**Language Model Creation**

Output of Problem1:

Accuracy of English test on English vs French Uni-gram models: 74.3 %

Accuracy of English test on English vs French bigram models: 50.0 %

Accuracy of English test on English vs French Tri-gram models: 48.8 %

Accuracy of Spanish test on Spanish vs Italian Uni-gram models: 49.1 %

Accuracy of Spanish test on Spanish vs Italian bi-gram models: 48.5 %

Accuracy of Spanish test on Spanish vs Italian tri-gram models: 63.5 %

**Language Model Comparison**

Based on the accuracy analysis, I would like to conclude that Spanish vs Italian is harder to distinguish than English vs French language model.

On the other hand, this would depend on lot of parameters like below and solution might vary.

How pre-processing was performed on the test mode.

How the stemming was done and what kind of stemmer used.

**Methodology**:

**Pre-processing** the text corpus

I have performed Pre-processing of text to enhance the accuracy of the identification task. I have considered the following pre-processing steps before creating language model….

All the texts are converted to lower case.

Punctuation marks and special characters are removed.

Series of contiguous white spaces are replaced by single space.

Uni-gram Language model:

To build this model, I have created dictionary **onecharfreqdict** which frequency of all the characters in the training corpus.

{'u': 19, 'n': 70, 'i': 57, 'v': 7, 'e': 107, 'r': 51, 's': 46, 'a': 73, 'l': 32, ' ': 166, 'd': 35, 'c': 18, 't': 66, 'o': 58, 'f': 29, 'h': 53, 'm': 26, 'g': 13, 'p': 17, 'b': 13, 'w': 13, 'y': 8, 'q': 1, 'j': 2, 'k': 1}

**Bi-gram Language model**

To build this model, I have created dictionary **bicharfreqdict** which frequency of all the characters in the training corpus.

eng\_bicharprob = freq[‘am’]/freq[‘a’]

eng\_bicharprob = round(bicharfreqdict[bichar],4)/round(onecharfreqdict[bichar[0]],4);

For eg:

({'e ': 31, 's ': 25, 'n ': 24, 'he': 24, ' t': 23, 'd ': 23, 'an': 22, 'th': 20, ' a': 20, 're': 19, ' o': 16, 'f ': 16, ' i': 16, 'nd': 15, 'of': 14, 'on': 13, ' h': 13, ' r': 13, 't ': 13, 'in': 12, 'en': 12, ' w': 11, ' f': 11, 'er': 10, 'ti': 10, 'as': 10, 'ed': 10, 'io': 9, 'ma': 9, 'ea': 9, 'al': 8, 'ig': 8, ' p': 8, 'nt': 8, 'be': 8, 'ha': 8, ' b': 8, 've': 7, 'l ': 7, 'at': 7, 'hu': 7, 'um': 7, 'wh': 7, 'it': 7, 'y ': 7, 'om': 7, 'te': 7, 'es': 7, 'ar': 6, 'ri': 6, 'gh': 6, 'ts': 6, 'le': 6, 'co': 6, 'ee': 6, 'pe': 6, 'ni': 5, ' d': 5, 'ec': 5, 'la': 5, 'ht': 5, 'pr': 5, 'me': 5, 'is': 5, 'ou': 5, 'fr': 5, 'or': 5, ' c': 5, 'ro': 5, 'h ': 5, 'ns': 5, 'el': 5, 'un': 4, 'rs': 4, 'ra': 4, ' e': 4, 'na': 4, 'ie': 4, 'll': 4, 'm ': 4, 'st': 4, 'r ': 4, 'av': 4, 'ch': 4, 'op': 4, 'to': 4, 'o ': 4, 'am': 3, 'gn': 3, 'di': 3, 'ty': 3, 'li': 3, ' m': 3, 'do': 3, 'ic': 3, 'ce': 3, 'wo': 3, 'ld': 3, 'ul': 3, 'hi': 3, ' s': 3, 'ai': 3, 'ir': 3, 'ss': 3, 'se': 3, ' n': 3, 'ot': 3, 'rt': 3, 'de': 2, 'cl': 2, 'mb': 2, 'bl': 2, 'em': 2, 'fa': 2, 'ly': 2, 'fo': 2, 'da': 2, 'us': 2, 'ac': 2, 'rl': 2, 'ga': 2, 'mp': 2, 'ba': 2, 'ct': 2, 'ag': 2, 'a ': 2, 'ei': 2, 'sh': 2, 'sp': 2, 'mo': 2, 'eo': 2, 'pl': 2, 'ia': 2, ' l': 2, 'so': 2, 'iv': 1, 'sa': 1, 'og': 1, 'nh': 1, 'eq': 1, 'qu': 1, 'ua': 1, 'ab': 1, 'mi': 1, 'il': 1, ' j': 1, 'ju': 1, 'sr': 1, 'eg': 1, 'rd': 1, 'pt': 1, 'su': 1, 'lt': 1, 'rb': 1, 'ut': 1, 'tr': 1, 'ge': 1, 'sc': 1, 'ci': 1, 'nc': 1, 'nk': 1, 'ki': 1, 'ad': 1, 'dv': 1, 'ng': 1, 'gs': 1, 'nj': 1, 'jo': 1, 'oy': 1, 'ef': 1, 'fe': 1, 'wa': 1, 'oc': 1, 'im': 1, 'pi': 1, 'mm': 1, 'if': 1, 'no': 1, 'ur': 1, 'eb': 1, 'yr': 1, 'nn': 1, 'ny': 1, 'pp': 1, 'si': 1, 'ho': 1, 'by': 1, 'ru': 1, 'aw': 1, 'w ': 1, 'ev': 1, 'lo': 1, 'pm': 1, 'dl': 1, 'et': 1, 'tw': 1, 'we': 1, ' u': 1, 'af': 1, 'ff': 1, 'fi': 1, 'rm': 1, 'fu': 1, 'ta': 1})

