_train[:5])
print(Y_train[:5])
```

```
(132,) (132,)
['never talk to me again' 'I am proud of your achievements'
 'It is the worst day in my life' 'Miss you so much' 'food is life']
[3 2 3 0 4]
```

In [9]:
```
# Seeing our dataset with the actual emojis
for i in range(5):
    print(X_train[i], emoji.emojize(emoji_dictionary[str(Y_train[i])]))
```

```
never talk to me again 😞
I am proud of your achievements 😄
It is the worst day in my life 😞
Miss you so much ❤
food is life 🍴
```

## 3) Importing and working with Glove Vectors

In [10]:
```
from gensim.models import KeyedVectors
```

In [14]:
```
word_vec = KeyedVectors.load_word2vec_format("Datasets/word2vec_from_glove.6B.50d.txt")
print(len(word_vec.vocab))
print(len(word_vec["the"]))
print(word_vec["the"])

# This pre-processed glove vector dataset contains 4,00,000 words in lower case and each
```

```
 vector of
# dimention 50 associated with each word
```

```
400000
50
[ 4.1800e-01  2.4968e-01 -4.1242e-01  1.2170e-01  3.4527e-01 -4.4457e-02
 -4.9688e-01 -1.7862e-01 -6.6023e-04 -6.5660e-01  2.7843e-01 -1.4767e-01
 -5.5677e-01  1.4658e-01 -9.5095e-03  1.1658e-02  1.0204e-01 -1.2792e-01
 -8.4430e-01 -1.2181e-01 -1.6801e-02 -3.3279e-01 -1.5520e-01 -2.3131e-01
 -1.9181e-01 -1.8823e+00 -7.6746e-01  9.9051e-02 -4.2125e-01 -1.9526e-01
  4.0071e+00 -1.8594e-01 -5.2287e-01 -3.1681e-01  5.9213e-04  7.4449e-03
  1.7778e-01 -1.5897e-01  1.2041e-02 -5.4223e-02 -2.9871e-01 -1.5749e-01
 -3.4758e-01 -4.5637e-02 -4.4251e-01  1.8785e-01  2.7849e-03 -1.8411e-01
 -1.1514e-01 -7.8581e-01]
```

## 4) Forming an Embedding Matrix

- We will form an **embedding matrix** that will be a **3D tensor of shape ==> (no_of_examples, maxlength_of_each_sentence, 50)**.
- Will set the max length equal to 10 as in this dataset sentences are very small.
- Last dimention of the embedding matrix is 50 because it represents the length of each vector of that corresponding word

In [15]:
```
def getOutputEmbeddings(X):

    emb_matrix = np.zeros((X.shape[0], 10, 50))
    for sentences in range(X.shape[0]): # iterating over each sentence
        words = X[sentences].split() # Breaking each sentence into respective words
        for word in range(len(words)):
            emb_matrix[sentences][word] = word_vec[words[word].lower()]

    return emb_matrix
```

In [16]:
```
embedding_matrix = getOutputEmbeddings(X_train)
print(embedding_matrix.shape)

# We converted each sentence into a matrix of 10 words with each being represented
# by a vector of dimension 50
```

```
(132, 10, 50)
```

## 5) Defining the RNN/LSTM Model

In this model we will use just a simple LSTM layer adn the activation state vector(a) from that will be passed on to the final dense layer with 5 outputs and a softmax activation.
 simple lstm

In [17]:
```
from keras.models import Sequential
from keras.layers import Dense, LSTM, Dropout, Activation
```

```
Using TensorFlow backend.
```

In [83]:
```
model = Sequential()
model.add(LSTM(64, input_shape=(10,50))) # Dimention of activation vector: 64
model.add(Dropout(0.5))
model.add(Dense(5))
model.add(Activation("softmax"))
# At the end a dense layer with softmax activation is added so we can get a
# probability for each output class/emoji
model.compile(loss="categorical_crossentropy", optimizer="adam", metrics=["accuracy"])
model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
```

```
=================================================================
lstm_20 (LSTM)               (None, 64)                29440
_________________________________________________________________
dropout_20 (Dropout)         (None, 64)                0
_________________________________________________________________
dense_20 (Dense)             (None, 5)                 325
_________________________________________________________________
activation_20 (Activation)   (None, 5)                 0
=================================================================
Total params: 29,765
Trainable params: 29,765
Non-trainable params: 0
_________________________________________________________________
```

In [19]:

```
# Before training we have to convert our Y_train values into one hot vectors
# value:1 for corresponding class and value:0 for rest classes

from keras.utils import to_categorical

print(Y_train.shape)
print(Y_train[:5])

Y_train = to_categorical(Y_train, num_classes=5)
print(Y_train.shape)
print(Y_train[:5])
```