**Tri-gram Language model:**

To build this model, I have created dictionary **tricharfreqdict** which frequency of all the characters in the training corpus.

eng\_tricharprob = round(tricharfreqdict[trichar],4)/round(onecharfreqdict[trichar[0]],4)

Counter({' th': 18, 'the': 17, 'he ': 16, ' of': 14, 'of ': 14, 'nd ': 12, ' an': 11, 'and': 11, 'ion': 9, 'man': 9, ' in': 9, 'on ': 8, 'an ': 8, 'as ': 8, 'tio': 7, ' hu': 7, 'hum': 7, 'uma': 7, 'rea': 7, ' wh': 7, ' re': 7, 'ed ': 7, 'ati': 6, 'igh': 6, 'ts ': 6, 'her': 6, 'ere': 6, 'f t': 6, 'ent': 6, 'in ': 6, ' be': 6, 'al ': 5, ' ri': 5, 'rig': 5, 'ght': 5, 'hts': 5, 'whe': 5, 'eas': 5, ' fr': 5, ' ha': 5, 'n o': 4, 'n r': 4, ' pr': 4, 'le ': 4, 'nit': 4, 'nt ': 4, 'd i': 4, 'e r': 4, ' is': 4, 'is ': 4, 'om ': 4, ' pe': 4, 'n t': 4, ' co': 4, 'hav': 4, 'ave': 4, 've ': 4, 'd t': 4, 'e c': 4, 'ons': 4, ' to': 4, 'to ': 4, 'ers': 3, 'e w': 3, 'gni': 3, 'e i': 3, ' di': 3, 'y a': 3, 'ien': 3, 's o': 3, 'f a': 3, 'e h': 3, 's t': 3, 'fre': 3, 'ree': 3, 'eed': 3, 'edo': 3, 'dom': 3, 'ce ': 3, 'e a': 3, ' wo': 3, 'wor': 3, 'ld ': 3, 'd w': 3, 'd a': 3, 'res': 3, 'ted': 3, 'ch ': 3, 'e o': 3, 'd b': 3, 'n p': 3, 'pro': 3, ' as': 3, 'st ': 3, 's i': 3, 'ess': 3, 't t': 3, 'ns ': 3, 'uni': 2, ' de': 2, 'cla': 2, 'rat': 2, 'pre': 2, 'ble': 2, 'rec': 2, 'eco': 2, 'dig': 2, 'ign': 2, 'ity': 2, 'ty ': 2, 'd o': 2, 'lie': 2, 'all': 2, 'll ': 2, 'n f': 2, ' fa': 2, 'ly ': 2, ' fo': 2, 'und': 2, 'nda': 2, 'f f': 2, 'orl': 2, 'rld': 2, 'con': 2, 's h': 2, 'n b': 2, 'bar': 2, 's a': 2, 's w': 2, 'whi': 2, 'hic': 2, 'ich': 2, 'h h': 2, 'f m': 2, ' ma': 2, 't o': 2, ' a ': 2, 's s': 2, ' sh': 2, 'bel': 2, 'm f': 2, 'rom': 2, 't h': 2, 's b': 2, 'een': 2, 'en ': 2, 'med': 2, 'com': 2, 'peo': 2, 'eop': 2, 'opl': 2, 'ple': 2, ' it': 2, 'it ': 2, 't i': 2, 's e': 2, ' es': 2, 'sse': 2, 'sen': 2, 'nti': 2, 'tia': 2, 'ial': 2, 'be ': 2, 'ell': 2, ' la': 2, 'ort': 2, 'n a': 2, 'e p': 2, 'ote': 2, 'e d': 2, 'men': 2, ' na': 2, 'nat': 2, 'th ': 2, 'niv': 1, 'ive': 1, 'ver': 1, 'rsa': 1, 'sal': 1, 'l d': 1, 'dec': 1, 'ecl': 1, 'lar': 1, 'ara': 1, 'f h': 1, 's p': 1, 'eam': 1, 'amb': 1, 'mbl': 1, 's r': 1, 'cog': 1, 'ogn': 1, 'iti': 1, 'inh': 1, 'nhe': 1, 'ren': 1, 't d': 1, 'e e': 1, ' eq': 1, 'equ': 1, 'qua': 1, 'ual': 1, 'l a': 1, 'ina': 1, 'nal': 1, 'ali': 1, 'ena': 1, 'nab': 1, 'abl': 1, ' al': 1, 'l m': 1, ' me': 1, 'mem': 1, 'emb': 1, 'mbe': 1, 'ber': 1, 'rs ': 1, 'fam': 1, 'ami': 1, 'mil': 1, 'ily': 1, 'y i': 1, 'e f': 1, 'fou': 1, 'oun': 1, 'dat': 1, 'm j': 1, ' ju': 1, 'jus': 1, 'ust': 1, 'sti': 1, 'tic': 1, 'ice': 1, 'd p': 1, 'pea': 1, 'eac': 1, 'ace': 1, 's d': 1, 'dis': 1, 'isr': 1, 'sre': 1, 'reg': 1, 'ega': 1, 'gar': 1, 'ard': 1, 'rd ': 1, 'd c': 1, 'ont': 1, 'nte': 1, 'tem': 1, 'emp': 1, 'mpt': 1, 'pt ': 1, 't f': 1, 'for': 1, 'or ': 1, 'r h': 1, 'esu': 1, 'sul': 1, 'ult': 1, 'lte': 1, ' ba': 1, 'arb': 1, 'rba': 1, 'aro': 1, 'rou': 1, 'ous': 1, 'us ': 1, ' ac': 1, 'act': 1, 'cts': 1, ' ou': 1, 'out': 1, 'utr': 1, 'tra': 1, 'rag': 1, 'age': 1, 'ged': 1, 'nsc': 1, 'sci': 1, 'cie': 1, 'enc': 1, 'nce': 1, 'ank': 1, 'nki': 1, 'kin': 1, 'ind': 1, ' ad': 1, 'adv': 1, 'dve': 1, 'ven': 1, 'a w': 1, 'n w': 1, 'bei': 1, 'ein': 1, 'ing': 1, 'ngs': 1, 'gs ': 1, 'sha': 1, 'hal': 1, 'l e': 1, ' en': 1, 'enj': 1, 'njo': 1, 'joy': 1, 'oy ': 1, 'y f': 1, 'm o': 1, 'f s': 1, ' sp': 1, 'spe': 1, 'pee': 1, 'eec': 1, 'ech': 1, 'h a': 1, 'eli': 1, 'ief': 1, 'ef ': 1, 'd f': 1, 'fro': 1, ' fe': 1, 'fea': 1, 'ear': 1, 'ar ': 1, 'r a': 1, ' wa': 1, 'wan': 1, 'ant': 1, 'has': 1, 'bee': 1, 'roc': 1, 'ocl': 1, 'lai': 1, 'aim': 1, 'ime': 1, ' hi': 1, 'hig': 1, 'ghe': 1, 'hes': 1, 'est': 1, 't a': 1, 'asp': 1, 'spi': 1, 'pir': 1, 'ira': 1, 'omm': 1, 'mmo': 1, 'mon': 1, 'l i': 1, ' if': 1, 'if ': 1, 'n i': 1, 's n': 1, ' no': 1, 'not': 1, 'ot ': 1, 'o b': 1, 'omp': 1, 'mpe': 1, 'pel': 1, 'lle': 1, 'led': 1, 'o h': 1, 'cou': 1, 'our': 1, 'urs': 1, 'rse': 1, 'se ': 1, 'a l': 1, 'las': 1, 'ast': 1, 't r': 1, 'eso': 1, 'sor': 1, 'rt ': 1, 'o r': 1, 'reb': 1, 'ebe': 1, 'lli': 1, 'lio': 1, ' ag': 1, 'aga': 1, 'gai': 1, 'ain': 1, 'ins': 1, 'nst': 1, ' ty': 1, 'tyr': 1, 'yra': 1, 'ran': 1, 'ann': 1, 'nny': 1, 'ny ': 1, ' op': 1, 'opp': 1, 'ppr': 1, 'ssi': 1, 'sio': 1, 'tha': 1, 'hat': 1, 'at ': 1, 'sho': 1, 'hou': 1, 'oul': 1, 'uld': 1, 'rot': 1, 'tec': 1, 'ect': 1, 'cte': 1, ' by': 1, 'by ': 1, 'y t': 1, ' ru': 1, 'rul': 1, 'ule': 1, 'f l': 1, 'law': 1, 'aw ': 1, 'w w': 1, 'l t': 1, 'o p': 1, 'omo': 1, 'mot': 1, 'te ': 1, 'e t': 1, 'dev': 1, 'eve': 1, 'vel': 1, 'elo': 1, 'lop': 1, 'opm': 1, 'pme': 1, 'fri': 1, 'rie': 1, 'end': 1, 'ndl': 1, 'dly': 1, 'y r': 1, 'rel': 1, 'ela': 1, 'lat': 1, 'bet': 1, 'etw': 1, 'twe': 1, 'wee': 1, 'n n': 1, 'les': 1, 'es ': 1, 'e u': 1, ' un': 1, 'ite': 1, 'd n': 1, ' ch': 1, 'cha': 1, 'har': 1, 'art': 1, 'rte': 1, 'ter': 1, 'er ': 1, 'r r': 1, 'eaf': 1, 'aff': 1, 'ffi': 1, 'fir': 1, 'irm': 1, 'rme': 1, 'hei': 1, 'eir': 1, 'ir ': 1, 'r f': 1, 'fai': 1, 'ait': 1, 'ith': 1, 'h i': 1, ' fu': 1, 'fun': 1, 'dam': 1, 'ame': 1, 'nta': 1, 'tal': 1, 'l h': 1, 'rth': 1, 'h o': 1, 'per': 1, 'rso': 1, 'son': 1})