```
(132,)
[3 2 3 0 4]
(132, 5)
[[0. 0. 0. 1. 0.]
 [0. 0. 1. 0. 0.]
 [0. 0. 0. 1. 0.]
 [1. 0. 0. 0. 0.]
 [0. 0. 0. 0. 1.]]
```

In [20]:
```
# Training Model
hist = model.fit(embedding_matrix, Y_train, epochs=100, batch_size=64, shuffle=True, val
idation_split=0.2)
```

```
WARNING:tensorflow:From C:\ProgramData\Anaconda3\lib\site-packages\tensorflow\python\ops\math_grad.py:1250: add_dispatch_sup
port.<locals>.wrapper (from tensorflow.python.ops.array_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
Train on 105 samples, validate on 27 samples
Epoch 1/100
105/105 [==============================] - 1s 8ms/step - loss: 1.6280 - acc: 0.1619 - val_loss: 1.5994 - val_acc: 0.2222
Epoch 2/100
105/105 [==============================] - 0s 199us/step - loss: 1.5896 - acc: 0.3238 - val_loss: 1.6030 - val_acc: 0.2222
Epoch 3/100
105/105 [==============================] - 0s 180us/step - loss: 1.5589 - acc: 0.3143 - val_loss: 1.6087 - val_acc: 0.2222
Epoch 4/100
105/105 [==============================] - 0s 190us/step - loss: 1.5589 - acc: 0.3143 - val_loss: 1.6178 - val_acc: 0.2593
Epoch 5/100
105/105 [==============================] - 0s 180us/step - loss: 1.5143 - acc: 0.4000 - val_loss: 1.6289 - val_acc: 0.2963
Epoch 6/100
105/105 [==============================] - 0s 180us/step - loss: 1.5049 - acc: 0.3619 - val_loss: 1.6411 - val_acc: 0.2963
Epoch 7/100
105/105 [==============================] - 0s 161us/step - loss: 1.4896 - acc: 0.3238 - val_loss: 1.6543 - val_acc: 0.2593
Epoch 8/100
105/105 [==============================] - 0s 161us/step - loss: 1.4838 - acc: 0.3524 - val_loss: 1.6644 - val_acc: 0.2593
Epoch 9/100
105/105 [==============================] - 0s 152us/step - loss: 1.4544 - acc: 0.3810 - val_loss: 1.6726 - val_acc: 0.2593
Epoch 10/100
105/105 [==============================] - 0s 142us/step - loss: 1.4768 - acc: 0.3429 - val_loss: 1.6720 - val_acc: 0.2963
Epoch 11/100
105/105 [==============================] - 0s 142us/step - loss: 1.4526 - acc: 0.3714 - val_loss: 1.6659 - val_acc: 0.2593
Epoch 12/100
105/105 [==============================] - 0s 161us/step - loss: 1.4438 - acc: 0.4286 - val_loss: 1.6527 - val_acc: 0.3333
Epoch 13/100
105/105 [==============================] - 0s 142us/step - loss: 1.4103 - acc: 0.4095 - val_loss: 1.6323 - val_acc: 0.2593
Epoch 14/100
105/105 [==============================] - 0s 142us/step - loss: 1.3929 - acc: 0.4095 - val_loss: 1.6093 - val_acc: 0.2593

Epoch 15/100
```

```
105/105 [==============================] - 0s 161us/step - loss: 1.3651 - acc: 0.4476 - val_loss: 1.5787 - val_acc: 0.2593
Epoch 16/100
105/105 [==============================] - 0s 152us/step - loss: 1.3618 - acc: 0.4476 - val_loss: 1.5468 - val_acc: 0.2963
Epoch 17/100
105/105 [==============================] - 0s 142us/step - loss: 1.3304 - acc: 0.4286 - val_loss: 1.5143 - val_acc: 0.3333
Epoch 18/100
105/105 [==============================] - 0s 171us/step - loss: 1.3235 - acc: 0.4571 - val_loss: 1.4809 - val_acc: 0.2963
Epoch 19/100
105/105 [==============================] - 0s 142us/step - loss: 1.2541 - acc: 0.5238 - val_loss: 1.4445 - val_acc: 0.2963
Epoch 20/100
105/105 [==============================] - 0s 142us/step - loss: 1.2086 - acc: 0.5905 - val_loss: 1.4099 - val_acc: 0.2963
Epoch 21/100
105/105 [==============================] - 0s 142us/step - loss: 1.2039 - acc: 0.5524 - val_loss: 1.3765 - val_acc: 0.2963
Epoch 22/100
105/105 [==============================] - 0s 142us/step - loss: 1.1500 - acc: 0.5619 - val_loss: 1.3454 - val_acc: 0.3704
Epoch 23/100
105/105 [==============================] - 0s 142us/step - loss: 1.0668 - acc: 0.6000 - val_loss: 1.3191 - val_acc: 0.4074
Epoch 24/100
105/105 [==============================] - 0s 152us/step - loss: 1.0018 - acc: 0.6381 - val_loss: 1.2800 - val_acc: 0.4444
Epoch 25/100
105/105 [==============================] - 0s 152us/step - loss: 1.0076 - acc: 0.6571 - val_loss: 1.2223 - val_acc: 0.4444
Epoch 26/100
105/105 [==============================] - 0s 142us/step - loss: 0.9431 - acc: 0.6571 - val_loss: 1.1453 - val_acc: 0.5185
Epoch 27/100
105/105 [==============================] - 0s 142us/step - loss: 0.9021 - acc: 0.7238 - val_loss: 1.0852 - val_acc: 0.5556
Epoch 28/100
105/105 [==============================] - 0s 142us/step - loss: 0.8677 - acc: 0.7714 - val_loss: 1.0526 - val_acc: 0.5556
Epoch 29/100
105/105 [==============================] - 0s 152us/step - loss: 0.7815 - acc: 0.7810 - val_loss: 1.0459 - val_acc: 0.5556
Epoch 30/100
105/105 [==============================] - 0s 152us/step - loss: 0.7861 - acc: 0.7429 - val_loss: 0.9758 - val_acc: 0.5926
Epoch 31/100
105/105 [==============================] - 0s 142us/step - loss: 0.7088 - acc: 0.7810 - val_loss: 0.9329 - val_acc: 0.7037
Epoch 32/100
105/105 [==============================] - 0s 161us/step - loss: 0.7137 - acc: 0.7810 - val_loss: 0.8983 - val_acc: 0.7037
Epoch 33/100
105/105 [==============================] - 0s 152us/step - loss: 0.6287 - acc: 0.8000 - val_loss: 0.8644 - val_acc: 0.7407
Epoch 34/100

105/105 [==============================] - 0s 152us/step - loss: 0.5801 - acc: 0.8381 - val_loss: 0.8259 - val_acc: 0.7407
```

```
Epoch 35/100
105/105 [==============================] - 0s 323us/step - loss: 0.5150 - acc: 0.8857 - val_loss: 0.8140 - val_acc: 0.7037
Epoch 36/100
105/105 [==============================] - 0s 161us/step - loss: 0.4783 - acc: 0.8762 - val_loss: 0.8248 - val_acc: 0.7407
Epoch 37/100
105/105 [==============================] - 0s 142us/step - loss: 0.4533 - acc: 0.9143 - val_loss: 0.8744 - val_acc: 0.7407
Epoch 38/100
105/105 [==============================] - 0s 142us/step - loss: 0.3804 - acc: 0.9048 - val_loss: 0.8792 - val_acc: 0.7037
Epoch 39/100
105/105 [==============================] - 0s 152us/step - loss: 0.3724 - acc: 0.9238 - val_loss: 0.8389 - val_acc: 0.7407
Epoch 40/100
105/105 [==============================] - 0s 171us/step - loss: 0.3705 - acc: 0.9143 - val_loss: 0.8408 - val_acc: 0.6667
Epoch 41/100
105/105 [==============================] - 0s 152us/step - loss: 0.3866 - acc: 0.8952 - val_loss: 0.8028 - val_acc: 0.7037
Epoch 42/100
105/105 [==============================] - 0s 142us/step - loss: 0.4014 - acc: 0.8857 - val_loss: 0.9143 - val_acc: 0.6296
Epoch 43/100
105/105 [==============================] - 0s 142us/step - loss: 0.3894 - acc: 0.8762 - val_loss: 0.9707 - val_acc: 0.6667
Epoch 44/100
105/105 [==============================] - 0s 142us/step - loss: 0.2939 - acc: 0.9143 - val_loss: 1.0169 - val_acc: 0.6296
Epoch 45/100
105/105 [==============================] - ETA: 0s - loss: 0.2795 - acc: 0.937 - 0s 142us/step - loss: 0.2873 - acc: 0.9238
- val_loss: 0.8967 - val_acc: 0.6296
Epoch 46/100
105/105 [==============================] - 0s 152us/step - loss: 0.2349 - acc: 0.9429 - val_loss: 0.9002 - val_acc: 0.6296
Epoch 47/100
105/105 [==============================] - 0s 142us/step - loss: 0.3134 - acc: 0.9238 - val_loss: 0.8712 - val_acc: 0.7037
Epoch 48/100
105/105 [==============================] - 0s 152us/step - loss: 0.2441 - acc: 0.9238 - val_loss: 0.9036 - val_acc: 0.7407
Epoch 49/100
105/105 [==============================] - 0s 152us/step - loss: 0.2354 - acc: 0.9333 - val_loss: 0.9209 - val_acc: 0.7407
Epoch 50/100
105/105 [==============================] - 0s 152us/step - loss: 0.1989 - acc: 0.9524 - val_loss: 0.9612 - val_acc: 0.7407
Epoch 51/100
105/105 [==============================] - 0s 171us/step - loss: 0.1547 - acc: 0.9810 - val_loss: 0.9095 - val_acc: 0.6667
Epoch 52/100
105/105 [==============================] - 0s 171us/step - loss: 0.1342 - acc: 0.9714 - val_loss: 0.8740 - val_acc: 0.6667
Epoch 53/100