**Evaluation of Test Corpus:**

Lets classify a word “equal” from the English test model using the training model.

To find **unigram probability**. I find probability of each letter in the training corpus.

Unicharprob = freq[‘e’]/freq[corpus]

Then multiply each probability of a character to find the probability of the word.

To find **Bi-gram probability**. I fetch the frequency of bi-char from the training model

Along with the frequency of starting character from the training model. The evaluate the probability of bi-character. Using this bi- character probability I find the probability of the word to determine its language.

For Bigrams:

\_e, eq, qu, ua, al, l\_

**P(equal) = (P(e|<s>)P(q|e)P(u|q)P(a|u)P(l|a)P(</s> |l)**

Bi-charprob = freq[‘eq’]/freq[‘e’]

Then multiply each probability of a character to find the probability of the word.

To find **Tri-gram probability**. I fetch the frequency of tri-char from the training model. Along with the frequency of starting character from the training model. Then, evaluate the probability of tri-character. Using this tri- character probability I find the probability of the word to determine its language.

For eg: Let’s take the same test word “equal”

**P(equal)=P(q|<s>e)P(u|eq)P(a|qu)P(l|ua)P(</s>|al)**

For eg:

dis 1 d 35 0.0286

tio 7 t 66 0.1061

ion 9 i 57 0.1579

dissolution 0.0004784689 0.0003620325

Marriage 0.0136986301 0.0172413793

sha 1 s 46 0.0217

hal 1 h 53 0.0189

all 2 a 73 0.0274

shall 1.12376e-05 1

**Challenges:**

Initially, I had built the model using the words instead of characters. That gave probabilities very low. This gave inaccurate results during classification. I overcome this by building character level model.

**Results**

The results of the evaluation on 1,000 words.

Output of Problem1:

Accuracy of English test on English vs French Uni-gram models: 74.3 %

Accuracy of English test on English vs French bigram models: 50.0 %

Accuracy of English test on English vs French Tri-gram models: 48.8 %

Output of Problem2:

Accuracy of Spanish test on Spanish vs Italian Uni-gram models: 49.1 %

Accuracy of Spanish test on Spanish vs Italian bi-gram models: 48.2 %

Accuracy of Spanish test on Spanish vs Italian tri-gram models: 63.5 %

Based on the accuracy analysis, I would like to conclude that Spanish vs Italian is harder to distinguish than English vs French language model.

**Concluding Remarks**

Based on the accuracy analysis, I would like to conclude that Spanish vs Italian is harder to distinguish than English vs French language model.

There are few areas which can be worked upon in order to improve the accuracy of the language model further, like

Size of training set – Larger training corpus will lead to calculation of accurate statistics (frequency counts) of bi-grams, tri-grams in the language.