105/105 [==============================] - 0s 171us/step - loss: 0.1475 - acc: 0.9714 - val_loss: 0.8942 - val_acc: 0.6667
```

```
Epoch 54/100
105/105 [==============================] - 0s 152us/step - loss: 0.1300 - acc: 0.9810 - val_loss: 0.9443 - val_acc: 0.6667
Epoch 55/100
105/105 [==============================] - 0s 161us/step - loss: 0.1269 - acc: 0.9810 - val_loss: 0.9333 - val_acc: 0.6667
Epoch 56/100
105/105 [==============================] - 0s 161us/step - loss: 0.1309 - acc: 0.9714 - val_loss: 0.9427 - val_acc: 0.6667
Epoch 57/100
105/105 [==============================] - 0s 152us/step - loss: 0.1243 - acc: 0.9619 - val_loss: 0.9881 - val_acc: 0.7037
Epoch 58/100
105/105 [==============================] - 0s 152us/step - loss: 0.0904 - acc: 0.9905 - val_loss: 1.0051 - val_acc: 0.6296
Epoch 59/100
105/105 [==============================] - 0s 161us/step - loss: 0.1011 - acc: 0.9714 - val_loss: 0.9602 - val_acc: 0.6667
Epoch 60/100
105/105 [==============================] - 0s 152us/step - loss: 0.0896 - acc: 0.9905 - val_loss: 0.9389 - val_acc: 0.6667
Epoch 61/100
105/105 [==============================] - 0s 161us/step - loss: 0.0767 - acc: 0.9905 - val_loss: 0.9315 - val_acc: 0.6667
Epoch 62/100
105/105 [==============================] - 0s 161us/step - loss: 0.0651 - acc: 1.0000 - val_loss: 0.9536 - val_acc: 0.6667
Epoch 63/100
105/105 [==============================] - 0s 161us/step - loss: 0.0551 - acc: 1.0000 - val_loss: 1.0133 - val_acc: 0.7037
Epoch 64/100
105/105 [==============================] - 0s 142us/step - loss: 0.0744 - acc: 0.9905 - val_loss: 1.0569 - val_acc: 0.7037
Epoch 65/100
105/105 [==============================] - 0s 142us/step - loss: 0.0617 - acc: 1.0000 - val_loss: 1.2290 - val_acc: 0.6296
Epoch 66/100
105/105 [==============================] - 0s 152us/step - loss: 0.1050 - acc: 0.9714 - val_loss: 1.0677 - val_acc: 0.6296
Epoch 67/100
105/105 [==============================] - 0s 142us/step - loss: 0.0572 - acc: 1.0000 - val_loss: 1.1669 - val_acc: 0.6296
Epoch 68/100
105/105 [==============================] - 0s 152us/step - loss: 0.1040 - acc: 0.9524 - val_loss: 1.1712 - val_acc: 0.6296
Epoch 69/100
105/105 [==============================] - 0s 142us/step - loss: 0.0591 - acc: 1.0000 - val_loss: 1.2653 - val_acc: 0.6296
Epoch 70/100
105/105 [==============================] - 0s 142us/step - loss: 0.0711 - acc: 0.9905 - val_loss: 1.3048 - val_acc: 0.6296
Epoch 71/100
105/105 [==============================] - 0s 142us/step - loss: 0.0427 - acc: 1.0000 - val_loss: 1.2422 - val_acc: 0.6296
Epoch 72/100
105/105 [==============================] - 0s 152us/step - loss: 0.0497 - acc: 1.0000 - val_loss: 1.1901 - val_acc: 0.6296

Epoch 73/100
```

```
105/105 [==============================] - 0s 142us/step - loss: 0.0457 - acc: 1.0000 - val_loss: 1.1776 - val_acc: 0.5926
Epoch 74/100
105/105 [==============================] - 0s 142us/step - loss: 0.0444 - acc: 1.0000 - val_loss: 1.2474 - val_acc: 0.5926
Epoch 75/100
105/105 [==============================] - 0s 152us/step - loss: 0.0426 - acc: 0.9905 - val_loss: 1.3150 - val_acc: 0.5926
Epoch 76/100
105/105 [==============================] - 0s 142us/step - loss: 0.0335 - acc: 1.0000 - val_loss: 1.3330 - val_acc: 0.5926
Epoch 77/100
105/105 [==============================] - 0s 152us/step - loss: 0.0330 - acc: 1.0000 - val_loss: 1.3156 - val_acc: 0.5926
Epoch 78/100
105/105 [==============================] - 0s 142us/step - loss: 0.0318 - acc: 1.0000 - val_loss: 1.2885 - val_acc: 0.6296
Epoch 79/100
105/105 [==============================] - 0s 152us/step - loss: 0.0330 - acc: 1.0000 - val_loss: 1.2806 - val_acc: 0.6667
Epoch 80/100
105/105 [==============================] - 0s 142us/step - loss: 0.0204 - acc: 1.0000 - val_loss: 1.2771 - val_acc: 0.6667
Epoch 81/100
105/105 [==============================] - 0s 142us/step - loss: 0.0293 - acc: 1.0000 - val_loss: 1.2888 - val_acc: 0.6667
Epoch 82/100
105/105 [==============================] - 0s 133us/step - loss: 0.0323 - acc: 0.9905 - val_loss: 1.2969 - val_acc: 0.5926
Epoch 83/100
105/105 [==============================] - 0s 142us/step - loss: 0.0288 - acc: 1.0000 - val_loss: 1.3529 - val_acc: 0.5926
Epoch 84/100
105/105 [==============================] - 0s 142us/step - loss: 0.0339 - acc: 1.0000 - val_loss: 1.4229 - val_acc: 0.5556
Epoch 85/100
105/105 [==============================] - 0s 142us/step - loss: 0.0287 - acc: 1.0000 - val_loss: 1.3321 - val_acc: 0.5926
Epoch 86/100
105/105 [==============================] - 0s 142us/step - loss: 0.0267 - acc: 1.0000 - val_loss: 1.2564 - val_acc: 0.5926
Epoch 87/100
105/105 [==============================] - 0s 142us/step - loss: 0.0262 - acc: 1.0000 - val_loss: 1.2108 - val_acc: 0.6667
Epoch 88/100
105/105 [==============================] - 0s 152us/step - loss: 0.0222 - acc: 1.0000 - val_loss: 1.2070 - val_acc: 0.6667
Epoch 89/100
105/105 [==============================] - 0s 142us/step - loss: 0.0211 - acc: 1.0000 - val_loss: 1.2214 - val_acc: 0.6667
Epoch 90/100
105/105 [==============================] - 0s 152us/step - loss: 0.0460 - acc: 0.9810 - val_loss: 1.2504 - val_acc: 0.6667
Epoch 91/100
105/105 [==============================] - 0s 152us/step - loss: 0.0195 - acc: 1.0000 - val_loss: 1.2833 - val_acc: 0.6296
Epoch 92/100

105/105 [==============================] - 0s 142us/step - loss: 0.0198 - acc: 1.0000 - val_loss: 1.3449 - val_acc: 0.6296
```

```
Epoch 93/100
105/105 [==============================] - 0s 142us/step - loss: 0.0223 - acc: 1.0000 - val_loss: 1.3971 - val_acc: 0.6667
Epoch 94/100
105/105 [==============================] - 0s 142us/step - loss: 0.0192 - acc: 1.0000 - val_loss: 1.4412 - val_acc: 0.6667
Epoch 95/100
105/105 [==============================] - 0s 142us/step - loss: 0.0165 - acc: 1.0000 - val_loss: 1.4640 - val_acc: 0.6296
Epoch 96/100
105/105 [==============================] - 0s 142us/step - loss: 0.0204 - acc: 1.0000 - val_loss: 1.4802 - val_acc: 0.6296
Epoch 97/100
105/105 [==============================] - 0s 152us/step - loss: 0.0221 - acc: 1.0000 - val_loss: 1.4704 - val_acc: 0.5926
Epoch 98/100
105/105 [==============================] - 0s 142us/step - loss: 0.0178 - acc: 1.0000 - val_loss: 1.4790 - val_acc: 0.6296
Epoch 99/100
105/105 [==============================] - 0s 161us/step - loss: 0.0294 - acc: 1.0000 - val_loss: 1.5021 - val_acc: 0.6296
Epoch 100/100
105/105 [==============================] - 0s 142us/step - loss: 0.0187 - acc: 1.0000 - val_loss: 1.4972 - val_acc: 0.5926
```

## 6) Processing Test dataset for Model Evaluation

In [23]:
```
data_test = pd.read_csv("Datasets/test_emoji.csv", header=None)
print(data_test.head(5))
```

```
                          0  1
0            I want to eat\t  4
1        he did not answer\t  3
2           he got a raise\t  2
3     she got me a present\t  0
4  ha ha ha it was so funny\t  2
```

In [25]:
```
X_test = data_test.values[:,0]
Y_test = data_test.values[:,1]

print(X_test.shape, Y_test.shape)
```

```
# Converting X_test ==> Embedding matrix || Y_test ==> Categorical(Y_test)
embedding_matrix_test = getOutputEmbeddings(X_test)
Y_test = to_categorical(Y_test, num_classes=5)
print(embedding_matrix_test.shape, Y_test.shape)
```

```
(56,) (56,)
(56, 10, 50) (56, 5)
```

In [26]:

```
print(model.metrics_names)
print(model.evaluate(embedding_matrix_test, Y_test))

# As we can see we have an accuracy of 57% but we can improve this by including EarlySto
p callbacks and
# ModelCheckpoint
```

```
['loss', 'acc']
56/56 [==============================] - 0s 107us/step
[2.197503055844988, 0.5714285714285714]
```

In [27]:

```
from keras.callbacks import EarlyStopping, ModelCheckpoint
```

In [84]:

```
checkpoint = ModelCheckpoint("best_model.h5", verbose=1, save_best_only=True, monitor='v
al_loss', save_weights_only=True)
earlystop = EarlyStopping(monitor='val_acc', patience=20)
# We have given patience=20 becasue there is a lot of varience in the val_acc because ou
r
# dataset is very small.

hist2 = model.fit(embedding_matrix, Y_train, epochs=100, batch_size=64, shuffle=True, va
lidation_split=0.1, callbacks=[earlystop, checkpoint])
```

```
Train on 118 samples, validate on 14 samples
Epoch 1/100
118/118 [==============================] - 3s 24ms/step - loss: 1.6249 - acc: 0.1949 - val_loss: 1.6374 - val_acc: 0.1429

Epoch 00001: val_loss improved from inf to 1.63736, saving model to best_model.h5
Epoch 2/100
118/118 [==============================] - 0s 237us/step - loss: 1.5898 - acc: 0.3051 - val_loss: 1.6290 - val_acc: 0.0714

Epoch 00002: val_loss improved from 1.63736 to 1.62896, saving model to best_model.h5
Epoch 3/100
118/118 [==============================] - 0s 211us/step - loss: 1.5741 - acc: 0.3475 - val_loss: 1.6253 - val_acc: 0.0714

Epoch 00003: val_loss improved from 1.62896 to 1.62532, saving model to best_model.h5
Epoch 4/100
118/118 [==============================] - 0s 203us/step - loss: 1.5500 - acc: 0.3390 - val_loss: 1.6217 - val_acc: 0.2143

Epoch 00004: val_loss improved from 1.62532 to 1.62173, saving model to best_model.h5
Epoch 5/100
118/118 [==============================] - 0s 194us/step - loss: 1.5334 - acc: 0.3814 - val_loss: 1.6199 - val_acc: 0.2143

Epoch 00005: val_loss improved from 1.62173 to 1.61987, saving model to best_model.h5
Epoch 6/100
118/118 [==============================] - 0s 161us/step - loss: 1.5154 - acc: 0.3898 - val_loss: 1.6192 - val_acc: 0.2143

Epoch 00006: val_loss improved from 1.61987 to 1.61918, saving model to best_model.h5
Epoch 7/100
118/118 [==============================] - 0s 161us/step - loss: 1.5078 - acc: 0.3898 - val_loss: 1.6163 - val_acc: 0.3571

Epoch 00007: val_loss improved from 1.61918 to 1.61625, saving model to best_model.h5
Epoch 8/100
118/118 [==============================] - 0s 169us/step - loss: 1.4951 - acc: 0.4237 - val_loss: 1.6101 - val_acc: 0.3571

Epoch 00008: val_loss improved from 1.61625 to 1.61008, saving model to best_model.h5
Epoch 9/100
118/118 [==============================] - 0s 177us/step - loss: 1.4783 - acc: 0.3729 - val_loss: 1.6027 - val_acc: 0.3571

Epoch 00009: val_loss improved from 1.61008 to 1.60269, saving model to best_model.h5
Epoch 10/100
118/118 [==============================] - 0s 152us/step - loss: 1.4503 - acc: 0.3814 - val_loss: 1.5916 - val_acc: 0.3571
```

```
Epoch 00010: val_loss improved from 1.60269 to 1.59165, saving model to best_model.h5
Epoch 11/100
118/118 [==============================] - 0s 161us/step - loss: 1.4500 - acc: 0.3729 - val_loss: 1.5814 - val_acc: 0.3571

Epoch 00011: val_loss improved from 1.59165 to 1.58136, saving model to best_model.h5
Epoch 12/100
118/118 [==============================] - 0s 161us/step - loss: 1.4073 - acc: 0.4153 - val_loss: 1.5718 - val_acc: 0.3571

Epoch 00012: val_loss improved from 1.58136 to 1.57183, saving model to best_model.h5
Epoch 13/100
118/118 [==============================] - 0s 161us/step - loss: 1.3896 - acc: 0.4068 - val_loss: 1.5547 - val_acc: 0.3571

Epoch 00013: val_loss improved from 1.57183 to 1.55472, saving model to best_model.h5
Epoch 14/100
118/118 [==============================] - 0s 194us/step - loss: 1.3693 - acc: 0.4322 - val_loss: 1.5351 - val_acc: 0.3571

Epoch 00014: val_loss improved from 1.55472 to 1.53513, saving model to best_model.h5
Epoch 15/100
118/118 [==============================] - 0s 161us/step - loss: 1.3500 - acc: 0.4746 - val_loss: 1.5144 - val_acc: 0.2143

Epoch 00015: val_loss improved from 1.53513 to 1.51438, saving model to best_model.h5
Epoch 16/100
118/118 [==============================] - 0s 161us/step - loss: 1.2719 - acc: 0.5169 - val_loss: 1.4939 - val_acc: 0.3571

Epoch 00016: val_loss improved from 1.51438 to 1.49394, saving model to best_model.h5
Epoch 17/100
118/118 [==============================] - 0s 152us/step - loss: 1.2506 - acc: 0.5339 - val_loss: 1.4619 - val_acc: 0.2857

Epoch 00017: val_loss improved from 1.49394 to 1.46195, saving model to best_model.h5
Epoch 18/100
118/118 [==============================] - 0s 169us/step - loss: 1.2286 - acc: 0.5424 - val_loss: 1.4267 - val_acc: 0.3571

Epoch 00018: val_loss improved from 1.46195 to 1.42671, saving model to best_model.h5
Epoch 19/100
118/118 [==============================] - 0s 161us/step - loss: 1.1781 - acc: 0.5678 - val_loss: 1.3795 - val_acc: 0.4286

Epoch 00019: val_loss improved from 1.42671 to 1.37947, saving model to best_model.h5
Epoch 20/100
```

```
118/118 [==============================] - 0s 186us/step - loss: 1.0941 - acc: 0.5932 - val_loss: 1.3479 - val_acc: 0.4286

Epoch 00020: val_loss improved from 1.37947 to 1.34794, saving model to best_model.h5
Epoch 21/100
118/118 [==============================] - 0s 161us/step - loss: 1.0647 - acc: 0.5932 - val_loss: 1.3185 - val_acc: 0.4286

Epoch 00021: val_loss improved from 1.34794 to 1.31850, saving model to best_model.h5
Epoch 22/100
118/118 [==============================] - 0s 152us/step - loss: 1.0564 - acc: 0.6610 - val_loss: 1.2797 - val_acc: 0.5000

Epoch 00022: val_loss improved from 1.31850 to 1.27967, saving model to best_model.h5
Epoch 23/100
118/118 [==============================] - 0s 161us/step - loss: 1.0232 - acc: 0.6525 - val_loss: 1.2348 - val_acc: 0.5000

Epoch 00023: val_loss improved from 1.27967 to 1.23483, saving model to best_model.h5
Epoch 24/100
118/118 [==============================] - 0s 161us/step - loss: 0.9315 - acc: 0.6695 - val_loss: 1.2146 - val_acc: 0.5000

Epoch 00024: val_loss improved from 1.23483 to 1.21459, saving model to best_model.h5
Epoch 25/100
118/118 [==============================] - 0s 152us/step - loss: 0.8923 - acc: 0.7034 - val_loss: 1.1829 - val_acc: 0.5000

Epoch 00025: val_loss improved from 1.21459 to 1.18287, saving model to best_model.h5
Epoch 26/100
118/118 [==============================] - 0s 161us/step - loss: 0.8764 - acc: 0.6780 - val_loss: 1.1983 - val_acc: 0.5000

Epoch 00026: val_loss did not improve from 1.18287
Epoch 27/100
118/118 [==============================] - 0s 169us/step - loss: 0.7687 - acc: 0.7712 - val_loss: 1.1065 - val_acc: 0.5714

Epoch 00027: val_loss improved from 1.18287 to 1.10648, saving model to best_model.h5
Epoch 28/100
118/118 [==============================] - 0s 152us/step - loss: 0.7495 - acc: 0.7119 - val_loss: 1.0778 - val_acc: 0.6429

Epoch 00028: val_loss improved from 1.10648 to 1.07775, saving model to best_model.h5
Epoch 29/100
118/118 [==============================] - 0s 152us/step - loss: 0.6717 - acc: 0.7881 - val_loss: 1.0865 - val_acc: 0.6429

Epoch 00029: val_loss did not improve from 1.07775
```

```
Epoch 30/100
118/118 [==============================] - 0s 161us/step - loss: 0.7082 - acc: 0.7881 - val_loss: 1.0651 - val_acc: 0.5714

Epoch 00030: val_loss improved from 1.07775 to 1.06510, saving model to best_model.h5
Epoch 31/100
118/118 [==============================] - 0s 169us/step - loss: 0.6344 - acc: 0.8220 - val_loss: 1.0286 - val_acc: 0.5714

Epoch 00031: val_loss improved from 1.06510 to 1.02864, saving model to best_model.h5
Epoch 32/100
118/118 [==============================] - 0s 161us/step - loss: 0.5517 - acc: 0.8729 - val_loss: 0.9823 - val_acc: 0.5714

Epoch 00032: val_loss improved from 1.02864 to 0.98234, saving model to best_model.h5
Epoch 33/100
118/118 [==============================] - 0s 177us/step - loss: 0.5463 - acc: 0.8729 - val_loss: 1.0403 - val_acc: 0.5714

Epoch 00033: val_loss did not improve from 0.98234
Epoch 34/100
118/118 [==============================] - 0s 161us/step - loss: 0.4693 - acc: 0.8983 - val_loss: 1.1739 - val_acc: 0.5714

Epoch 00034: val_loss did not improve from 0.98234
Epoch 35/100
118/118 [==============================] - 0s 161us/step - loss: 0.4149 - acc: 0.8814 - val_loss: 1.0620 - val_acc: 0.5714

Epoch 00035: val_loss did not improve from 0.98234
Epoch 36/100
118/118 [==============================] - 0s 152us/step - loss: 0.4136 - acc: 0.9153 - val_loss: 0.9452 - val_acc: 0.5714

Epoch 00036: val_loss improved from 0.98234 to 0.94519, saving model to best_model.h5
Epoch 37/100
118/118 [==============================] - 0s 177us/step - loss: 0.3493 - acc: 0.8983 - val_loss: 1.0158 - val_acc: 0.5714

Epoch 00037: val_loss did not improve from 0.94519
Epoch 38/100
118/118 [==============================] - 0s 203us/step - loss: 0.3505 - acc: 0.9068 - val_loss: 1.1708 - val_acc: 0.5714

Epoch 00038: val_loss did not improve from 0.94519
Epoch 39/100
118/118 [==============================] - 0s 203us/step - loss: 0.3116 - acc: 0.9068 - val_loss: 1.0596 - val_acc: 0.5714
```

```
Epoch 00039: val_loss did not improve from 0.94519
Epoch 40/100
118/118 [==============================] - 0s 169us/step - loss: 0.2520 - acc: 0.9407 - val_loss: 1.1269 - val_acc: 0.5714

Epoch 00040: val_loss did not improve from 0.94519
Epoch 41/100
118/118 [==============================] - 0s 169us/step - loss: 0.2811 - acc: 0.9153 - val_loss: 1.2856 - val_acc: 0.5714

Epoch 00041: val_loss did not improve from 0.94519
Epoch 42/100
118/118 [==============================] - 0s 152us/step - loss: 0.2401 - acc: 0.9153 - val_loss: 1.2684 - val_acc: 0.5714

Epoch 00042: val_loss did not improve from 0.94519
Epoch 43/100
118/118 [==============================] - 0s 152us/step - loss: 0.2124 - acc: 0.9492 - val_loss: 1.1040 - val_acc: 0.6429

Epoch 00043: val_loss did not improve from 0.94519
Epoch 44/100
118/118 [==============================] - 0s 152us/step - loss: 0.3073 - acc: 0.8983 - val_loss: 1.4022 - val_acc: 0.5714

Epoch 00044: val_loss did not improve from 0.94519
Epoch 45/100
118/118 [==============================] - 0s 161us/step - loss: 0.2667 - acc: 0.8983 - val_loss: 1.1773 - val_acc: 0.5714

Epoch 00045: val_loss did not improve from 0.94519
Epoch 46/100
118/118 [==============================] - 0s 161us/step - loss: 0.2206 - acc: 0.9322 - val_loss: 0.9425 - val_acc: 0.5714

Epoch 00046: val_loss improved from 0.94519 to 0.94251, saving model to best_model.h5
Epoch 47/100
118/118 [==============================] - 0s 144us/step - loss: 0.2409 - acc: 0.9153 - val_loss: 1.0813 - val_acc: 0.6429

Epoch 00047: val_loss did not improve from 0.94251
Epoch 48/100
118/118 [==============================] - 0s 161us/step - loss: 0.1603 - acc: 0.9576 - val_loss: 1.6085 - val_acc: 0.5714

Epoch 00048: val_loss did not improve from 0.94251
```

In [86]:
```
# Testing accuracy of our new model with callbacks
```

```
model.load_weights("best_model.h5")
model.evaluate(embedding_matrix_test, Y_test)

# We can see that we have increased our testing accuracy by 11%
```

```
56/56 [==============================] - 0s 142us/step


[1.180537419659751, 0.6785714285714286]
```

# In the next notebook we will be using a Stacked LSTM Model to Predict Emojis

In [ ]:

In [ ]:

In [5]:
```
emoji.demojize("☺")
```

```
':slightly_smiling_face:'
```

In [ ]:

In [ ]:

In [ ]:

In [ ]: