IS567 - Text Mining
Assignment 4

Word Embeddings

Task 1: Create your own word embeddings

Task 1: Create your own word embeddings

```
fairy_data = pd.read_csv("fairy_tale.csv", low_memory=False)
fairy_data.head(1)
```

	fairy_tale	country	character	gender	count_of_appearance	if_main(>=20)	sentences	sentences_list	Unnamed: 8	Unnamed: 9	...	Unnamed: 183	Unnamed: 184
0	cannetella	NaN	wife	female	3	0	but at last, after he had been married for many years, and was quite an old man, his wife renzolla presented him with a fine daughter, whom they called cannetella.if you will step in here you shall have my daughter for a wife, and i will send attendants with her, and as many horses and servants as you wish.?€? \n?€?a thousand thanks,?€? returned scioravante, ?€?i shall be delighted to marry your daughter, but it is quite unnecessary to send anyone to accompany her.scioravante	['but at last, after he had been married for many years, and was quite an old man, his wife renzolla presented him with a fine daughter, whom they called cannetella.', 'if you will step in here you shall have my daughter for a wife, and i will send attendants with her, and as many horses and servants as you wish.?€? \n?€?a thousand thanks,?€? returned scioravante, ?€?i shall be delighted to marry your daughter, but it is quite unnecessary to send anyone to accompany her.', 'scioravante	NaN	NaN	...	NaN	NaN

```
fairy_data.drop(fairy_data.iloc[:, 8:], axis=1, inplace=True)
fairy_data.head(1)
```

	fairy_tale	country	character	gender	count_of_appearance	if_main(>=20)	sentences	sentences_list
0	cannetella	NaN	wife	female	3	0	but at last, after he had been married for many years, and was quite an old man, his wife renzolla presented him with a fine daughter, whom they called cannetella.if you will step in here you shall have my daughter for a wife, and i will send attendants with her, and as many horses and servants as you wish.?€?\n?€?a thousand thanks,?€? returned scioravante, ?€?i shall be delighted to marry your daughter, but it is quite unnecessary to send anyone to accompany her.scioravante was furious when he heard this, and seizing a huge knife from his pocket he threatened to kill his wife for her disobedience.	['but at last, after he had been married for many years, and was quite an old man, his wife renzolla presented him with a fine daughter, whom they called cannetella.', 'if you will step in here you shall have my daughter for a wife, and i will send attendants with her, and as many horses and servants as you wish.?€?\n?€?a thousand thanks,?€? returned scioravante, ?€?i shall be delighted to marry your daughter, but it is quite unnecessary to send anyone to accompany her.', 'scioravante was furious when he heard this, and seizing a huge knife from his pocket he threatened to kill his wife for her disobedience.']

```
fairy_data.columns
```

```
Index(['fairy_tale', 'country', 'character', 'gender', 'count_of_appearance',
       'if_main(>=20)', 'sentences', 'sentences_list'],
      dtype='object')
```

Number of instances in the training dataset with blank reviews

```
len(fairy_data)
```

```
6541
```

```
fairy_data = fairy_data[~fairy_data["sentences"].isnull()]
```

Number of instances in the training dataset without blank reviews

```
len(fairy_data)
```

```
6537
```

After removing blank reviews, we have **6537** instances in the fairy_tale dataset.

Since, I encountered punctuations and joint words in the word embeddings, I decided to perform preprocessing on the sentences column by removing all the punctuations and non-alphanumeric characters using the regex module.

```
import re

def preprocess_sentences(text):

    text = re.sub(r'\W+', ' ', text)

    return text
```

```
fairy_data.loc[:, ('preprocessed')] = fairy_data.loc[:, ('sentences')].apply(preprocess_sentences)
fairy_data['preprocessed'][1]
```

```
'the king who felt himself growing old and feeble and longed to see an heir to the throne before he died was very u
nhappy at her words and begged her earnestly not to disappoint him the spirits said at first that the task was beyo
nd their powers and suggested that a pair of golden horns attached to his forehead would both be easier to make and
more comfortable to wear but scioravante would allow no compromise and insisted on having a head and teeth made of
the finest gold when the servants came in haste and saw only the cooper standing at the gate they were very indigna
nt and scolded him soundly for coming at such an hour and waking them all out of their sleep '
```

```
sentences = [sentence.split() for sentence in fairy_data['preprocessed']]
```


```
sentences = [sentence.split() for sentence in fairy_data['preprocessed']]
```


Skip gram word embeddings

```
skipgram_model = Word2Vec(sentences, sg=1, vector_size=100, window=5, min_count=2, workers=10)
```

Top 20 similar words using Skip Gram word embeddings

```
for word in word_list:
    print(f"Similar words for '{word}' using Skip-gram: \n")
    find_similar_words(skipgram_model, word)
    print("\n")
    print("-----------------------------------------------\n")
```

```
Similar words for 'man' using Skip-gram:

('woodcutter', 0.6842487454414368)
('woman', 0.6739736795425415)
('hag', 0.6737663149833679)
('traveler', 0.6663625836372375)
('traveller', 0.6636632084846497)
('cowherd', 0.6620761156082153)
('rascals', 0.6505754590034485)
('butler', 0.6499751210212708)
('buffalo', 0.6416180729866028)
('cobbler', 0.6409675478935242)
('singeing', 0.6362209916114807)
('demon', 0.6315685510635376)
('barber', 0.6314840912818909)
('labourer', 0.6309095621109009)
('shoemaker', 0.6249475479125977)
('riou', 0.6241922378540039)
('driver', 0.6235538721084595)
('ox', 0.6209613680839539)
('customer', 0.618779182434082)
('weaver', 0.6182336807250977)


-----------------------------------------------
```

```
Similar words for 'woman' using Skip-gram:

('man', 0.673973560333252)
('toad', 0.6651650667190552)
('hag', 0.6482008695602417)
('old', 0.6459565758705139)
('witch', 0.6329318284988403)
('butler', 0.6303555965423584)
('washerwoman', 0.5940973162651062)
('crook', 0.5885552763938904)
('sorceress', 0.5866442322731018)
('widow', 0.5839696526527405)
('ogress', 0.5824714303016663)
('bustling', 0.5789439678192139)
('traveler', 0.5758263468742371)
('girl', 0.5743936896324158)
('corva', 0.5704936385154724)
('dame', 0.5695381760597229)
('maisie', 0.5691765546798706)
('cobbler', 0.5687671303749084)
('salad', 0.5682714581489563)
('roughly', 0.5658972263336182)


----------------------------------------------

Similar words for 'king' using Skip-gram:

('kostiei', 0.6748406291007996)
('merlin', 0.6598561406135559)
('dungting', 0.65935879945755)
('doctor', 0.6590296626091003)
('tidings', 0.6484134793281555)
('betroth', 0.647691011428833)
('finnuala', 0.6466795802116394)
('playfellow', 0.6464390754699707)
('slayer', 0.6446539759635925)
('marvelled', 0.6445484757423401)
('pasties', 0.6436018943786621)
('fafnir', 0.6428729891777039)
('timus', 0.6398259401321411)
('quimus', 0.6368054747581482)
('malcolm', 0.6354914903640747)
('lillwacker', 0.6346101760864258)
('duke', 0.6340128183364868)
('frivola', 0.6325249075889587)
('decision', 0.6323352456092834)
('rajah', 0.6320629715919495)


Similar words for 'queen' using Skip-gram:

('fiordelisa', 0.5802242159843445)
('turritella', 0.5669391751289368)
('princess', 0.5665605068206787)
('olof', 0.5577449202537537)
('malcolm', 0.5570718050003052)
('hermod', 0.5520205497741699)
('graciosa', 0.54844069480896)
('christening', 0.5418056845664978)
('bellissima', 0.5405347347259521)
('cerisette', 0.5374841094017029)
('sylvia', 0.5366541743278503)
('muffette', 0.5364970564842224)
('vexing', 0.5345731377601624)
('delicia', 0.5343741178512573)
('heathen', 0.5314059257507324)
('maids', 0.5313679575920105)
('gemdelovely', 0.5309333205223083)
('dotterine', 0.5292369723320007)
('tulip', 0.5291243195533752)
('slayer', 0.5290386080741882)
```

CBOW word embeddings

```
cbow_model = Word2Vec(sentences, sg=0, vector_size=100, window=5, min_count=2, workers=10)
```

Top 20 similar words using CBOW word embeddings

```
for word in word_list:
    print(f"Similar words for '{word}' using CBOW: \n")
    find_similar_words(cbow_model, word)
    print("\n")
    print("-----------------------------------------------\n")
```

```
Similar words for 'man' using CBOW:

('fellow', 0.763799786567688)
('gentleman', 0.7411456108093262)
('woman', 0.673238217830658)
('shepherd', 0.6700393557548523)
('farmer', 0.6568666696548462)
('lad', 0.6504793167114258)
('wolf', 0.6330461502075195)
('hag', 0.6157364249229431)
('manito', 0.6144159436225891)
('magician', 0.6137053370475769)
('spitfire', 0.5928245782852173)
('dame', 0.5880370140075684)
('valentine', 0.5639962553977966)
('men', 0.5579580068588257)
('shoemaker', 0.5547490119934082)
('squire', 0.5529981851577759)
('traveler', 0.5525742173194885)
('negro', 0.5440123677253723)
('merchant', 0.5356326103210449)
('grandmother', 0.5355800986289978)
```

```
Similar words for 'woman' using CBOW:

('witch', 0.827531099319458)
('nurse', 0.7142103910446167)
('hag', 0.7112809419631958)
('dame', 0.7081018090248108)
('valentine', 0.6829209923744202)
('hopgiant', 0.6821596026420593)
('man', 0.6732382774353027)
('manito', 0.6549601554870605)
('sheepskins', 0.6324551701545715)
('manuscripts', 0.6240206956863403)
('lady', 0.6231081485748291)
('peggy', 0.6159660220146179)
('stepmother', 0.6067373156547546)
('magician', 0.6038603186607361)
('washerwoman', 0.6015623807907104)
('eric', 0.6006999015808105)
('birscha', 0.6006133556365967)
('tout', 0.5932418704032898)
('dragonbeard', 0.587490439414978)
('grandmother', 0.5827929377555847)


-----------------------------------------------

Similar words for 'king' using CBOW:

('emperor', 0.6693485975265503)
('sultan', 0.6074044704437256)
('youth', 0.5605049133300781)
('miller', 0.5479802489280701)
('governor', 0.5479605793952942)
('apothecary', 0.5295941233634949)
('magician', 0.5140270590782166)
('merchant', 0.5074111819267273)
('princess', 0.5059531927108765)
('gardener', 0.49870774149894714)
('bride', 0.4961302578449249)
('knight', 0.49398288130760193)
('vizier', 0.4866202473640442)
('doctor', 0.4857308566570282)
('prince', 0.48108401894569397)
('duke', 0.4787166118621826)
('fairy', 0.47768592834472656)
('sigurd', 0.4766392409801483)
('minister', 0.471764475107193)
('ambassador', 0.47008389234542847)
```

```
Similar words for 'queen' using CBOW:

('princess', 0.6378299593925476)
('stepmother', 0.6247984766960144)
('fairy', 0.5980861783027649)
('fiordelisa', 0.5596822500228882)
('turritella', 0.5579517483711243)
('nurse', 0.5327401757240295)
('lady', 0.5325422883033752)
('ladies', 0.525444746017456)
('prince', 0.5167262554168701)
('marriage', 0.5121680498123169)
('maiden', 0.5061949491500854)
('godmother', 0.48343807458877563)
('husband', 0.48004889488220215)
('girl', 0.4766191244125366)
('she', 0.4711868166923523)
('youth', 0.4694986641407013)
('daughter', 0.4562901556491852)
('genius', 0.4436969459056854)
('countess', 0.44267627596855164)
('beauty', 0.4376605153083801)
```

Fast Text word embeddings

```
fasttext_model = FastText(sentences, vector_size=100, window=5, min_count=2, workers=10)
```

Top 20 similar words using FastText word embeddings

```
: for word in word_list:
    print(f"Similar words for '{word}' using Fast Text: \n")
    find_similar_words(fasttext_model, word)
    print("\n")
    print("-----------------------------------------------\n")
```

```
Similar words for 'man' using Fast Text:

('zaman', 0.9670705199241638)
('lyman', 0.9660375714302063)
('bugleman', 0.9590309262275696)
('ferryman', 0.9462440013885498)
('madman', 0.9433199763298035)
('german', 0.9352989196777344)
('milkman', 0.931079626083374)
('marksman', 0.9307622909545898)
('suliman', 0.9292066693305969)
('bergman', 0.9287408590316772)
('scotsman', 0.9251372218132019)
('craftsman', 0.9248241782188416)
('dorman', 0.923179566860199)
('kemperman', 0.9204888343811035)
('merman', 0.9201430678367615)
('gintleman', 0.9197134971618652)
('roman', 0.9118409752845764)
('boatman', 0.9095393419265747)
('lawman', 0.9026129245758057)
('irishman', 0.8987494707107544)
```

```
Similar words for 'woman' using Fast Text:

('gentlewoman', 0.9593326449394226)
('washerwoman', 0.9513521790504456)
('roman', 0.942244291305542)
('womanish', 0.9330605268478394)
('womanhood', 0.9310193061828613)
('milkman', 0.9062278866767883)
('lyman', 0.9041537642478943)
('herdsman', 0.8905116319656372)
('zaman', 0.8900577425956726)
('bergman', 0.8822266459465027)
('boatman', 0.8805782794952393)
('woodman', 0.8800581097602844)
('ferryman', 0.8643255829811096)
('man', 0.8640374541282654)
('bugleman', 0.8632209300994873)
('workman', 0.8628254532814026)
('german', 0.8609150052070618)
('scotsman', 0.8608345985412598)
('lawman', 0.8600528836250305)
('craftsman', 0.8580897450447083)


----------------------------------------------

Similar words for 'king' using Fast Text:

('kingly', 0.8547879457473755)
('kingfisher', 0.8407551646232605)
('kinglet', 0.8249655961990356)
('kingship', 0.8040273189544678)
('kiang', 0.7756890654563904)
('kicking', 0.7688755989074707)
('kin', 0.7587001323699951)
('kindling', 0.7558193802833557)
('pricking', 0.7525895833969116)
('emperor', 0.7500121593475342)
('kings', 0.7404184937477112)
('hawking', 0.7298561334609985)
('provoking', 0.7256158590316772)
('risking', 0.725462794303894)
('jerking', 0.7213072776794434)
('mocking', 0.718149721622467)
('lurking', 0.7181344032287598)
('ducking', 0.7153449654579163)
('basking', 0.7152655720710754)
('nanking', 0.710342526435852)


Similar words for 'queen' using Fast Text:

('queens', 0.9337019920349121)
('queenly', 0.8975504040718079)
('quen', 0.8263441920280457)
('queer', 0.8086615800857544)
('queerly', 0.7828441262245178)
('quesnoy', 0.7571499347686768)
('fairly', 0.7444673776626587)
('queerest', 0.7440266013145447)
('fair', 0.7351124286651611)
('turritella', 0.7239276766777039)
('dairy', 0.7235289216041565)
('fairy', 0.7119830250740051)
('sorceress', 0.7093159556388855)
('priestess', 0.7076519727706909)
('miry', 0.7063767313957214)
('princess', 0.7045500874519348)
('squeeze', 0.7023947834968567)
('empress', 0.7006140351295471)
('cess', 0.7005020976066589)
('jewess', 0.6950124502182007)
```

Task 2: Use pre-trained word embeddings via GloVe

Task 2: Use pretrained word embeddings

```
# First we have to convert the Glove format into w2v format; this creates a new file

glove_file = "Datasets/wordembeddings/glove.6B.100d.100K.txt"
glove_in_w2v_format = "Datasets/wordembeddings/glove.6B.100d.100K.w2v.txt"
_ = glove2word2vec(glove_file, glove_in_w2v_format)
```

```
/var/folders/j2/sk_tpq311dn0rsb2ksfpkzb00000gn/T/ipykernel_58827/1099280873.py:5: DeprecationWarning: Call to deprecat
word2vec_format(.., binary=False, no_header=True) loads GLoVE text vectors.).
  _ = glove2word2vec(glove_file, glove_in_w2v_format)
```

```

```

```
glove_model = KeyedVectors.load_word2vec_format("Datasets/wordembeddings/glove.6B.100d.100K.w2v.txt", binary=False)
```

```
def find_similar_words_glove(model, word, top_n=20):
    try:
        similar_words = model.most_similar(word, topn=top_n)
        for sim_word in similar_words:
            print(sim_word)

    except KeyError:
        return []
```

```
def find_similar_words_glove(model, word, top_n=20):
    try:
        similar_words = model.most_similar(word, topn=top_n)
        for sim_word in similar_words:
            print(sim_word)

    except KeyError:
        return []
```

```
for word in word_list:
    print(f"Similar words for '{word}' using GloVe: \n")
    find_similar_words_glove(glove_model, word)
    print("\n")
    print("-----------------------------------------------\n")
```

```
Similar words for 'man' using GloVe:

('woman', 0.8323495388031006)
('boy', 0.7914870977401733)
('one', 0.7788748741149902)
('person', 0.7526815533638)
('another', 0.752223551273346)
('old', 0.7409117221832275)
('life', 0.737169623374939)
('father', 0.7370322346687317)
('turned', 0.7347695231437683)
('who', 0.734551191329956)
('whose', 0.7326126098632812)
('girl', 0.7291691303253174)
('he', 0.7255576252937317)
('him', 0.7238516807556152)
('young', 0.7218634486198425)
('himself', 0.7214202284812927)
('friend', 0.7170529961585999)
('once', 0.7132790684700012)
('being', 0.7123121023178101)
('a', 0.7093364000320435)
```

```
Similar words for 'woman' using GloVe:

('girl', 0.8472672700881958)
('man', 0.832349419593811)
('mother', 0.8275688290596008)
('boy', 0.7720510959625244)
('she', 0.7632068395614624)
('child', 0.7601761817932129)
('wife', 0.7505022883415222)
('her', 0.7445706129074097)
('herself', 0.7426273822784424)
('daughter', 0.726445734500885)
('victim', 0.7244972586631775)
('person', 0.7226606011390686)
('female', 0.7213833332061768)
('husband', 0.7211335897445679)
('pregnant', 0.7158757448196411)
('teenager', 0.7041847109794617)
('couple', 0.7007153034210205)
('women', 0.6860839128494263)
('old', 0.6820175051689148)
('young', 0.6819227933883667)


------------------------------------------

Similar words for 'king' using GloVe:

('prince', 0.7682329416275024)
('queen', 0.7507689595222473)
('son', 0.7020888328552246)
('brother', 0.6985775828361511)
('monarch', 0.6977890729904175)
('throne', 0.691999077796936)
('kingdom', 0.6811409592628479)
('father', 0.680202841758728)
('emperor', 0.67128586769104)
('ii', 0.6676074266433716)
('crown', 0.664752721786499)
('reign', 0.66302889585495)
('uncle', 0.6608153581619263)
('henry', 0.6578148603439331)
('ruler', 0.6538782119750977)
('iii', 0.6498465538024902)
('nephew', 0.6484619975090027)
('edward', 0.6482085585594177)
('charles', 0.6414701342582703)
('george', 0.6374154090881348)



Similar words for 'queen' using GloVe:

('princess', 0.7947244644165039)
('king', 0.7507690191268921)
('elizabeth', 0.7355711460113525)
('royal', 0.7065026164054871)
('lady', 0.7044796943664551)
('victoria', 0.6853757500648499)
('monarch', 0.6683257818222046)
('crown', 0.6680562496185303)
('prince', 0.6640505790710449)
('consort', 0.6570538282394409)
('majesty', 0.6483592391014099)
('daughter', 0.6395184993743896)
('duchess', 0.6349949240684509)
('coronation', 0.6347757577896118)
('throne', 0.6307163834571838)
('mary', 0.6274112462997437)
('sister', 0.6231560111045837)
('mother', 0.6169339418411255)
('wife', 0.6161715984344482)
('margaret', 0.6038230061531067)
```

When we compare within our own word embeddings using the CBOW and Skip-gram, we see that the skip-gram word embeddings try to predict some rare words that may be similar to the target word whereas the CBOW does the opposite and tries to find frequent words. The Skip-gram takes more training time as compared to the CBOW word embeddings.

When comparing the FastText word embeddings makes use of character n-grams for generating the word embeddings and as a result it also helps in identifying word embeddings for words that are infrequent. Another big advantage is that it can also produce embeddings for words that do not appear in the text corpus.

Talking about the distinction between the own word embeddings using word2vec techniques and GloVe, here are the observations:

- Word2vec works on the idea of predicting context words based on target words (Skip-grams) and vice versa (CBOW), whereas the GloVe uses a global corpora for identifying word-to-word cooccurrence statistics to obtain lower dimensional features.
- Word2vec captures the semantics between the words based on the text corpora and hence is able to obtain similar words related to the text under consideration. They are aware of the local context information and are suitable for contextual word meanings.
- On the other hand, since GloVe takes into consideration global corpora they may not be able to obtain words that are closer in meaning with respect to the text. They will give generalised solution for similar words.
- In our case, when we see the output of the word2vec similar words we see that for ‘man’ we obtain rare words like ‘squire’, ‘shoemaker’, ‘riou’, ‘cobbler’, ‘ox’. Here we see that ‘ox’ is an animal and not very related to man. But the essence that ‘ox’ is masculine it is similar to ‘man’.
- In our case, when we see the output of the GloVe similar words we see that for ‘man’ we obtain common words like ‘boy’, ‘friend’, ‘father’ and so on. Here we see that these words are highly similar to ‘man’ but they are general words that we may use in everyday language and not specific to the text corpus under consideration (fairy_tale data).

Topic Modeling Tasks

Task 1: Data preprocessing

Task 1: Data preprocessing

```
fairy_data = pd.read_csv("fairy_tale.csv", low_memory=False)
fairy_data.head(1)
```

	fairy_tale	country	character	gender	count_of_appearance	if_main(>=20)	sentences	sentences_list	Unnamed: 8	Unnamed: 9
0	cannetella	NaN	wife	female	3	0	but at last, after he had been married for many years, and was quite an old man, his wife renzolla presented him with a fine daughter, whom they called cannetella.if you will step in here you shall have my daughter for a wife, and i will send attendants with her, and as many horses and servants as you wish.?€?\n?€?a thousand thanks,?€? returned scioravante, ?€?i shall be delighted to marry your daughter, but it is quite unnecessary to send anyone to accompany her.scioravante was furious when he heard this, and seizing a huge knife from his pocket he threatened to kill his wife for her	['but at last, after he had been married for many years, and was quite an old man, his wife renzolla presented him with a fine daughter, whom they called cannetella.', 'if you will step in here you shall have my daughter for a wife, and i will send attendants with her, and as many horses and servants as you wish.?€?\n?€?a thousand thanks,?€? returned scioravante, ?€?i shall be delighted to marry your daughter, but it is quite unnecessary to send anyone to accompany her.', 'scioravante was furious when he heard this, and seizing a huge knife from his pocket he threatened to kill his wife for her	NaN	NaN

```
[681]: fairy_data.drop(fairy_data.iloc[:, 8:], axis=1, inplace=True)
       fairy_data
```

```
rs/j2/sk_tpq311dn0rsb2ksfpkzb00000gn/T/ipykernel_58827/1987663237.py:1: SettingWithCopyWarning:
 trying to be set on a copy of a slice from a DataFrame

veats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
ta.drop(fairy_data.iloc[:, 8:], axis=1, inplace=True)
```

fairy_tale	country	character	gender	count_of_appearance	if_main(>=20)	sentences	sentences_list
cannetella	NaN	wife	female	3	0	but at last, after he had been married for many years, and was quite an old man, his wife renzolla presented him with a fine daughter, whom they called cannetella.if you will step in here you shall have my daughter for a wife, and i will send attendants with her, and as many horses and servants as you wish.?€?\n?€?a thousand thanks,?€? returned scioravante, ?€?i shall be delighted to marry your daughter, but it is quite unnecessary to send anyone to accompany her.scioravante was furious when he heard this, and seizing a huge knife from his pocket he threatened to kill his wife for her disobedience.	['but at last, after he had been married for many years, and was quite an old man, his wife renzolla presented him with a fine daughter, whom they called cannetella.', 'if you will step in here you shall have my daughter for a wife, and i will send attendants with her, and as many horses and servants as you wish.?€?\n?€?a thousand thanks,?€? returned scioravante, ?€?i shall be delighted to marry your daughter, but it is quite unnecessary to send anyone to accompany her.', 'scioravante was furious when he heard this, and seizing a huge knife from his pocket he threatened to kill his wife for her disobedience.']
						the king, who felt himself growing old and feeble, and longed to see	['the king, who felt himself growing old and feeble, and longed to

```
male_chars = fairy_data[fairy_data['gender'] == 'male']
male_chars = male_chars.reset_index()
male_chars.head(1)
```

	index	fairy_tale	country	character	gender	count_of_appearance	if_main(>=20)	sentences	sentences_list
0	1	cannetella	NaN	heir	male	3	0	the king, who felt himself growing old and feeble, and longed to see an heir to the throne before he died, was very unhappy at her words, and begged her earnestly not to disappoint him.the spirits said, at first, that the task was beyond their powers, and suggested that a pair of golden horns attached to his forehead would both be easier to make and more comfortable to wear, but scioravante would allow no compromise, and insisted on having a head and teeth made of the finest gold.when the servants came in haste and saw only the cooper standing at the gate, they were very indignant, and scolded him soundly for coming at such an hour and waking them all out of their sleep.	['the king, who felt himself growing old and feeble, and longed to see an heir to the throne before he died, was very unhappy at her words, and begged her earnestly not to disappoint him.', 'the spirits said, at first, that the task was beyond their powers, and suggested that a pair of golden horns attached to his forehead would both be easier to make and more comfortable to wear, but scioravante would allow no compromise, and insisted on having a head and teeth made of the finest gold.', 'when the servants came in haste and saw only the cooper standing at the gate, they were very indignant, and scolded him soundly for coming at such an hour and waking them all out of their sleep.']

Number of male characters in the dataset

```
len(male_chars)
```

```
4403
```

The number of male characters is 4403.

```
female_chars = fairy_data[fairy_data['gender'] == 'female']
female_chars = female_chars.reset_index()
female_chars.head(1)
```

	index	fairy_tale	country	character	gender	count_of_appearance	if_main(>=20)	sentences	sentences_list
0	0	cannetella	NaN	wife	female	3	0	but at last, after he had been married for many years, and was quite an old man, his wife renzolla presented him with a fine daughter, whom they called cannetella.if you will step in here you shall have my daughter for a wife, and i will send attendants with her, and as many horses and servants as you wish.?€?\n?€?a thousand thanks,?€? returned scioravante, ?€?i shall be delighted to marry your daughter, but it is quite unnecessary to send anyone to accompany her.scioravante was furious when he heard this, and seizing a huge knife from his pocket he threatened to kill his wife for her disobedience.	['but at last, after he had been married for many years, and was quite an old man, his wife renzolla presented him with a fine daughter, whom they called cannetella.', 'if you will step in here you shall have my daughter for a wife, and i will send attendants with her, and as many horses and servants as you wish.?€?\n?€?a thousand thanks,?€? returned scioravante, ?€?i shall be delighted to marry your daughter, but it is quite unnecessary to send anyone to accompany her.', 'scioravante was furious when he heard this, and seizing a huge knife from his pocket he threatened to kill his wife for her disobedience.']

Number of female characters in the dataset

```
len(female_chars)
```

```
2125
```

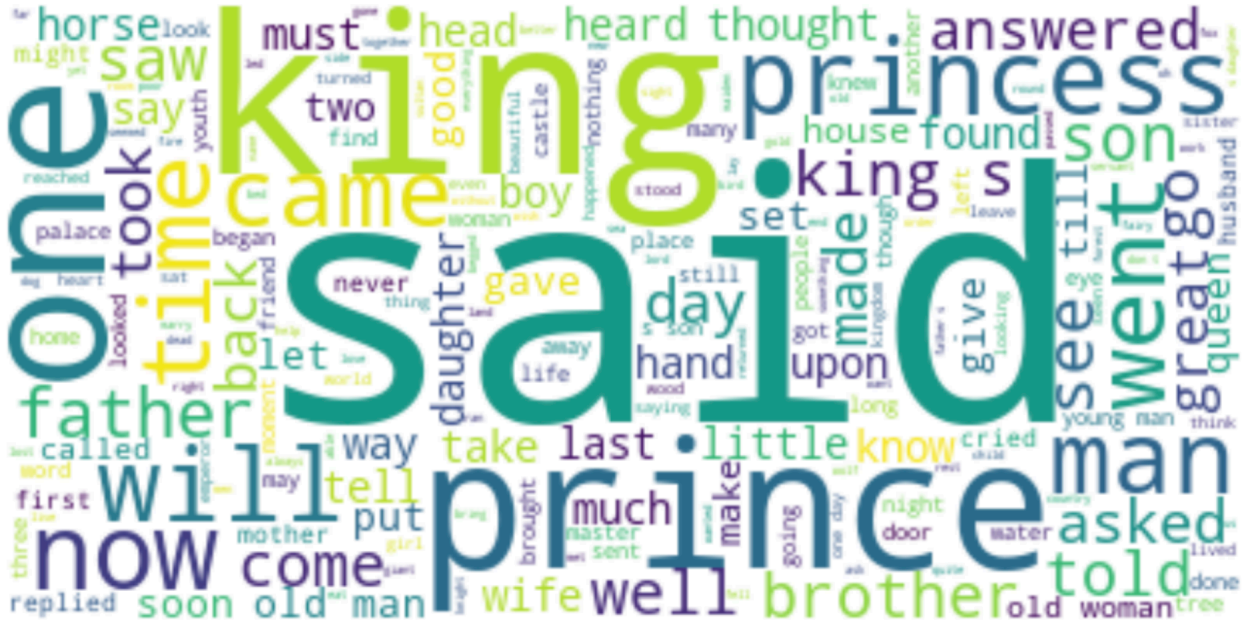
The number of female characters is 2125.

Task 2: Topic modeling for male characters

Task 2: Topic modeling for male characters.

```
male_chars.loc[:, ('preprocessed')] = male_chars.loc[:, ('sentences')].apply(preprocess_sentences)
```

```
text = " ".join(str(i) for i in male_chars['preprocessed'])
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords, background_color="white").generate(text)
plt.figure(figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



From the word cloud we observe that for males, we can see that the top and most frequent words include ‘said’, ‘king’, ‘prince’, ‘princess’, ‘one’, ‘man’, ‘father’. This suggests that the key theme of this is on the **masculine** gender and there are some words like ‘princess’ that are related to a male character. Multiple words are repeated suggesting redundancy in the data.

Preparing the data for LDA Topic Modeling and an example

Preprocess sentences using word tokenizationa and stopword removal

```
import string
```

```
def preprocess_sentences(text):

    text = re.sub(r'\W+', ' ', text)

    word_token = word_tokenize(text)
    filtered_words = [word for word in word_token if word not in stop_words]

    return filtered_words
```

```
male_chars.loc[:, ('preprocessed_sentences')] = male_chars.loc[:, ('sentences')].apply(preprocess_sentences)
```

Examples of preprocessed sentences

```
grateful_prince = male_chars[(male_chars.character == 'wizard') & (male_chars.fairy_tale == 'the_grateful_prince')]
grateful_prince
```

	index	fairy_tale	country	character	gender	count_of_appearance	if_main(>=20)	sentences	sentences_list	preprocessed	preprocessed_sentences
680	986	the_grateful_prince	NaN	wizard	male	2	0	?€?here the old wizard has no more power over us, and we can guard ourselves from his spells.but the father was no longer there, for the loss of his son had killed him, and on his deathbed he confessed to his people how he had contrived that the old wizard should carry away a peasant?€?s child instead of the prince, wherefore this punishment had fallen upon him.	['?€?here the old wizard has no more power over us, and we can guard ourselves from his spells.', 'but the father was no longer there, for the loss of his son had killed him, and on his deathbed he confessed to his people how he had contrived that the old wizard should carry away a peasant?€?s child instead of the prince, wherefore this punishment had fallen upon him.']	here the old wizard has no more power over us and we can guard ourselves from his spells but the father was no longer there for the loss of his son had killed him and on his deathbed he confessed to his people how he had contrived that the old wizard should carry away a peasant s child instead of the prince wherefore this punishment had fallen upon him	[old, wizard, power, us, guard, spells, father, longer, loss, son, killed, deathbed, confessed, people, contrived, old, wizard, carry, away, peasant, child, instead, prince, wherefore, punishment, fallen, upon]

BOW representation

```
from gensim import corpora, models
from gensim.models import LdaModel
```

```
dictionary = corpora.Dictionary(male_chars['preprocessed_sentences'])
dictionary.filter_extremes(no_below=5, no_above=0.2, keep_n=10000)

corpus = [dictionary.doc2bow(sentence) for sentence in male_chars['preprocessed_sentences']]
```

Restricting top 10K terms that appear in 5 documents and not more that 20% documents.

5 Topics for males using LDA

Top 20 words under the 5 topics

```
i=0
for i, topic in lda_model_males_5.show_topics(num_words=20):
    print(f"TOPIC: \n\n", topic, "\n\n-------------------------\n")
    i+=1
```

```
TOPIC:

 0.022*"sister" + 0.010*"witch" + 0.008*"let" + 0.008*"brothers" + 0.008*"morning" + 0.007*"house" + 0.007*"father" + 0.006*"cat" + 0.006*"shall" + 0.006*"child" + 0.006*"two" + 0.006*"castle" + 0.005*"white" + 0.005*"dear" + 0.005*"brother" + 0.005*"boy" + 0.005*"called" + 0.005*"cried" + 0.005*"poor" + 0.005*"lived"

-------------------------

TOPIC:

 0.027*"maid" + 0.025*"prince" + 0.010*"jamila" + 0.008*"care" + 0.008*"forest" + 0.008*"pretty" + 0.007*"golden" + 0.007*"fairy" + 0.007*"lady" + 0.007*"trouble" + 0.007*"youngest" + 0.006*"named" + 0.006*"know" + 0.006*"taken" + 0.005*"let" + 0.005*"deer" + 0.005*"throne" + 0.005*"moment" + 0.005*"house" + 0.005*"heaven"

-------------------------

TOPIC:

 0.027*"queen" + 0.010*"two" + 0.009*"cried" + 0.007*"miranda" + 0.007*"shall" + 0.007*"bride" + 0.007*"sheep" + 0.007*"heart" + 0.006*"nurse" + 0.006*"father" + 0.006*"life" + 0.005*"head" + 0.005*"golden" + 0.005*"know" + 0.005*"even" + 0.005*"love" + 0.005*"lovely" + 0.005*"prince" + 0.005*"fairy" + 0.004*"hand"

-------------------------

TOPIC:

 0.032*"fairy" + 0.018*"queen" + 0.010*"desert" + 0.008*"upon" + 0.008*"cried" + 0.008*"dwarf" + 0.007*"love" + 0.007*"help" + 0.007*"make" + 0.007*"quite" + 0.006*"bellissima" + 0.006*"really" + 0.006*"lady" + 0.005*"die" + 0.005*"replied" + 0.005*"madam" + 0.005*"mermaid" + 0.005*"mistress" + 0.005*"gold" + 0.005*"tell"

-------------------------

TOPIC:

 0.010*"prince" + 0.008*"queen" + 0.007*"lady" + 0.005*"might" + 0.005*"youth" + 0.005*"door" + 0.005*"last" + 0.005*"upon" + 0.005*"found" + 0.004*"palace" + 0.004*"son" + 0.004*"poor" + 0.004*"hair" + 0.004*"night" + 0.004*"first" + 0.004*"till" + 0.004*"say" + 0.003*"find" + 0.003*"left" + 0.003*"put"
```

For visualization using LDA, we had to downgrade the pandas version to 1.5.3 and pyLDAvis to 3.4.0

Vizualization of LDA Model

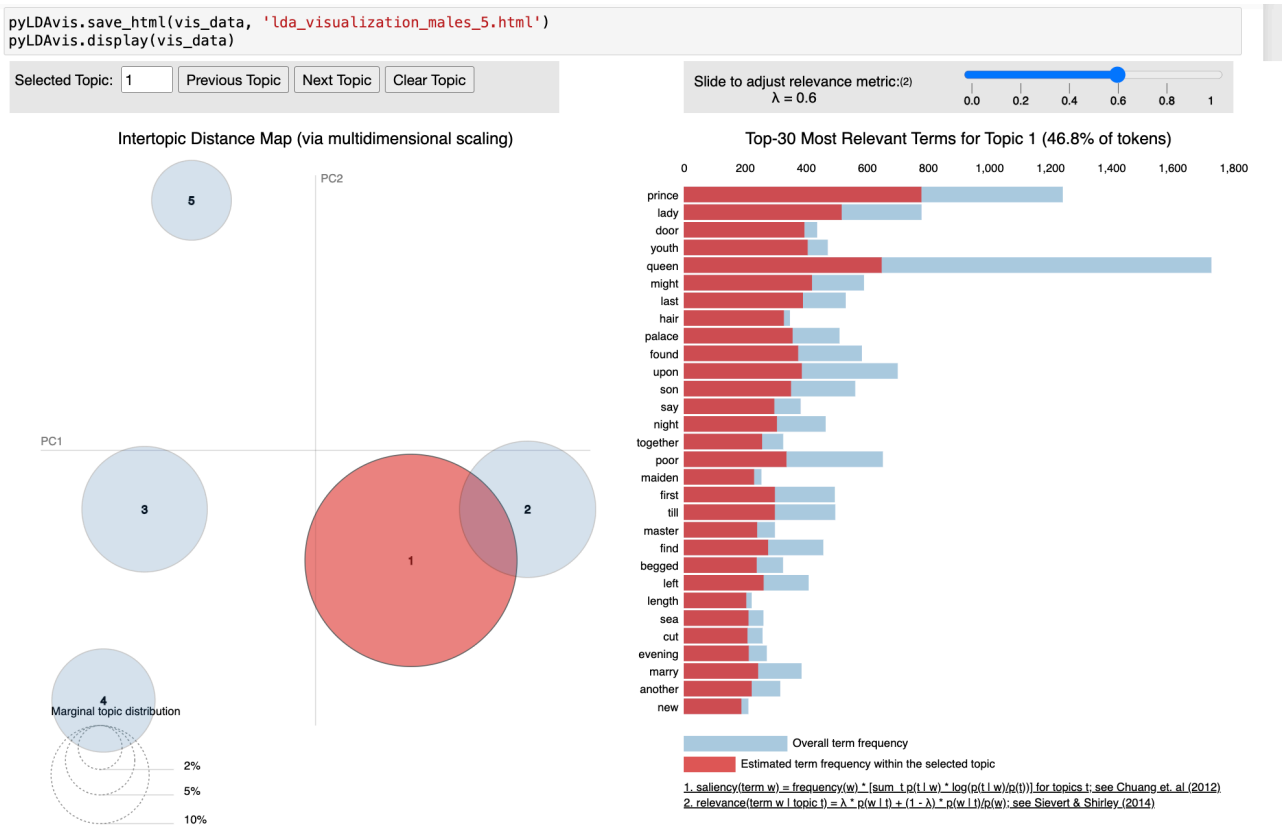
```
: pip install pandas==1.5.3
```

```
: pip install pyLDAvis==3.4.0
```

```
: pd.__version__
: '1.5.3'
```

```
: pyLDAvis.__version__
: '3.4.0'
```

```
: import pyLDAvis.gensim
  import pyLDAvis.gensim_models
  import pyLDAvis.gensim_models as gensimvis

  pyLDAvis.enable_notebook()
```



The 5 topics for males using sentences are shown as above. Keeping $\lambda = 0.6$ this helps us in selecting words that are relevant to that topic to give more importance to ratio of frequency of word given topic to the overall frequency of word.

- We observe that topic 1, topic 2, topic 3, topic 4, topic 5 correspond to 46.8%, 19.6%, 16.3%, 11.1%, and 6.6% of the tokens in the corpus respectively.
- Topic 1 is separated from topics 3, 4 and 5 but overlaps with topic 2.
- All other topics are well separated from each other and have clear distinction in terms of the inter topic distance.
- Topic 1 includes top terms like ‘prince’, ‘lady’, ‘door’, ‘youth’, ‘queen’ etc suggesting that the theme is related to king, queen or royal people.
- Topic 2 includes top terms like ‘sister’, ‘witch’, ‘brothers’, ‘morning’, etc suggesting that the theme is somewhat related to siblings.
- Topic 3 includes top terms like ‘queen’, ‘miranda’, ‘bride’, ‘woman’ etc suggesting that the theme is somewhat related to feminine gender and royal people.
- Topic 4 includes top terms like ‘fairy’, ‘desert’, ‘dwarf’, ‘queen’, ‘madam’ etc suggesting that the theme is related to royal people and fairy tale characters.
- Topic 5 includes top terms like ‘maid’, ‘jamila’, ‘prince’ etc suggesting that the theme is somewhat related to lower class people. But this is not guaranteed.
- The overall topics are distinct but there is not a specific distinct theme that we can identify each topic with.

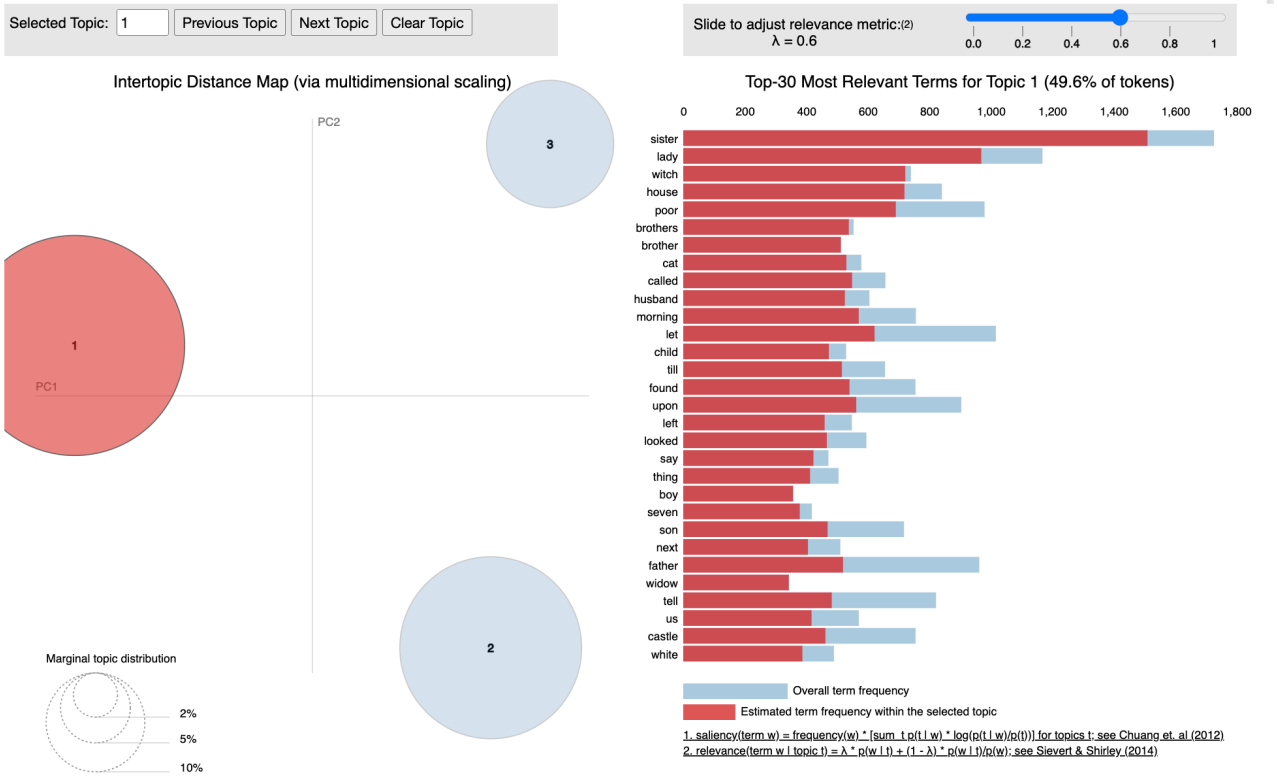
3 Topics for males using LDA

Top 20 words under the 3 topics

```
i=0
for i, topic in lda_model_males_3.show_topics(num_words=20):
    print(f"TOPIC {i+1}: \n\n", topic, "\n\n-------------------------\n")
    i+=1
```

```
TOPIC 1:

 0.012*"sister" + 0.008*"lady" + 0.006*"witch" + 0.006*"house" + 0.005*"poor" + 0.005*"let" + 0.005*"morning" + 0.004*"upon" + 0.004*"called" + 0.004*"found"
+ 0.004*"brothers" + 0.004*"cat" + 0.004*"husband" + 0.004*"father" + 0.004*"till" + 0.004*"two" + 0.004*"brother" + 0.004*"tell" + 0.004*"child" + 0.004*"so
n"

-------------------------

TOPIC 2:

 0.024*"fairy" + 0.014*"queen" + 0.012*"prince" + 0.008*"upon" + 0.008*"desert" + 0.006*"love" + 0.006*"help" + 0.005*"dwarf" + 0.005*"cried" + 0.005*"prett
y" + 0.005*"quite" + 0.005*"tell" + 0.005*"make" + 0.005*"know" + 0.004*"even" + 0.004*"bellissima" + 0.004*"jamila" + 0.004*"replied" + 0.004*"nurse" + 0.00
4*"without"

-------------------------

TOPIC 3:

 0.022*"queen" + 0.012*"maid" + 0.010*"prince" + 0.008*"golden" + 0.006*"cried" + 0.005*"miranda" + 0.005*"shall" + 0.005*"two" + 0.005*"sheep" + 0.004*"fore
st" + 0.004*"life" + 0.004*"heart" + 0.004*"youth" + 0.004*"bride" + 0.004*"father" + 0.004*"love" + 0.004*"though" + 0.004*"last" + 0.004*"moment" + 0.004
*"care"
```

```
pyLDAvis.save_html(vis_data, 'lda_visualization_males_3.html')
pyLDAvis.display(vis_data)
```



The 3 topics for males using sentences are shown as above. Keeping $\lambda = 0.6$ this helps us in selecting words that are relevant to that topic to give more importance to ratio of frequency of word given topic to the overall frequency of word.

- We observe that topic 1, topic 2, topic 3 correspond to 49.6%, 33.9%, 16.6% of the tokens in the corpus respectively.

- All other topics are well separated from each other and have clear distinction in terms of the inter topic distance.
- Topic 1 includes top terms like ‘sister’, ‘lady’, ‘witch’, ‘house’, ‘poor’ etc suggesting that the theme is related to common people and cheap items.
- Topic 2 includes top terms like ‘queen’, ‘maid’, ‘golden’, ‘prince’, ‘gold’ etc suggesting that the theme is somewhat related to royal people and expensive items.
- Topic 3 includes top terms like ‘fairy’, ‘desert’, ‘prince’, ‘dwarf’ etc suggesting that the theme is somewhat related to fairy tale characters..
- The overall topics are distinct and here we have specific distinct theme that we can identify each topic with - like royal people, common people, and fairy tale characters.

10 Topics for males using LDA

LDA topic modeling for male characters for 10 topics

```
lda_model_males_10 = models.LdaModel(corpus, num_topics=10, id2word=dictionary, passes=15, random_state=42)
```

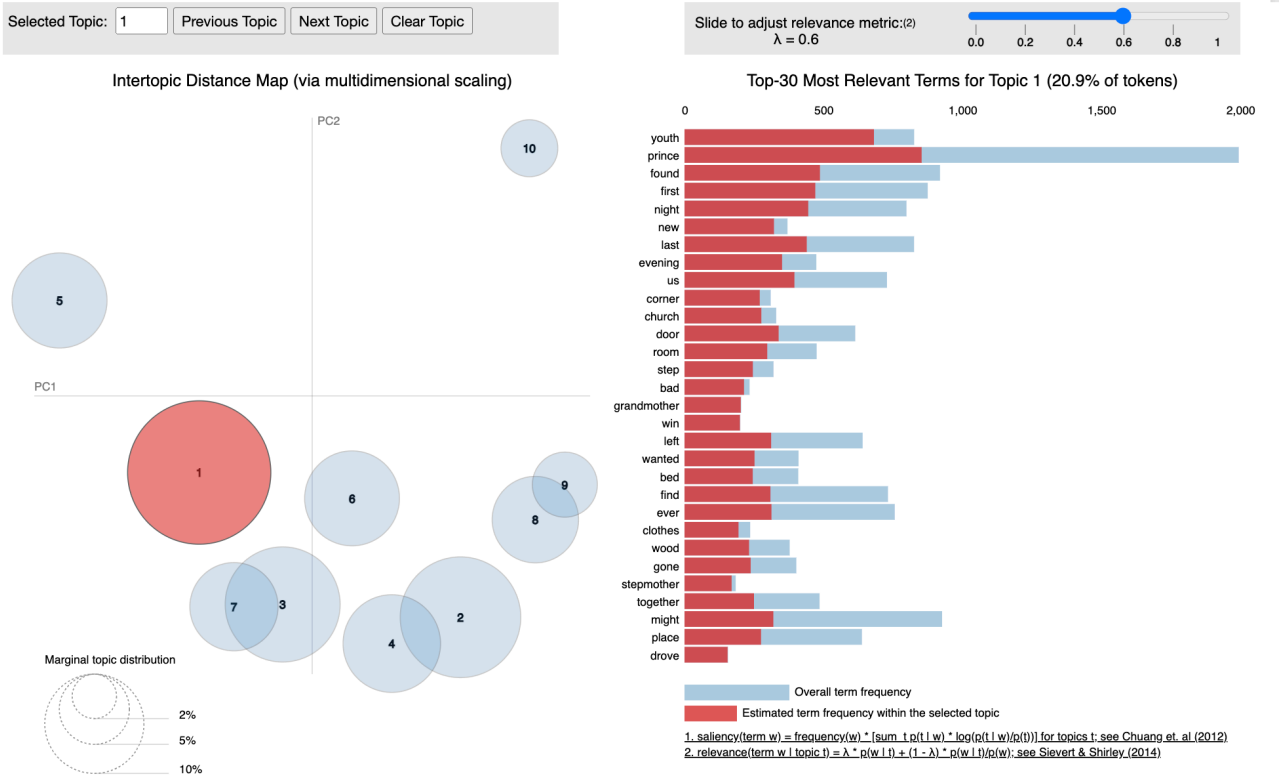
Top 20 words under the 10 topics

```
i=0
for i, topic in lda_model_males_10.show_topics(num_words=20):
    print(f"TOPIC {i+1}: \n\n", topic, "\n\n----------------------------\n")
    i+=1
```

```
TOPIC 1:

 0.030*"sister" + 0.016*"witch" + 0.012*"brothers" + 0.011*"brother" + 0.010*"let" + 0.009*"morning" + 0.008*"castle" + 0.008*"cat" + 0.008*"called" + 0.008
*"till" + 0.007*"negro" + 0.007*"tell" + 0.007*"udea" + 0.007*"seven" + 0.007*"dear" + 0.007*"father" + 0.006*"know" + 0.006*"next" + 0.006*"camel" + 0.006
*"child"

----------------------------

TOPIC 2:

 0.011*"get" + 0.010*"tower" + 0.010*"might" + 0.009*"palace" + 0.008*"hand" + 0.008*"say" + 0.007*"put" + 0.007*"pretty" + 0.007*"seen" + 0.006*"queen" + 0.
006*"poor" + 0.006*"thinking" + 0.006*"sat" + 0.006*"quite" + 0.006*"bride" + 0.006*"may" + 0.006*"gold" + 0.005*"high" + 0.005*"bear" + 0.005*"wished"

----------------------------

TOPIC 3:

 0.087*"queen" + 0.011*"children" + 0.010*"two" + 0.010*"bride" + 0.009*"land" + 0.009*"three" + 0.009*"life" + 0.008*"heart" + 0.007*"shall" + 0.007*"hand"
+ 0.006*"people" + 0.006*"opened" + 0.005*"john" + 0.005*"palace" + 0.005*"know" + 0.005*"trusty" + 0.005*"son" + 0.005*"suddenly" + 0.005*"stone" + 0.005*"t
urned"

----------------------------

TOPIC 4:

 0.024*"mermaid" + 0.024*"mistress" + 0.018*"master" + 0.012*"basket" + 0.012*"dog" + 0.009*"house" + 0.009*"son" + 0.008*"upon" + 0.008*"may" + 0.008*"gold"
+ 0.008*"lady" + 0.007*"elsa" + 0.007*"thing" + 0.007*"knight" + 0.007*"side" + 0.007*"help" + 0.007*"something" + 0.007*"hand" + 0.007*"together" + 0.006*"s
et"

----------------------------

TOPIC 5:

 0.014*"prince" + 0.012*"youth" + 0.008*"found" + 0.008*"first" + 0.008*"night" + 0.007*"last" + 0.007*"us" + 0.006*"evening" + 0.006*"door" + 0.005*"new" +
0.005*"might" + 0.005*"ever" + 0.005*"left" + 0.005*"find" + 0.005*"room" + 0.005*"church" + 0.005*"place" + 0.005*"corner" + 0.004*"wanted" + 0.004*"togethe
r"
```

```
TOPIC 6:

 0.049*"fairy" + 0.013*"queen" + 0.013*"desert" + 0.009*"dwarf" + 0.009*"love" + 0.009*"cried" + 0.008*"make" + 0.008*"help" + 0.007*"upon" + 0.007*"bellissima" + 0.007*"really" + 0.007*"quite" + 0.006*"poor" + 0.006*"pretty" + 0.006*"die" + 0.006*"madam" + 0.006*"married" + 0.005*"tell" + 0.005*"replied" + 0.005*"angry"

--------------------------

TOPIC 7:

 0.030*"maid" + 0.016*"golden" + 0.014*"youngest" + 0.013*"prince" + 0.010*"forest" + 0.009*"care" + 0.008*"second" + 0.008*"sister" + 0.008*"trouble" + 0.008*"morning" + 0.008*"hair" + 0.008*"marry" + 0.007*"yes" + 0.006*"every" + 0.006*"named" + 0.006*"moment" + 0.006*"everything" + 0.006*"want" + 0.006*"stand" + 0.005*"search"

--------------------------

TOPIC 8:

 0.028*"lady" + 0.021*"prince" + 0.010*"jamila" + 0.007*"house" + 0.006*"thing" + 0.006*"thou" + 0.006*"might" + 0.006*"green" + 0.006*"widow" + 0.006*"nurse" + 0.006*"put" + 0.005*"upon" + 0.005*"son" + 0.005*"throne" + 0.005*"palace" + 0.005*"lovely" + 0.005*"brought" + 0.005*"feet" + 0.005*"deer" + 0.005*"head"

--------------------------

TOPIC 9:

 0.028*"two" + 0.018*"queen" + 0.014*"sack" + 0.012*"house" + 0.011*"upon" + 0.009*"husband" + 0.008*"lived" + 0.008*"sent" + 0.008*"cried" + 0.007*"mare" + 0.007*"angry" + 0.007*"blue" + 0.007*"meet" + 0.007*"married" + 0.007*"madam" + 0.006*"times" + 0.006*"screamed" + 0.006*"indeed" + 0.006*"box" + 0.005*"turritella"

--------------------------

TOPIC 10:

 0.013*"cried" + 0.008*"father" + 0.008*"shall" + 0.007*"miranda" + 0.007*"nothing" + 0.007*"sheep" + 0.006*"think" + 0.006*"husband" + 0.006*"dog" + 0.006*"love" + 0.005*"quite" + 0.005*"let" + 0.005*"two" + 0.005*"make" + 0.005*"poor" + 0.005*"way" + 0.004*"nurse" + 0.004*"soon" + 0.004*"water" + 0.004*"anything"
```

```
pyLDAvis.save_html(vis_data, 'lda_visualization_males_10.html')
pyLDAvis.display(vis_data)
```



The 10 topics for males using sentences are shown as above. Keeping λ = 0.6 this helps us in selecting words that are relevant to that topic to give more importance to ratio of frequency of word given topic to the overall frequency of word.

- We observe that topic 1, topic 2, topic 3, topic 4, topic 5, topic 6, topic 7, topic 8, topic 9, topic 10 correspond to 20.9%, 14.8%, 13.4%, 9.7%, 9.2%, 9.1%, 7.9%, 7.6%, 4.3%, 3.3% of the tokens in the corpus respectively.
- Topics 1, 5, 6, 10 have clear distinction from other topics and have no overlaps.
- Topics 2, 4 and 3, 7 and 8, 9 overlap.
- Topic 1 includes top terms like ‘youth’, ‘prince’, ‘found’, ‘first’, ‘night’ etc suggesting that the theme is related to heirs of the royal family and some sort of time information.
- Topic 2 includes top terms like ‘cried’, ‘miranda’, ‘sheep’, ‘father’, ‘think’ etc suggesting that the theme is somewhat related to humans and associated activities.
- Topic 3 includes top terms like ‘sister’, ‘witch’, ‘brothers’, ‘brother’ etc suggesting that the theme is somewhat related to siblings.
- Topic 4 includes top terms like ‘queen’, ‘children’, ‘land’, ‘bride’ etc suggesting that the theme is related to heirs of the royal people mostly feminine.
- Topic 5 includes top terms like ‘tower’, ‘thinking’, ‘get’, ‘pretty’ etc suggesting that the theme is somewhat related to humans and associated activities.
- Topic 6 includes top terms like ‘lady’, ‘jamila’, ‘prince’, ‘deer’, ‘widow’ etc suggesting that the theme is somewhat related to feminine gender.
- Topic 7 includes top terms like ‘maid’, ‘golden’, ‘youngest’, ‘second’ etc suggesting opposite relationships in sense of maid and golden and youngest with second.
- Topic 8 includes top terms like ‘fairy’, ‘desert’, ‘dwarf’, ‘bellissima’ etc suggesting that the theme is somewhat related to fairytale characters.
- Topic 9 includes top terms like ‘sack’, ‘two’, ‘mare’, ‘screamed’ etc suggesting that the theme is somewhat related to common words.
- Topic 10 includes top terms like ‘mermaid’, ‘mistress’, ‘master’, ‘basket’ not giving a clear idea.
- The overall topics are not distinct and here we have a lot of difficulty in associating specific distinct theme for each topic.

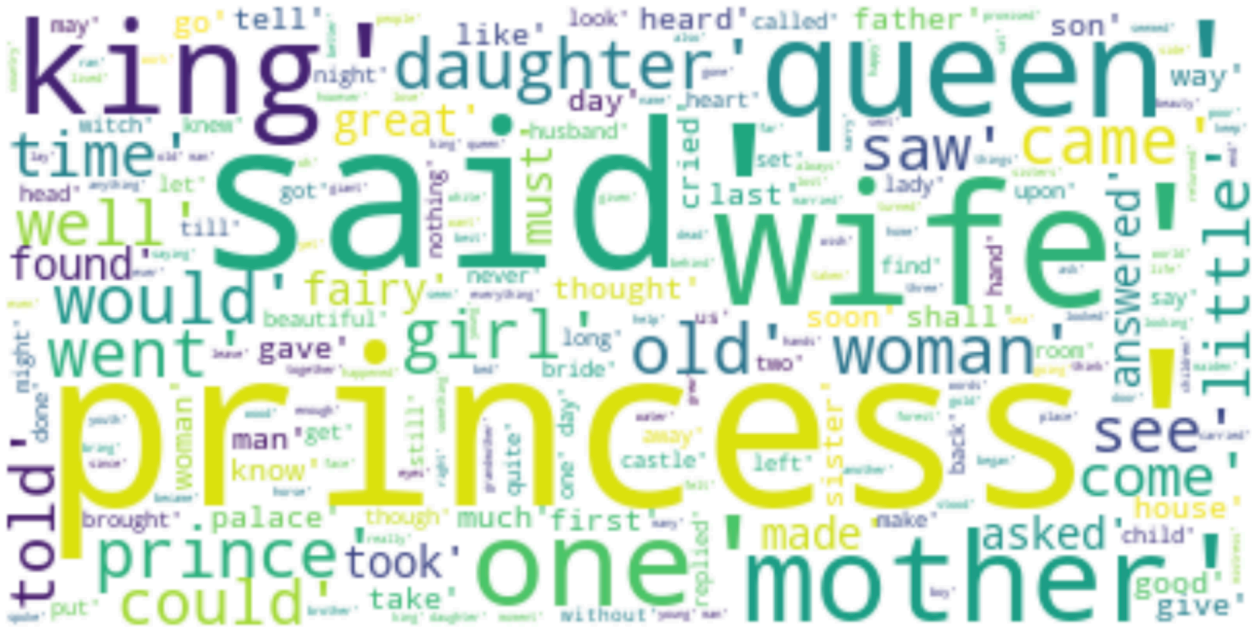
Based on this, we can understand that the number of topics need to be optimal for identifying distinct topics in LDA topic modelling. Too few topics may lead to underfitting and too many topics can lead to overfitting. Underfit models tend to group unrelated words together and overfit models can produce topics with many irrelevant words.

- In our case based on my assumption, the LDA model with 3 topics was better than 5 or 10 topics. The reason being, I was able to atleast identify the themes associated with the topics distinctly.
- The model with 10 topics was very difficult to work with and had many terms of opposing nature and unclear themes.
- The model with 5 topics was better than 10 topics in some sense.

Task 3: Compare topics between male and female characters

```
female_chars.loc[:, ('preprocessed')] = female_chars.loc[:, ('sentences')].apply(preprocess_sentences)
```

```
text = " ".join(str(i) for i in female_chars['preprocessed'])
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords, background_color="white").generate(text)
plt.figure(figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



From the word cloud we observe that for females, we can see that the top and most frequent words include ‘said’, ‘princess’, ‘wife’, ‘queen’, ‘mother’, ‘daughter’, ‘woman’, ‘girl’. This suggests that the key theme of this is on the **feminine** gender and there are some words like ‘king’ that are related to a female character.

```
female_chars.loc[:, ('preprocessed_sentences')] = female_chars.loc[:, ('sentences')].apply(preprocess_sentences)
```

```
female_chars.head(1)
```

	index	fairy_tale	country	character	gender	count_of_appearance	if_main(>=20)	sentences	sentences_list	preprocessed	preprocessed_sentences
0	0	cannetella	NaN	wife	female	3	0	but at last, after he had been married for many years, and was quite an old man, his wife renzolla presented him with a fine daughter, whom they called cannetella.if you will step in here you shall have my daughter for a wife, and i will send attendants with her, and as many horses and servants as you wish.?€?\n?€?a thousand thanks,?€? returned scioravante, ?€?i shall be delighted to marry your daughter, but it is quite unnecessary to send anyone to accompany her.scioravante was furious when he heard this, and seizing a huge knife from his pocket he threatened to kill his wife for her disobedience.	['but at last, after he had been married for many years, and was quite an old man, his wife renzolla presented him with a fine daughter, whom they called cannetella.', 'if you will step in here you shall have my daughter for a wife, and i will send attendants with her, and as many horses and servants as you wish.?€?\n?€?a thousand thanks,?€? returned scioravante, ?€?i shall be delighted to marry your daughter, but it is quite unnecessary to send anyone to accompany her.', 'scioravante was furious when he heard this, and seizing a huge knife from his pocket he threatened to kill his wife for her disobedience.']	[last, married, many, years, quite, old, man, wife, renzolla, presented, fine, daughter, called, cannetella, step, shall, daughter, wife, send, attendants, many, horses, servants, wish, thousand, thanks, returned, scioravante, shall, delighted, marry, daughter, quite, unnecessary, send, anyone, accompany, scioravante, furious, heard, seizing, huge, knife, pocket, threatened, kill, wife, disobedience]	[last, married, many, years, quite, old, man, wife, renzolla, presented, fine, daughter, called, cannetella, step, shall, daughter, wife, send, attendants, many, horses, servants, wish, thousand, thanks, returned, scioravante, shall, delighted, marry, daughter, quite, unnecessary, send, anyone, accompany, scioravante, furious, heard, seizing, huge, knife, pocket, threatened, kill, wife, disobedience]

Examples of preprocessed sentences

```
cannetella = female_chars[(female_chars.character == 'wife') & (female_chars.fairy_tale == 'cannetella')]
cannetella
```

	index	fairy_tale	country	character	gender	count_of_appearance	if_main(>=20)	sentences	sentences_list	preprocessed	preprocessed_sentences
0	0	cannetella	NaN	wife	female	3	0	but at last, after he had been married for many years, and was quite an old man, his wife renzolla presented him with a fine daughter, whom they called cannetella.if you will step in here you shall have my daughter for a wife, and i will send attendants with her, and as many horses and servants as you wish.?€?\n?€?a thousand thanks,?€? returned scioravante, ?€?i shall be delighted to marry your daughter, but it is quite unnecessary to send anyone to accompany her.scioravante was furious when he heard this, and seizing a huge knife from his pocket he threatened to kill his wife for her disobedience.	['but at last, after he had been married for many years, and was quite an old man, his wife renzolla presented him with a fine daughter, whom they called cannetella.', 'if you will step in here you shall have my daughter for a wife, and i will send attendants with her, and as many horses and servants as you wish.?€?\n?€?a thousand thanks,?€? returned scioravante, ?€?i shall be delighted to marry your daughter, but it is quite unnecessary to send anyone to accompany her.', 'scioravante was furious when he heard this, and seizing a huge knife from his pocket he threatened to kill his wife for her disobedience.']	[last, married, many, years, quite, old, man, wife, renzolla, presented, fine, daughter, called, cannetella, step, shall, daughter, wife, send, attendants, many, horses, servants, wish, thousand, thanks, returned, scioravante, shall, delighted, marry, daughter, quite, unnecessary, send, anyone, accompany, scioravante, furious, heard, seizing, huge, knife, pocket, threatened, kill, wife, disobedience]	[last, married, many, years, quite, old, man, wife, renzolla, presented, fine, daughter, called, cannetella, step, shall, daughter, wife, send, attendants, many, horses, servants, wish, thousand, thanks, returned, scioravante, shall, delighted, marry, daughter, quite, unnecessary, send, anyone, accompany, scioravante, furious, heard, seizing, huge, knife, pocket, threatened, kill, wife, disobedience]

Preparing the data for LDA Topic Modeling and an example

BOW representation

```
dictionary = corpora.Dictionary(female_chars['preprocessed_sentences'])
dictionary.filter_extremes(no_below=5, no_above=0.2, keep_n=10000)

corpus = [dictionary.doc2bow(sentence) for sentence in female_chars['preprocessed_sentences']]
```

Restricting top 10K terms that appear in 5 documents and not more that 20% documents.

5 Topics for females using LDA

LDA Modeling for female characters for 5 topics

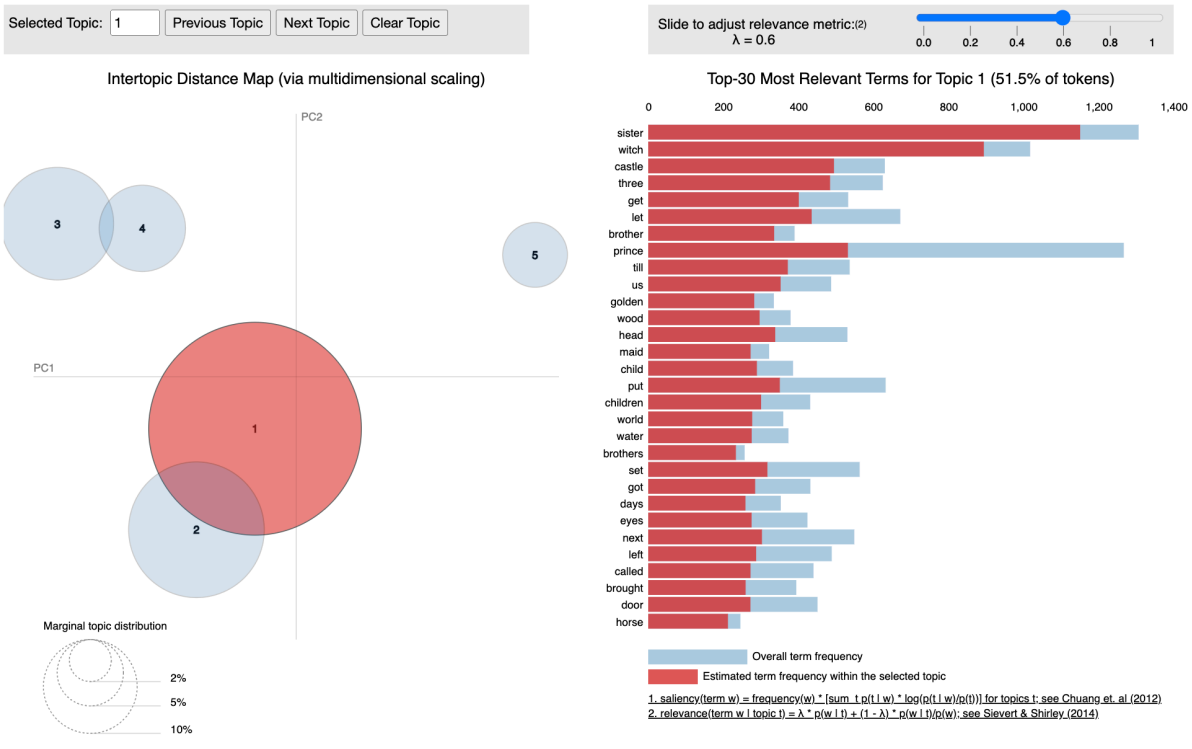
```
lda_model_females_5 = models.LdaModel(corpus, num_topics=5, id2word=dictionary, passes=15, random_state=42)
```

Top 20 topics under 5 topics

```
for i, topic in lda_model_females_5.show_topics(num_words=20):
    print(f"TOPIC: \n\n", topic, "\n\n-------------------------\n")
```

```
TOPIC:

 0.020*"catherine" + 0.016*"lady" + 0.016*"fairy" + 0.013*"prince" + 0.009*"poor" + 0.009*"cat" + 0.007*"destiny" + 0.007*"mistress" + 0.007*"step" + 0.007
*"happy" + 0.006*"really" + 0.006*"father" + 0.005*"hand" + 0.005*"marry" + 0.005*"grandmother" + 0.005*"hardly" + 0.005*"nose" + 0.005*"dear" + 0.005*"hous
e" + 0.005*"sister"

-------------------------

TOPIC:

 0.023*"fairy" + 0.016*"prince" + 0.007*"nothing" + 0.007*"step" + 0.006*"without" + 0.006*"celandine" + 0.006*"life" + 0.006*"even" + 0.006*"daughters" + 0.
005*"woods" + 0.005*"palace" + 0.005*"since" + 0.005*"first" + 0.004*"charming" + 0.004*"house" + 0.004*"make" + 0.004*"every" + 0.004*"always" + 0.004*"migh
t" + 0.004*"madam"

-------------------------

TOPIC:

 0.012*"sister" + 0.009*"witch" + 0.006*"prince" + 0.005*"castle" + 0.005*"three" + 0.005*"let" + 0.004*"get" + 0.004*"till" + 0.004*"us" + 0.004*"put" + 0.0
04*"head" + 0.004*"brother" + 0.003*"set" + 0.003*"next" + 0.003*"children" + 0.003*"wood" + 0.003*"child" + 0.003*"left" + 0.003*"got" + 0.003*"soon"

-------------------------

TOPIC:

 0.012*"aladdin" + 0.009*"palace" + 0.009*"son" + 0.007*"tell" + 0.006*"sultan" + 0.006*"husband" + 0.006*"night" + 0.005*"youngest" + 0.005*"first" + 0.005
*"round" + 0.004*"next" + 0.004*"father" + 0.004*"sisters" + 0.004*"set" + 0.004*"sitting" + 0.004*"put" + 0.004*"bring" + 0.004*"says" + 0.004*"eldest" + 0.
004*"want"

-------------------------

TOPIC:

 0.028*"elsa" + 0.016*"cat" + 0.014*"witch" + 0.014*"house" + 0.013*"stepmother" + 0.012*"bear" + 0.012*"lady" + 0.009*"cow" + 0.008*"eat" + 0.008*"bell" +
0.008*"stable" + 0.008*"dog" + 0.007*"thinking" + 0.007*"ought" + 0.007*"fat" + 0.007*"dish" + 0.007*"leaf" + 0.007*"whether" + 0.007*"ate" + 0.007*"orchard"

-------------------------
```

```
pyLDAvis.save_html(vis_data, 'lda_visualization_females_5.html')
pyLDAvis.display(vis_data)
```



The 5 topics for females using sentences are shown as above. Keeping λ = 0.6 this helps us in selecting words that are relevant to that topic to give more importance to ratio of frequency of word given topic to the overall frequency of word.

- We observe that topic 1, topic 2, topic 3, topic 4, topic 5 correspond to 51.5%, 20.9%, 14.4%, 8.5%, and 4.8% of the tokens in the corpus respectively.
- Topic 1 is separated from topics 3, 4 and 5 but overlaps with topic 2.
- Topic 3 and 4 show some overlap.
- Topic 1 includes top terms like ‘sister’, ‘witch’, ‘castle’, ‘three’, ‘get’, ‘let’ etc suggesting that the theme is related to common people and female gender.
- Topic 2 includes top terms like ‘aladdin’, ‘palace’, ‘son’, ‘sultan’ etc suggesting that the theme is somewhat related to royal male people and expensive items.
- Topic 3 includes top terms like ‘fairy’, ‘prince’, ‘celandine’, ‘step’, ‘woods’ etc suggesting that the theme is somewhat related to fairytale characters.
- Topic 4 includes top terms like ‘catherine’, ‘lady’, ‘destiny’, ‘fairy’, ‘madam’ etc suggesting that the theme is related to royal people and fairy tale characters.
- Topic 5 includes top terms like ‘elsa’, ‘stepmother’, ‘cat’, ‘bear’, ‘fox’ etc suggesting that the theme is somewhat related to animals.
- The overall topics are distinct but there is not a specific distinct theme that we can identify each topic with. There are a few topics that can be distinctly identifies with a key theme.

3 Topics for females using LDA

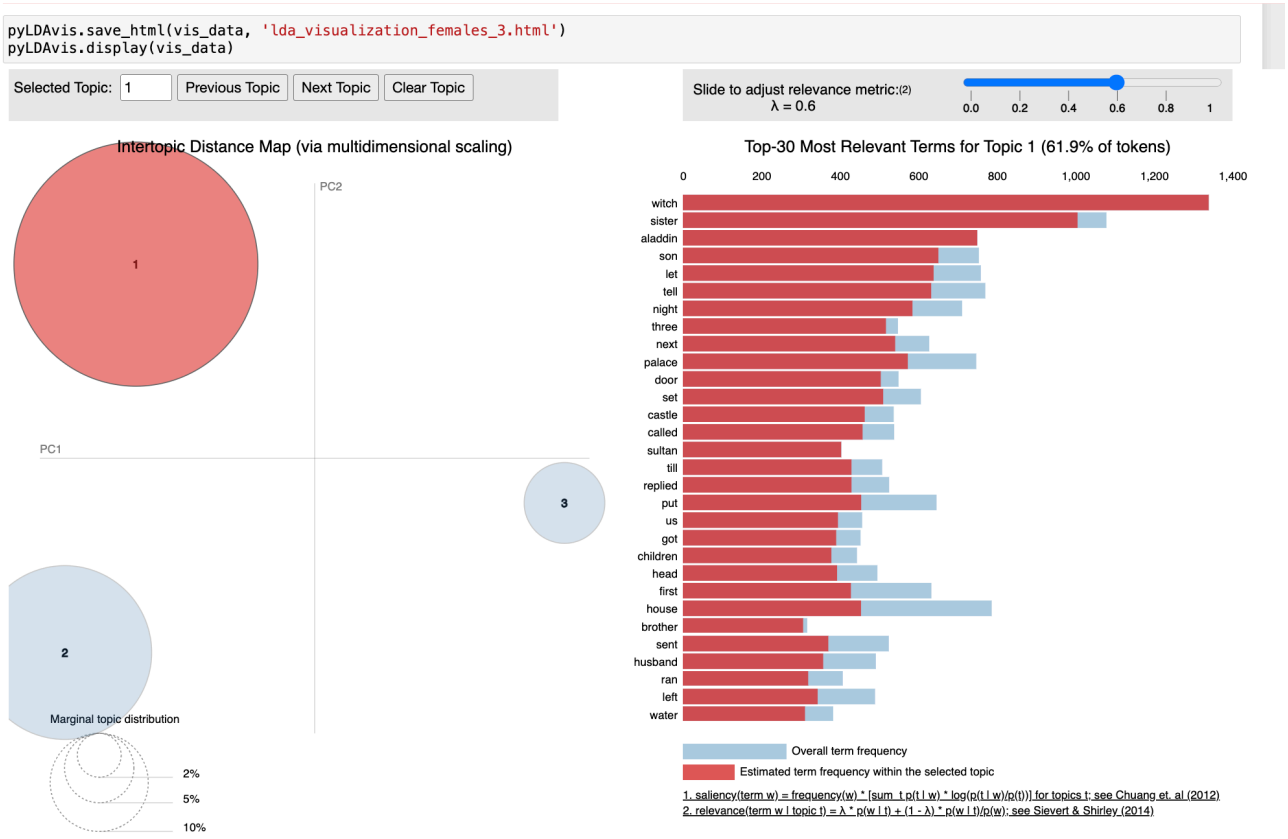
LDA Modeling for female characters for 3 topics

```
lda_model_females_3 = models.LdaModel(corpus, num_topics=3, id2word=dictionary, passes=15, random_state=42)
```

Top 20 words under the 3 topics

```
for i, topic in lda_model_females_3.show_topics(num_words=20):
    print(f"TOPIC: \n\n", topic, "\n\n----------------------\n")
```

```
TOPIC:

 0.018*"fairy" + 0.015*"prince" + 0.006*"step" + 0.005*"daughters" + 0.005*"nothing" + 0.004*"without" + 0.004*"began" + 0.004*"make" + 0.004*"really" + 0.004*"even" + 0.004*"might" + 0.004*"lived" + 0.004*"father" + 0.004*"quite" + 0.004*"every" + 0.004*"soon" + 0.004*"life" + 0.004*"poor" + 0.004*"care" + 0.004*"always"

----------------------

TOPIC:

 0.026*"elsa" + 0.025*"lady" + 0.018*"cat" + 0.014*"catherine" + 0.012*"house" + 0.011*"bear" + 0.011*"stepmother" + 0.008*"cow" + 0.007*"bell" + 0.007*"stable" + 0.007*"eat" + 0.007*"whether" + 0.006*"ought" + 0.006*"thinking" + 0.006*"stone" + 0.006*"fat" + 0.006*"dish" + 0.006*"ate" + 0.006*"leaf" + 0.006*"orchard"

----------------------

TOPIC:

 0.012*"witch" + 0.009*"sister" + 0.007*"aladdin" + 0.006*"son" + 0.006*"let" + 0.006*"tell" + 0.005*"night" + 0.005*"palace" + 0.005*"next" + 0.005*"three" + 0.004*"set" + 0.004*"door" + 0.004*"castle" + 0.004*"called" + 0.004*"put" + 0.004*"house" + 0.004*"replied" + 0.004*"till" + 0.004*"first" + 0.004*"sultan"
```



The 3 topics for females using sentences are shown as above. Keeping $\lambda = 0.6$ this helps us in selecting words that are relevant to that topic to give more importance to ratio of frequency of word given topic to the overall frequency of word.

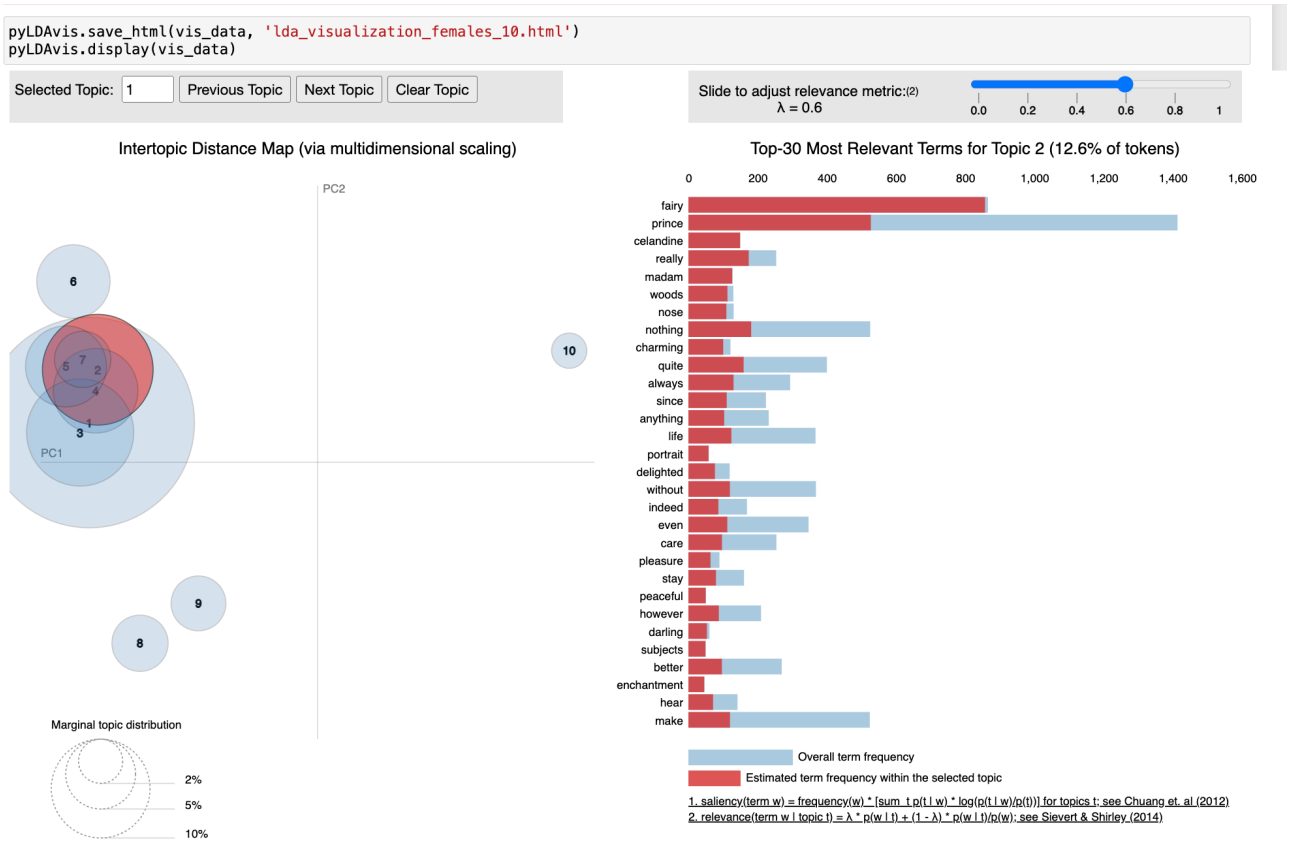
- We observe that topic 1, topic 2, topic 3 correspond to 61.9%, 31.3%, 6.8% of the tokens in the corpus respectively.
- All the topics are clearly separated from each other based on inter topic distance.
- Topic 1 includes top terms like ‘witch’, ‘sister’, ‘aladdin’, ‘son’ etc suggesting that the theme is related to common people of both gender.
- Topic 2 includes top terms like ‘fairy’, ‘prince’, ‘daughters’, ‘step’ etc suggesting that the theme is somewhat related to royal heirs.
- Topic 3 includes top terms like ‘elsa’, ‘lady’, ‘cat’, ‘bear’, ‘cow’ etc suggesting that the theme is somewhat related to animals.
- The overall topics are distinct and here we can have a specific distinct theme that we can identify each topic with.
- There are a few topics that can be distinctly identifies with a key theme. Like topic 3 is related to animals.

10 Topics for females using LDA

Top 20 words under the topic

```
for i, topic in lda_model_females_10.show_topics(num_words=20):
    print(f"TOPIC: \n\n", topic, "\n\n-------------------------\n")
```

```
TOPIC:

 0.028*"catherine" + 0.016*"daughters" + 0.016*"lady" + 0.010*"destiny" + 0.010*"poor" + 0.009*"father" + 0.008*"happy" + 0.008*"sister" + 0.008*"grandmothe
r" + 0.007*"step" + 0.007*"mistress" + 0.007*"court" + 0.007*"going" + 0.006*"hand" + 0.006*"marry" + 0.006*"lived" + 0.006*"everything" + 0.005*"suffered" +
0.005*"house" + 0.005*"turned"

-------------------------

TOPIC:

 0.042*"lady" + 0.014*"maid" + 0.011*"bride" + 0.009*"hand" + 0.008*"golden" + 0.006*"place" + 0.006*"men" + 0.006*"house" + 0.006*"years" + 0.006*"elsa" +
0.006*"first" + 0.006*"heart" + 0.005*"hair" + 0.005*"village" + 0.005*"make" + 0.005*"sent" + 0.005*"girls" + 0.005*"many" + 0.004*"lovely" + 0.004*"silver"

-------------------------

TOPIC:

 0.009*"prince" + 0.007*"sister" + 0.005*"castle" + 0.005*"till" + 0.004*"us" + 0.004*"soon" + 0.004*"head" + 0.004*"three" + 0.004*"father" + 0.004*"get" +
0.003*"child" + 0.003*"let" + 0.003*"world" + 0.003*"put" + 0.003*"brought" + 0.003*"heart" + 0.003*"got" + 0.003*"brother" + 0.003*"poor" + 0.003*"tell"

-------------------------

TOPIC:

 0.023*"aladdin" + 0.016*"palace" + 0.014*"son" + 0.012*"sultan" + 0.008*"set" + 0.007*"first" + 0.007*"night" + 0.006*"magician" + 0.006*"father" + 0.006*"s
ays" + 0.006*"next" + 0.006*"carried" + 0.006*"thou" + 0.005*"vizier" + 0.005*"saying" + 0.005*"youngest" + 0.005*"three" + 0.005*"put" + 0.005*"bring" + 0.0
05*"ask"

-------------------------

TOPIC:

 0.017*"step" + 0.013*"soldier" + 0.011*"dog" + 0.010*"night" + 0.009*"husband" + 0.007*"began" + 0.007*"morning" + 0.007*"ladies" + 0.006*"looked" + 0.006
*"put" + 0.006*"people" + 0.006*"house" + 0.006*"fell" + 0.005*"moment" + 0.005*"sisters" + 0.005*"master" + 0.005*"things" + 0.005*"broom" + 0.005*"door" +
0.005*"many"

-------------------------

TOPIC:

 0.037*"fairy" + 0.023*"prince" + 0.008*"nothing" + 0.008*"really" + 0.007*"quite" + 0.006*"celandine" + 0.006*"always" + 0.005*"madam" + 0.005*"life" + 0.00
5*"make" + 0.005*"without" + 0.005*"palace" + 0.005*"woods" + 0.005*"even" + 0.005*"since" + 0.005*"nose" + 0.005*"love" + 0.004*"anything" + 0.004*"charmin
g" + 0.004*"might"

-------------------------

TOPIC:

 0.060*"cat" + 0.017*"hardly" + 0.016*"fast" + 0.015*"broken" + 0.014*"sisters" + 0.013*"boy" + 0.010*"far" + 0.010*"fish" + 0.008*"tell" + 0.008*"every" +
0.007*"grew" + 0.007*"morning" + 0.007*"troll" + 0.007*"hut" + 0.007*"sister" + 0.007*"wood" + 0.007*"tree" + 0.007*"fat" + 0.006*"fire" + 0.006*"eldest"

-------------------------

TOPIC:

 0.087*"witch" + 0.015*"called" + 0.014*"let" + 0.012*"door" + 0.012*"house" + 0.012*"children" + 0.010*"wood" + 0.010*"whereas" + 0.010*"dog" + 0.010*"tell"
+ 0.009*"kill" + 0.008*"water" + 0.008*"replied" + 0.008*"casket" + 0.008*"wicked" + 0.007*"bread" + 0.007*"mistress" + 0.007*"sister" + 0.007*"poor" + 0.007
*"kiss"

-------------------------

TOPIC:

 0.050*"elsa" + 0.026*"stepmother" + 0.015*"sister" + 0.013*"nothing" + 0.012*"want" + 0.009*"let" + 0.008*"husband" + 0.008*"fisherman" + 0.008*"next" + 0.0
07*"night" + 0.007*"child" + 0.007*"brought" + 0.007*"first" + 0.007*"emperor" + 0.006*"live" + 0.006*"better" + 0.006*"till" + 0.006*"called" + 0.005*"hand"
+ 0.005*"make"

-------------------------

TOPIC:

 0.042*"bear" + 0.028*"bell" + 0.027*"stable" + 0.027*"cow" + 0.026*"house" + 0.024*"thinking" + 0.024*"eat" + 0.024*"fat" + 0.023*"ought" + 0.023*"dish" +
0.023*"leaf" + 0.023*"whether" + 0.023*"orchard" + 0.022*"ate" + 0.022*"potful" + 0.021*"fox" + 0.020*"pile" + 0.020*"stone" + 0.020*"squirrel" + 0.020*"wol
f"
```



The 10 topics for males using sentences are shown as above. Keeping $\lambda = 0.6$ this helps us in selecting words that are relevant to that topic to give more importance to ratio of frequency of word given topic to the overall frequency of word.

- We observe that topic 1, topic 2, topic 3, topic 4, topic 5, topic 6, topic 7, topic 8, topic 9, topic 10 correspond to 45.3%, 12.6%, 11.7%, 7.3%, 6.7%, 5.5%, 3.3%, 3.2%, 3.1%, 1.3% of the tokens in the corpus respectively.
- Topics 6, 8, 9, 10 have clear distinction from other topics and have no overlaps.
- Topics 1, 2, 3, 4, 5, 7 overlap a lot with topics within one another.
- Topic 1 includes top terms like ‘prince’, ‘sister’, ‘castle’ etc suggesting that the theme is related to heirs of the royal family and expensive items.
- Topic 2 includes top terms like ‘fairy’, ‘prince’, ‘celandine’ etc suggesting that the theme is somewhat related to fairytale characters.
- Topic 3 includes top terms like ‘aladdin’, ‘palace’, ‘sultan’, ‘son’ etc suggesting that the theme is somewhat related to male royal characters.
- Topic 4 includes top terms like ‘soldier’, ‘step’, ‘dog’, ‘ladies’ etc suggesting that the theme is somewhat related to common people.
- Topic 5 includes top terms like ‘catherine’, ‘daughters’, ‘lady’ etc suggesting that the theme is somewhat related to female characters.
- Topic 6 includes top terms like ‘lady’, ‘maid’, ‘bride’ etc suggesting that the theme is somewhat related to feminine gender of common people.
- Topic 7 includes top terms like ‘elsa’, ‘stepmother’, ‘fisherman’, ‘emperor’ etc with no distinct idea.

- Topic 8 includes top terms like ‘witch’, ‘whereas’, ‘casket’, ‘granny’, ‘mistress’ etc suggesting that the theme is somewhat related to female gender.
- Topic 9 includes top terms like ‘cat’, ‘broken’, ‘hardly’, ‘fast’ giving an idea about common words in normal life.
- Topic 10 includes top terms like ‘bear’, ‘bell’, ‘squirrel’, ‘fox’ etc suggesting that the theme is somewhat related to animals.
- The overall topics are not distinct and here we have a lot of difficulty in associating specific distinct theme for each topic.

Based on this, we can understand that the number of topics need to be optimal for identifying distinct topics in LDA topic modelling. Too few topics may lead to underfitting and too many topics can lead to overfitting. Underfit models tend to group unrelated words together and overfit models can produce topics with many irrelevant words.

- In our case based on my assumption, the LDA model with 3 topics was better than 5 or 10 topics. The reason being, I was able to atleast identify the themes associated with the topics distinctly.
- The model with 10 topics was very difficult to work with and had many terms of opposing nature and unclear themes.
- The model with 5 topics was better than 10 topics in some sense.

Performing LDA topic modeling on the male and female characters using sentences involved steps and we got some interesting topics corresponding to the male and female characters.

- We understood that optimal number of topics helps in identifying clear set of themes for the topics. - In our case it was 3 topics LDA model where we were able to get separate ideas for the topics and terms.
- Even though we separated the dataset for male and female characters, we obtained words from both genders for both the male and female topics. This is obvious considering he fact that male and female characters are related to each other and then sentences consisted of some relationships between them.
- The major terms for females included ‘prince’, ‘emperor’, ‘sultan’, ‘youth’. In fact, we obtained a lot of female terms associated with males like ‘princess’, ‘fairy’, ‘queen’, ‘daughters’. This is somewhat strange as the dataset was mostly for males.
- The major terms for males included ‘princess’, ‘queen’, ‘daughter’, ‘maid’, ‘fairy’ and some proper nouns like ‘catherine’, ‘celandine’ . In fact, we obtained some male terms associated with males like ‘prince’, ‘son’, ‘brother’, ‘sultan’, ‘aladdin’.
- Overall, we have a decent quality of topics that is not of very high quality for males and females as the words do not clearly help us in identifying the key ideas.
- Overall, we have a decent quality of topics that is not of very high quality for males and females as the words do not clearly help us in identifying the key ideas. LDA model with loess topics have a broader domain and can successfully separate the ideas.
- The LDA models with more number of topics tend to perform poor due to overfitting because a lot of words get repeated in multiple topics, thus losing the essence of a central idea specific to the topic.

Task 4: Run topic modeling on events instead of sentences

Male Events

```
events_male.head(1)
```

	fairy_tale	character	character_sentences	events	gender	sentences	preprocessed_sentences
0	adventures_of_an_indian_brave	man	a long, long way off, right away in the west of america, there once lived an old man who had one son. the country round was covered with forests, in which dwelt all kinds of wild beasts, and the young man and his companions used to spend whole days in hunting them, and he was the finest hunter of all the tribe. the old man and the wife, however, would not go out, but remained in the wigwam making bows and arrows. it soon grew so cold in the forest that at last one of the men declared they could walk no more, unless they could manage to warm themselves. then they started off to the place where the goats and deer were to be found in the greatest numbers, and soon had killed as many as they wanted. but when they reached a great river the young man did not want the trouble of carrying his pack any further, and left it on the bank. asked the old man, as his son opened the door. ' no; i have slain enough to feast us for many moons, but it was heavy, and i left the pack on the bank of the great river. so the old man rose and went, and strapped the meat on his shoulder; but as he was crossing the ford the strap broke and the pack fell into the river. but by this time it had lost all likeness to a man, and was changed into a piece of wood. there it was borne by the current close to the shore, and a woman who was down there washing her clothes caught it as it passed, and drew it out, saying to herself:' what a nice smooth plank! the woman had been working hard all day and was very hungry, so she took her biggest spoon and plunged it into the pot. the woman had been surprised before at the disappearance of her food, but she was more astonished still when, instead of the plank, she beheld a baby. the baby grew and throve as no baby in that country had ever done, and in four days he was a man, and as tall and strong as any brave of the tribe. it took him many days to get there, and when he saw his son sitting in his place his anger was kindled, and his heart was stirred to take vengeance upon him. the two went out together, and after walking for about half an hour they old man stopped. the higher the young man climbed the higher the birds seemed to be, and when he looked down the earth below appeared no bigger than a star. he crept up to them on tiptoe, and when one old woman passed her dinner to the other he held out his hand and took it and ate if for himself. ' how slow you are kneading that cake,' cried the other old woman at last. and again the young man stretched out his hand; and the two old women fell to quarrelling afresh. but when it happened for the third time the old women suspected some trick, and one of them exclaimed:' i am sure there is a man here; tell me, are you not my grandson?' ' yes,' answered the young man, who wished to please her,' and in return for your good dinner i will see if i can not restore your sight; for i was taught in the art of healing by the best medicine man in the tribe.' there was no night in that country, so, instead of going to bed very early, as he would have done in his own hut, the young man took another walk. and they wove him the net he asked for, and for many weeks he watched by the river, only going back to the old women when he wanted a fish cooked. at last, one day, when he was eating his dinner, the old woman who always spoke first, said to him:' we have been very glad to see you, grandson, but now it is time that you went home.' but when the basket did stop, the young man forgot what he had been told, and put his head out to see what was the matter. this time the young man was wiser, and though the basket often stopped, and strange creatures seemed to rest on him and to pluck at his blanket, he held it tight till he heard the crow calling. and, to satisfy him, the woman turned round and perceived her husband.	dwell make remain grow leave reach carry ask open leave break go strap cross fell rise change lose take thrive grow take stop go walk look seem climb hold take fell stretch exclaim happen suspect answer wish teach go take ask go go put stop see forget tell hear hold seem stop pluck	male	a long, long way off, right away in the west of america, there once lived an old man who had one son. the country round was covered with forests, in which dwelt all kinds of wild beasts, and the young man and his companions used to spend whole days in hunting them, and he was the finest hunter of all the tribe. the old man and the wife, however, would not go out, but remained in the wigwam making bows and arrows. but when they reached a great river the young man did not want the trouble of carrying his pack any further, and left it on the bank. asked the old man, as his son opened the door. so the old man rose and went, and strapped the meat on his shoulder; but as he was crossing the ford the strap broke and the pack fell into the river. but by this time it had lost all likeness to a man, and was changed into a piece of wood. the baby grew and throve as no baby in that country had ever done, and in four days he was a man, and as tall and strong as any brave of the tribe. the two went out together, and after walking for about half an hour they old man stopped. the higher the young man climbed the higher the birds seemed to be, and when he looked down the earth below appeared no bigger than a star. and again the young man stretched out his hand; and the two old women fell to quarrelling afresh. but when it happened for the third time the old women suspected some trick, and one of them exclaimed: 'i am sure there is a man here; tell me, are you not my grandson?' 'yes,' answered the young man, who wished to please her, 'and in return for your good dinner i will see if i cannot restore your sight; for i was taught in the art of healing by the best medicine man in the tribe.' there was no night in that country, so, instead of going to bed very early, as he would have done in his own hut, the young man took another walk. but when the basket did stop, the young man forgot what he had been told, and put his head out to see what was the matter. this time the young man was wiser, and though the basket often stopped, and strange creatures seemed to rest on him and to pluck at his blanket, he held it tight till he heard the crow calling.	[dwell, make, remain, grow, leave, reach, carry, ask, open, leave, break, go, strap, cross, fell, rise, change, lose, take, thrive, grow, take, stop, go, walk, look, seem, climb, hold, take, fell, stretch, exclaim, happen, suspect, answer, wish, teach, go, take, ask, go, go, put, stop, see, forget, tell, hear, hold, seem, stop, pluck]

From the word cloud we observe that for males, we can see that the top and most frequent words include ‘come’, ‘take’, ‘go’, ‘say’, ‘make’. This suggests that the key theme of this is on the activity of achieving something or adventurous.

Preparing the data for LDA Topic Modeling and an example

```
dictionary = corpora.Dictionary(events_male['preprocessed_sentences'])
dictionary.filter_extremes(no_below=5, no_above=0.2, keep_n=10000)

corpus = [dictionary.doc2bow(sentence) for sentence in events_male['preprocessed_sentences']]
```

Restricting top 10K terms that appear in 5 documents and not more that 20% documents.

5 Topics for males using events

LDA Topic Modeling for Males using events and 5 topics

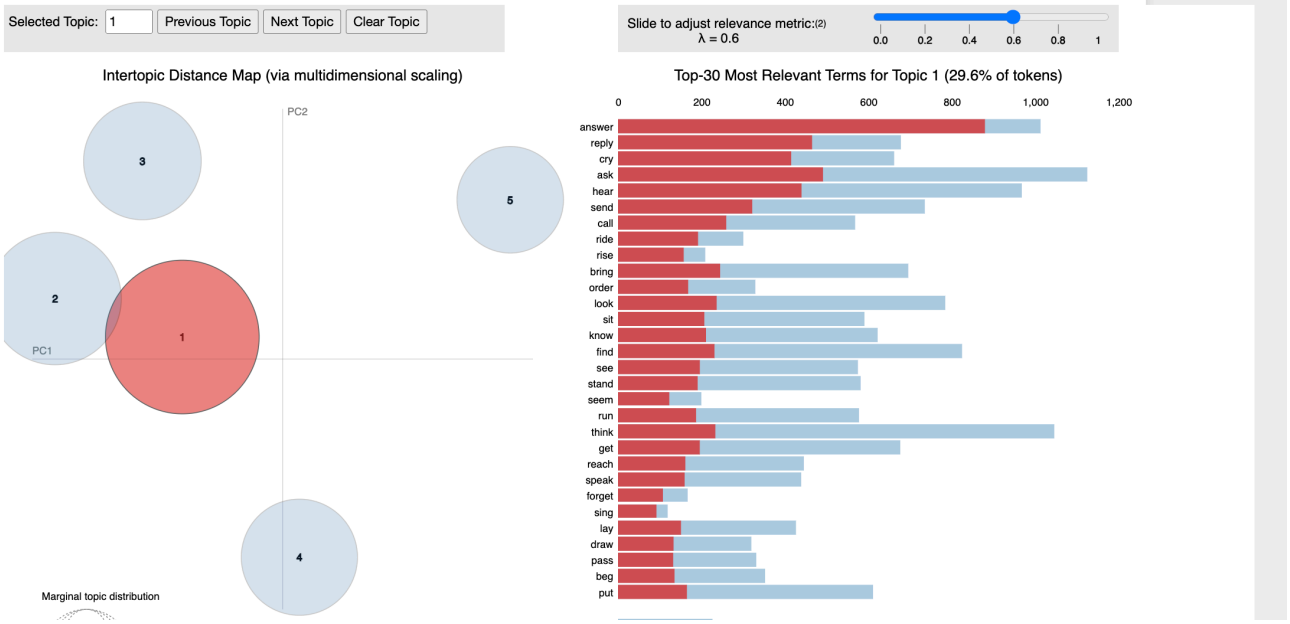
```
lda_model_events_males_5 = models.LdaModel(corpus, num_topics=5, id2word=dictionary, passes=15, random_state=42)
```

```
for i, topic in lda_model_events_males_5.show_topics(num_words=20):
    print(f"TOPIC {i+1}: \n\n", topic, "\n\n-------------------------\n")
```

```
TOPIC 1:

 0.033*"find" + 0.029*"think" + 0.026*"begin" + 0.019*"turn" + 0.017*"hear" + 0.017*"eat" + 0.015*"grow" + 0.015*"seize" + 0.015*"speak" + 0.014*"hold" + 0.011*"sit" + 0.011*"run" + 0.010*"know" + 0.010*"look" + 0.010*"set" + 0.010*"stand" + 0.010*"reach" + 0.009*"continue" + 0.009*"thank" + 0.009*"call"

-------------------------

TOPIC 2:

 0.030*"ask" + 0.028*"set" + 0.020*"live" + 0.018*"find" + 0.017*"become" + 0.015*"look" + 0.014*"stand" + 0.014*"arrive" + 0.013*"beg" + 0.013*"send" + 0.012*"reply" + 0.012*"put" + 0.011*"lose" + 0.010*"see" + 0.010*"meet" + 0.010*"bring" + 0.010*"think" + 0.010*"turn" + 0.010*"bid" + 0.009*"dress"

-------------------------

TOPIC 3:

 0.037*"get" + 0.022*"begin" + 0.020*"ask" + 0.019*"think" + 0.018*"set" + 0.018*"put" + 0.017*"cry" + 0.016*"look" + 0.016*"throw" + 0.015*"fell" + 0.015*"want" + 0.015*"lay" + 0.015*"sit" + 0.014*"run" + 0.013*"carry" + 0.012*"leave" + 0.012*"call" + 0.012*"bring" + 0.012*"kill" + 0.011*"fly"

-------------------------

TOPIC 4:

 0.029*"send" + 0.024*"leave" + 0.022*"return" + 0.021*"bring" + 0.020*"think" + 0.020*"hear" + 0.018*"know" + 0.016*"enter" + 0.015*"order" + 0.015*"lead" + 0.014*"die" + 0.013*"live" + 0.012*"declare" + 0.012*"begin" + 0.012*"carry" + 0.011*"look" + 0.011*"run" + 0.010*"present" + 0.009*"turn" + 0.009*"inform"

-------------------------

TOPIC 5:

 0.051*"answer" + 0.028*"ask" + 0.027*"reply" + 0.025*"hear" + 0.024*"cry" + 0.019*"send" + 0.015*"call" + 0.014*"bring" + 0.014*"look" + 0.013*"think" + 0.013*"find" + 0.012*"know" + 0.012*"sit" + 0.011*"see" + 0.011*"get" + 0.011*"ride" + 0.011*"stand" + 0.011*"run" + 0.010*"order" + 0.010*"put"

-------------------------
```

```
vis_data = gensimvis.prepare(lda_model_events_males_5, corpus, dictionary)
pyLDAvis.save_html(vis_data, 'lda_visualization_events_males_5.html')
pyLDAvis.display(vis_data)
```

```
/opt/anaconda3/envs/textmining/lib/python3.8/site-packages/pyLDAvis/_prepare.py:243: FutureWarning: In a future version of pandas all arguments of DataFrame.drop except for the argument 'labels' will be keyword-only.
  default_term_info = default_term_info.sort_values(
```



The 5 topics for males using events are shown as above. Keeping $\lambda = 0.6$ this helps us in selecting words that are relevant to that topic to give more importance to ratio of frequency of word given topic to the overall frequency of word.

- We observe that topic 1, topic 2, topic 3, topic 4, topic 5 correspond to 29.6%, 22%, 17.3%, 16.9%, and 14.2% of the tokens in the corpus respectively.
- Topic 1 is separated from topics 3, 4 and 5 but overlaps very slightly with topic 2.
- Topic 1 includes top terms like ‘answer’, ‘reply’, ‘cry’, ‘ask’, ‘hear’ etc suggesting that the overall theme is related to answering and asking questions.
- Topic 2 includes top terms like ‘get’, ‘want’, ‘begin’, ‘throw’ etc suggesting that the theme is somewhat related to achieving something or starting something.
- Topic 3 includes top terms like ‘find’, ‘begin’, ‘think’, ‘seize’ suggesting that the theme is somewhat related to getting something and starting. Similar to topic 2.
- Topic 4 includes top terms like ‘set’, ‘ask’, ‘live’, ‘become’ etc suggesting that the theme is related to living in some manner or fashion.
- Topic 5 includes top terms like ‘send’, ‘return’, ‘leave’, ‘enter’ etc suggesting that the theme is somewhat related to moving from one location to another.
- The overall topics are distinct and here we are able to identify a specific distinct theme that we can identify each topic with. There are a few topics that can be distinctly identifies with a key theme. Like topic 2, 3 is related to achieving something and topic 5 is related to moving from one location to another.

3 Topics for males using events

LDA Topic Modeling for Males using events and 3 topics

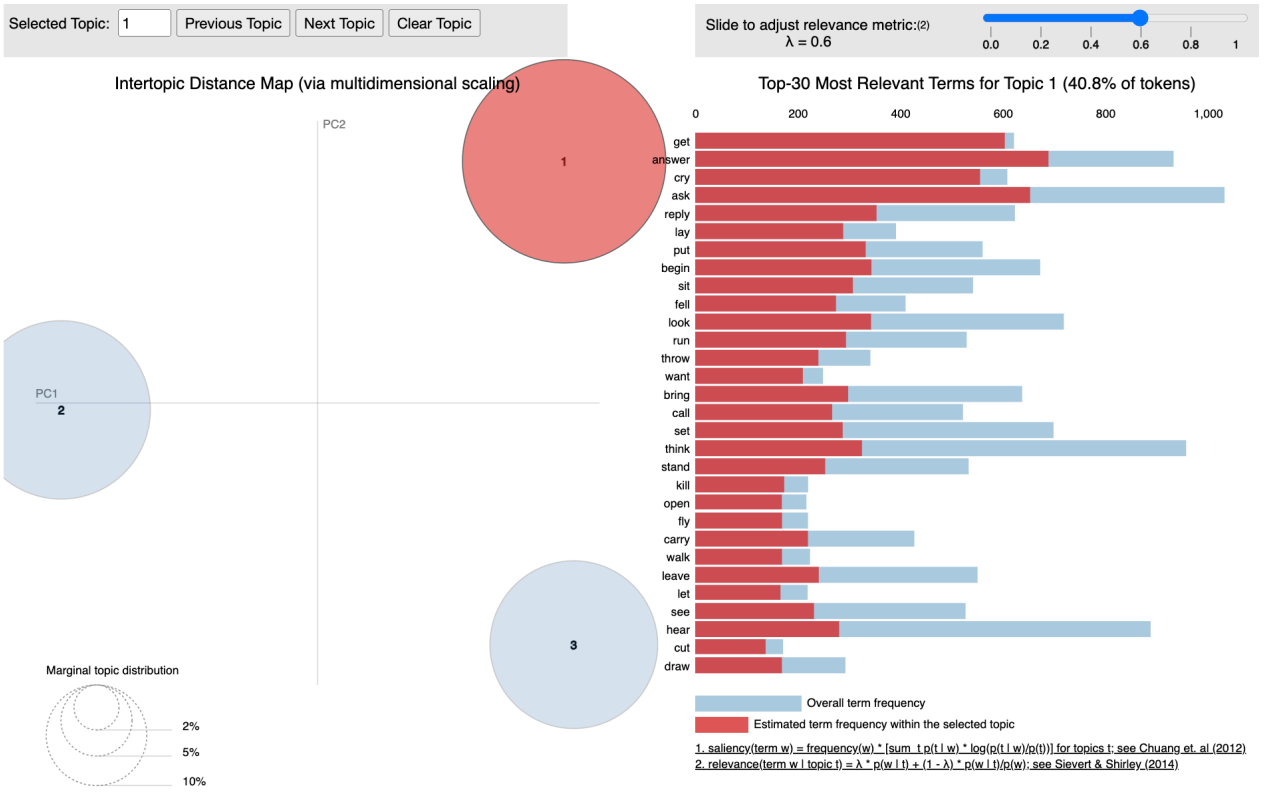
```
lda_model_events_males_3 = models.LdaModel(corpus, num_topics=3, id2word=dictionary, passes=15, random_state=42)
```

```
for i, topic in lda_model_events_males_3.show_topics(num_words=20):
    print(f"TOPIC {i+1}: \n\n", topic, "\n\n--------------------------\n")
```

```
TOPIC 1:

 0.033*"find" + 0.032*"think" + 0.027*"hear" + 0.018*"turn" + 0.016*"begin" + 0.016*"know" + 0.015*"grow" + 0.015*"lead" + 0.012*"look" + 0.011*"set" + 0.011
*"sit" + 0.011*"answer" + 0.011*"hold" + 0.011*"eat" + 0.011*"run" + 0.010*"seize" + 0.010*"stand" + 0.010*"call" + 0.009*"return" + 0.009*"reach"

--------------------------

TOPIC 2:

 0.025*"send" + 0.016*"bring" + 0.016*"ask" + 0.016*"live" + 0.014*"set" + 0.014*"leave" + 0.012*"find" + 0.012*"beg" + 0.012*"hear" + 0.012*"reply" + 0.012
*"become" + 0.011*"look" + 0.011*"order" + 0.011*"arrive" + 0.011*"return" + 0.009*"see" + 0.009*"follow" + 0.009*"think" + 0.008*"pass" + 0.008*"receive"

--------------------------

TOPIC 3:

 0.031*"answer" + 0.030*"ask" + 0.028*"get" + 0.025*"cry" + 0.016*"reply" + 0.016*"begin" + 0.016*"look" + 0.015*"put" + 0.015*"think" + 0.014*"sit" + 0.014
*"bring" + 0.013*"run" + 0.013*"lay" + 0.013*"set" + 0.013*"hear" + 0.013*"fell" + 0.012*"call" + 0.012*"stand" + 0.011*"leave" + 0.011*"throw"
```

```
vis_data = gensimvis.prepare(lda_model_events_males_3, corpus, dictionary)
pyLDAvis.save_html(vis_data, 'lda_visualization_events_male_3.html')
pyLDAvis.display(vis_data)
```

```
/opt/anaconda3/envs/textmining/lib/python3.8/site-packages/pyLDAvis/_prepare.py:243: FutureWarning: In a future version of pandas all arguments
drop except for the argument 'labels' will be keyword-only.
  default_term_info = default_term_info.sort_values(
```



The 3 topics for males using events are shown as above. Keeping $\lambda = 0.6$ this helps us in selecting words that are relevant to that topic to give more importance to ratio of frequency of word given topic to the overall frequency of word.

- We observe that topic 1, topic 2, topic 3 correspond to 40.8%, 31.4%, 27.8% of the tokens in the corpus respectively.
- All the topics are clearly separated from one another with large inter topic distances.
- Topic 1 includes top terms like ‘get’, ‘answer’, ‘cry’, ‘ask’, ‘reply’ etc suggesting that the overall theme is related to answering and asking questions.
- Topic 2 includes top terms like ‘send’, ‘live’, ‘arrive’, ‘begin’ etc suggesting that the theme is somewhat related to starting something and finishing it.
- Topic 3 includes top terms like ‘find’, ‘think’, ‘hear’, ‘lead’ etc suggesting that the theme is related to human thinking and spearheading others.
- The overall topics are distinct and here we are able to identify some topic themes clearly. There are a few topics that can be distinctly identifies with a key theme. Like topic 1 is related to answering and asking questions.

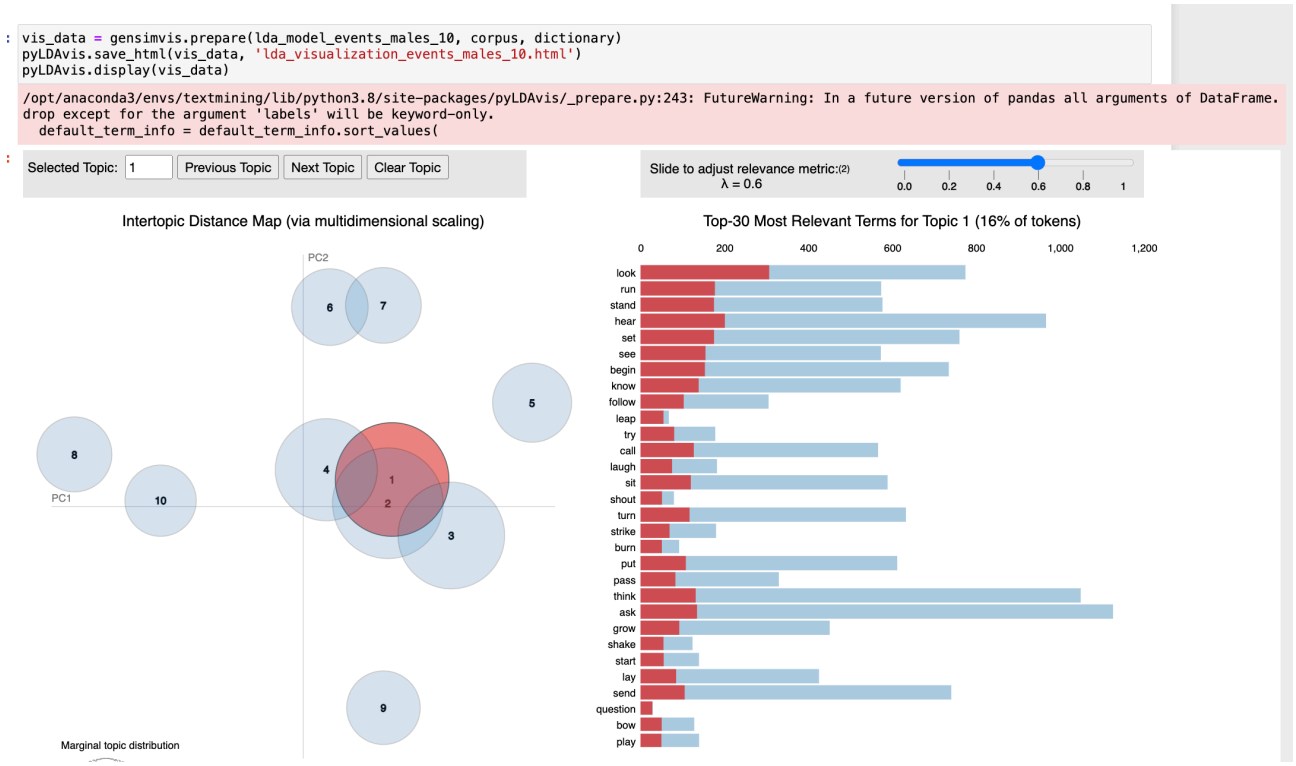
10 Topics for males using events

LDA Topic Modeling for Males using events and 10 topics

```
lda_model_events_males_10 = models.LdaModel(corpus, num_topics=10, id2word=dictionary, passes=15, random_state=42)
```

```
for i, topic in lda_model_events_males_10.show_topics(num_words=20):
    print(f"TOPIC {i+1}: \n\n", topic, "\n\n--------------------------\n")
```

```
TOPIC 1:

 0.065*"find" + 0.055*"think" + 0.043*"eat" + 0.030*"seize" + 0.023*"lead" + 0.018*"carry" + 0.017*"begin" + 0.016*"put" + 0.014*"continue" + 0.012*"turn" +
0.012*"return" + 0.012*"thank" + 0.011*"set" + 0.010*"ride" + 0.010*"appear" + 0.010*"reach" + 0.010*"bid" + 0.010*"hold" + 0.009*"run" + 0.008*"know"

--------------------------

TOPIC 2:

 0.031*"set" + 0.025*"find" + 0.020*"become" + 0.018*"live" + 0.017*"stand" + 0.014*"lose" + 0.013*"see" + 0.013*"grow" + 0.013*"speak" + 0.013*"fell" + 0.01
2*"address" + 0.012*"add" + 0.012*"talk" + 0.011*"put" + 0.010*"remain" + 0.010*"happen" + 0.009*"leave" + 0.009*"desire" + 0.009*"beg" + 0.008*"keep"

--------------------------

TOPIC 3:

 0.032*"get" + 0.028*"ask" + 0.025*"put" + 0.024*"sit" + 0.022*"fell" + 0.021*"begin" + 0.020*"set" + 0.019*"cry" + 0.018*"call" + 0.018*"want" + 0.017*"thro
w" + 0.016*"think" + 0.016*"fly" + 0.014*"carry" + 0.014*"kill" + 0.013*"leave" + 0.013*"lay" + 0.012*"run" + 0.012*"catch" + 0.010*"pull"

--------------------------

TOPIC 4:

 0.036*"leave" + 0.035*"return" + 0.032*"live" + 0.031*"grow" + 0.030*"order" + 0.027*"die" + 0.021*"run" + 0.019*"think" + 0.017*"hear" + 0.017*"enter" + 0.
015*"send" + 0.015*"determine" + 0.013*"bow" + 0.013*"lead" + 0.012*"write" + 0.012*"reach" + 0.012*"bring" + 0.012*"call" + 0.012*"happen" + 0.011*"show"

--------------------------

TOPIC 5:

 0.094*"answer" + 0.050*"reply" + 0.039*"cry" + 0.025*"ask" + 0.018*"find" + 0.018*"send" + 0.018*"hear" + 0.014*"look" + 0.014*"ride" + 0.013*"bring" + 0.01
3*"speak" + 0.012*"call" + 0.012*"lay" + 0.011*"stand" + 0.011*"leave" + 0.011*"meet" + 0.011*"get" + 0.010*"bid" + 0.010*"run" + 0.010*"grow"

--------------------------

TOPIC 6:

 0.033*"look" + 0.021*"hear" + 0.019*"run" + 0.019*"set" + 0.019*"stand" + 0.017*"see" + 0.016*"begin" + 0.015*"know" + 0.014*"ask" + 0.014*"think" + 0.014
*"call" + 0.013*"sit" + 0.012*"turn" + 0.012*"put" + 0.011*"send" + 0.011*"follow" + 0.010*"grow" + 0.009*"lay" + 0.009*"pass" + 0.009*"find"

--------------------------

TOPIC 7:

 0.032*"stand" + 0.031*"hold" + 0.026*"turn" + 0.024*"speak" + 0.019*"draw" + 0.016*"know" + 0.013*"put" + 0.013*"look" + 0.013*"sit" + 0.013*"let" + 0.013
*"touch" + 0.013*"sing" + 0.012*"enter" + 0.012*"listen" + 0.012*"return" + 0.011*"cut" + 0.011*"walk" + 0.011*"become" + 0.011*"appear" + 0.010*"bring"

--------------------------

TOPIC 8:

 0.067*"send" + 0.029*"arrive" + 0.023*"hear" + 0.020*"beg" + 0.019*"carry" + 0.015*"begin" + 0.015*"promise" + 0.014*"use" + 0.013*"receive" + 0.013*"find"
+ 0.013*"break" + 0.013*"order" + 0.012*"return" + 0.012*"felt" + 0.011*"see" + 0.011*"agree" + 0.010*"become" + 0.010*"try" + 0.009*"place" + 0.009*"start"

--------------------------

TOPIC 9:

 0.053*"ask" + 0.036*"hear" + 0.031*"think" + 0.019*"leave" + 0.018*"begin" + 0.016*"know" + 0.016*"turn" + 0.015*"set" + 0.015*"see" + 0.015*"love" + 0.014
*"bring" + 0.014*"meet" + 0.013*"find" + 0.012*"look" + 0.011*"reach" + 0.011*"lead" + 0.010*"declare" + 0.010*"return" + 0.009*"answer" + 0.009*"follow"

--------------------------

TOPIC 10:

 0.057*"bring" + 0.049*"get" + 0.026*"think" + 0.025*"ask" + 0.020*"eat" + 0.019*"order" + 0.018*"send" + 0.017*"know" + 0.016*"look" + 0.015*"open" + 0.015
*"throw" + 0.014*"shut" + 0.013*"live" + 0.013*"hear" + 0.011*"summon" + 0.010*"drive" + 0.010*"want" + 0.010*"fill" + 0.010*"keep" + 0.010*"drink"
```



The 10 topics for males using events are shown as above. Keeping $\lambda = 0.6$ this helps us in selecting words that are relevant to that topic to give more importance to ratio of frequency of word given topic to the overall frequency of word.

- We observe that topic 1, topic 2, topic 3, topic 4, topic 5, topic 6, topic 7, topic 8, topic 9, topic 10 correspond to 16%, 12.3%, 11.6%, 11.5%, 11%, 10.6%, 8.1%, 7.9%, 5.4%, 4.8% of the tokens in the corpus respectively.
- Topics 5, 8, 9, 10 have clear distinction from other topics and have no overlaps.
- Topics 1, 2, 3, 4 have overlapping and topics 6, 7 overlap.
- Topic 1 includes top terms like ‘look’, ‘run’, ‘stand’, ‘hear’, ‘see’ etc suggesting that the theme is related to seeing and performing motion.
- Topic 2 includes top terms like ‘answer’, ‘reply’, ‘cry’, ‘ask’ etc suggesting that the theme is somewhat related to answering and asking questions.
- Topic 3 includes top terms like ‘get’, ‘put’, ‘fell’, ‘sit’ etc suggesting that the theme is somewhat related to human motion.
- Topic 4 includes top terms like ‘ask’, ‘hear’, ‘think’, ‘love’ etc suggesting that the theme is somewhat related to human emotions.
- Topic 5 includes top terms like ‘hold’, ‘stand’, ‘speak’, ‘touch’ etc suggesting that the theme is somewhat related to human locomotion and senses.
- Topic 6 includes top terms like ‘address’, ‘set’, ‘become’ etc suggesting that the theme is somewhat related to converting from one to another.
- Topic 7 includes top terms like ‘find’, ‘eat’, ‘seize’, ‘think’ etc suggests to achieve something.
- Topic 8 includes top terms like ‘send’, ‘arrive’, ‘use’, ‘beg’ etc suggesting item sending and receiving.
- Topic 9 includes top terms like ‘bring’, ‘get’, ‘shut’, ‘summon’ giving an idea calling or getting called by someone.

- Topic 10 includes top terms like ‘die’, ‘return’, ‘order’, ‘leave’ etc suggesting departure of item or person.
- The overall topics are distinct and here we are able to identify some topic themes clearly. However there is some overlap in the topics and the ideas are shared between them due to presence of multiple words in other topics.

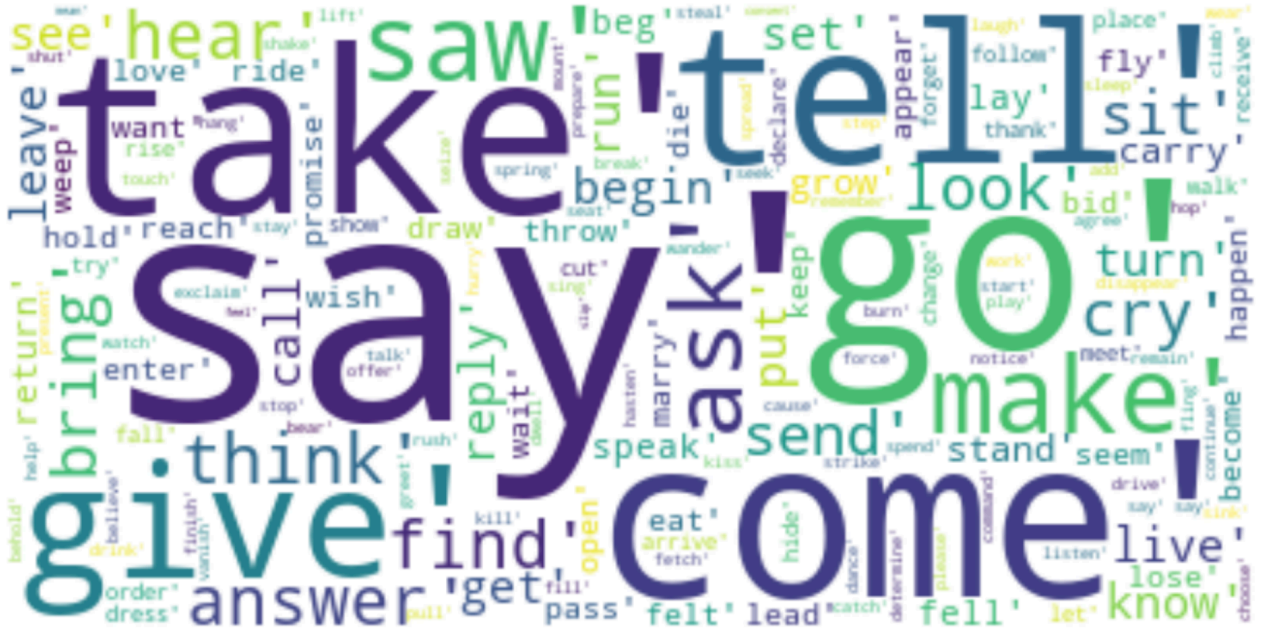
Based on this, we can understand that the number of topics need to be optimal for identifying distinct topics in LDA topic modelling. Too few topics may lead to underfitting and too many topics can lead to overfitting. Underfit models tend to group unrelated words together and overfit models can produce topics with many irrelevant words.

- In our case based on my assumption, the LDA model for males with events with 3 topics was better than 5 or 10 topics. The reason being, I was able to atleast identify the themes associated with the topics distinctly.
- The model with 5 topics was also good in terms of segregating the different themes with respect to the topics. There are 2 topics with similar interpretations. We can say that it may not have suffered from overfitting as we were able to identify themes and it did not produce irrelevant words.
- The model with 10 topics was the worst as it suffered from overfitting with ideas being repeated in multiple other topics. There were words in topics but they could not help in identifying a clear idea relevant to the topic.

Female Events

```
events_female.head(1)
```

	fairy_tale	character	character_sentences	events	gender	sentences	preprocessed_sentences
0	adventures_of_an_indian_brave	women	he walked through the maize without knowing where he was going, when he heard a sound of knocking, and saw two old blind women crushing their food between two stones. and again the young man stretched out his hand; and the two old women fell to quarrelling afresh. but when it happened for the third time the old women suspected some trick, and one of them exclaimed:' i am sure there is a man here; tell me, are you not my grandson?' then he hastened back to the old women, and begging them to boil him some water, he threw the herb in. as soon as the pot began to sing he took off the lid, and sprinkled the eyes of the women, and sight came back to them once more. he had beheld no one except the old women, and it was not very likely that they would be able to help him. and they wove him the net he asked for, and for many weeks he watched by the river, only going back to the old women when he wanted a fish cooked. in an instant the basket moved, but, to his horror, instead of going down, he felt himself being drawn upwards, and shortly after he beheld the faces of the old women.	walk crush saw hear fell stretch exclaim happen suspect throw hasten beg boil take sprinkle come begin watch weave ask go want behold draw felt move	female	he walked through the maize without knowing where he was going, when he heard a sound of knocking, and saw two old blind women crushing their food between two stones. and again the young man stretched out his hand; and the two old women fell to quarrelling afresh. but when it happened for the third time the old women suspected some trick, and one of them exclaimed: 'i am sure there is a man here; tell me, are you not my grandson?' then he hastened back to the old women, and begging them to boil him some water, he threw the herb in. as soon as the pot began to sing he took off the lid, and sprinkled the eyes of the women, and sight came back to them once more. he had beheld no one except the old women, and it was not very likely that they would be able to help him. and they wove him the net he asked for, and for many weeks he watched by the river, only going back to the old women when he wanted a fish cooked. in an instant the basket moved, but, to his horror, instead of going down, he felt himself being drawn upwards, and shortly after he beheld the faces of the old women.	[walk, crush, saw, hear, fell, stretch, exclaim, happen, suspect, throw, hasten, beg, boil, take, sprinkle, come, begin, watch, weave, ask, go, want, behold, draw, felt, move]



From the word cloud we observe that for males, we can see that the top and most frequent words include ‘come’, ‘take’, ‘go’, ‘say’, ‘tell’, ‘make’, ‘give’. This suggests that the key theme of this is on the activity of achieving something or adventurous.

Preparing the data for LDA Topic Modeling and an example

```
dictionary = corpora.Dictionary(events_female['preprocessed_sentences'])
dictionary.filter_extremes(no_below=5, no_above=0.2, keep_n=10000)

corpus = [dictionary.doc2bow(sentence) for sentence in events_female['preprocessed_sentences']]
```

Restricting top 10K terms that appear in 5 documents and not more that 20% documents.

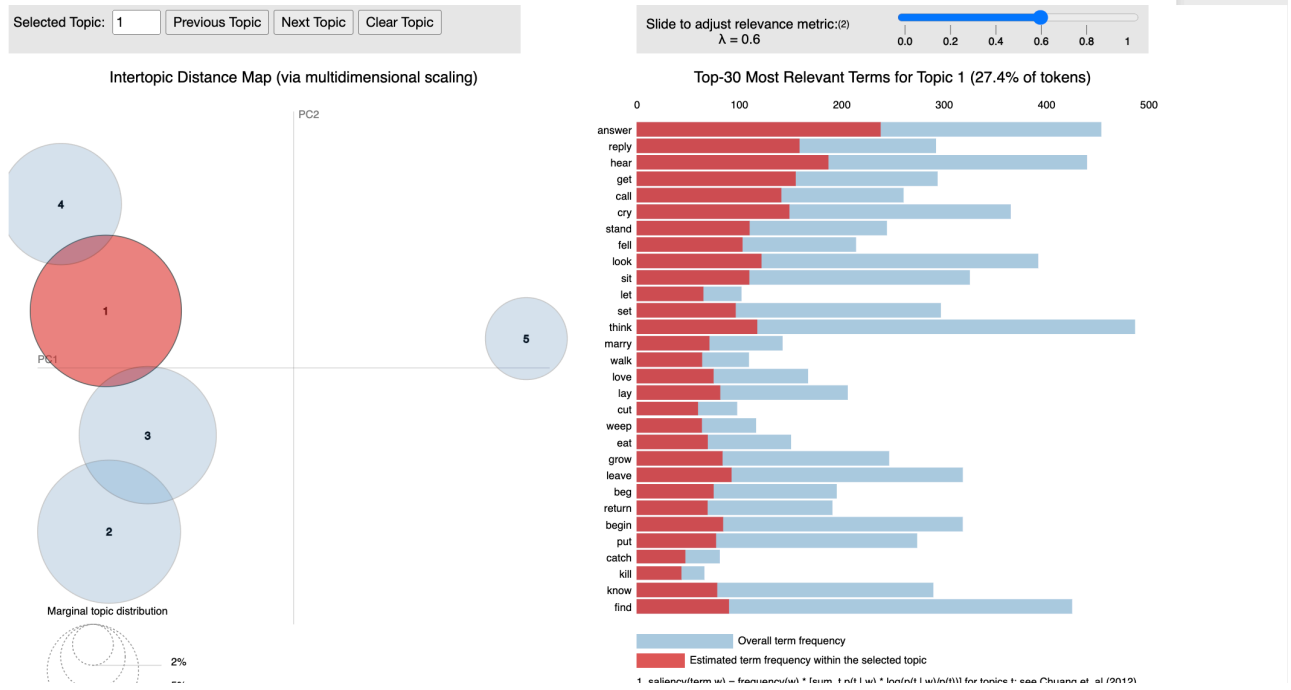
5 Topics for females using events

```
for i, topic in lda_model_events_females_5.show_topics(num_words=20):
    print(f"TOPIC {i+1}: \n\n", topic, "\n\n-------------------------\n")
```

```
TOPIC 1:

 0.029*"send" + 0.020*"think" + 0.017*"answer" + 0.017*"turn" + 0.016*"set" + 0.016*"carry" + 0.015*"look" + 0.014*"sit" + 0.014*"live" + 0.013*"bring" + 0.0
13*"find" + 0.012*"bid" + 0.011*"know" + 0.011*"leave" + 0.011*"felt" + 0.010*"eat" + 0.010*"return" + 0.009*"beg" + 0.009*"pass" + 0.009*"get"

-------------------------

TOPIC 2:

 0.024*"find" + 0.023*"think" + 0.019*"see" + 0.016*"hear" + 0.015*"cry" + 0.014*"know" + 0.014*"put" + 0.012*"run" + 0.012*"enter" + 0.012*"fly" + 0.011*"ap
pear" + 0.011*"call" + 0.010*"hold" + 0.010*"grow" + 0.010*"become" + 0.009*"lay" + 0.009*"set" + 0.009*"promise" + 0.009*"speak" + 0.008*"begin"

-------------------------

TOPIC 3:

 0.024*"begin" + 0.022*"turn" + 0.021*"bring" + 0.019*"leave" + 0.017*"look" + 0.017*"run" + 0.015*"find" + 0.014*"hear" + 0.014*"carry" + 0.014*"cry" + 0.01
2*"think" + 0.012*"ride" + 0.011*"throw" + 0.011*"answer" + 0.011*"put" + 0.010*"grow" + 0.009*"love" + 0.009*"reply" + 0.009*"get" + 0.009*"appear"

-------------------------

TOPIC 4:

 0.032*"answer" + 0.025*"hear" + 0.021*"reply" + 0.021*"get" + 0.020*"cry" + 0.019*"call" + 0.016*"look" + 0.016*"think" + 0.015*"stand" + 0.015*"sit" + 0.01
4*"fell" + 0.013*"set" + 0.012*"leave" + 0.012*"find" + 0.011*"bring" + 0.011*"begin" + 0.011*"grow" + 0.011*"lay" + 0.010*"know" + 0.010*"put"

-------------------------

TOPIC 5:

 0.036*"live" + 0.029*"bring" + 0.021*"look" + 0.020*"sit" + 0.015*"pass" + 0.015*"see" + 0.015*"lose" + 0.014*"think" + 0.013*"begin" + 0.012*"find" + 0.011
*"happen" + 0.011*"want" + 0.011*"reach" + 0.011*"call" + 0.011*"change" + 0.010*"steal" + 0.010*"carry" + 0.010*"return" + 0.010*"lead" + 0.010*"hear"

-------------------------
```

```
vis_data = gensimvis.prepare(lda_model_events_females_5, corpus, dictionary)
pyLDAvis.save_html(vis_data, 'lda_visualization_events_females_5.html')
pyLDAvis.display(vis_data)
```

```
/opt/anaconda3/envs/textmining/lib/python3.8/site-packages/pyLDAvis/_prepare.py:243: FutureWarning: In a future version of pandas all arguments of DataFrame.
drop except for the argument 'labels' will be keyword-only.
  default_term_info = default_term_info.sort_values(
```



The 5 topics for females using events are shown as above. Keeping $\lambda = 0.6$ this helps us in selecting words that are relevant to that topic to give more importance to ratio of frequency of word given topic to the overall frequency of word.

- We observe that topic 1, topic 2, topic 3, topic 4, topic 5 correspond to 27.4%, 24.5%, 22.5%, 17.6%, and 8% of the tokens in the corpus respectively.
- Topic 5 is separated from others.
- Topics 1, 2, 3, 4 and 5 show some overlapping amongst them.
- Topic 1 includes top terms like ‘answer’, ‘reply’, ‘hear’, ‘get’ etc suggesting that the overall theme is related to answering replying to the questions.
- Topic 2 includes top terms like ‘find’, ‘think’, ‘see’, ‘fly’ etc suggesting that the theme is somewhat related to human activities.
- Topic 3 includes top terms like ‘send’, ‘turn’, ‘carry’, ‘think’ etc suggesting that the theme is somewhat related to human activities.
- Topic 4 includes top terms like ‘begin’, ‘turn’, ‘leave’, ‘bring’, ‘mermaid’ etc suggesting that the theme is related to starting and completing tasks.
- Topic 5 includes top terms like ‘live’, ‘bring’, ‘address’, ‘lose’ etc suggesting that the theme is somewhat related to the human activities.
- The overall topics are distinct but we are not able to associate specific distinct categories for the topics. There are a few topics that can be distinctly identifies with a key theme. Like only topic 1 is clearly identifiable with answering replying to the questions.

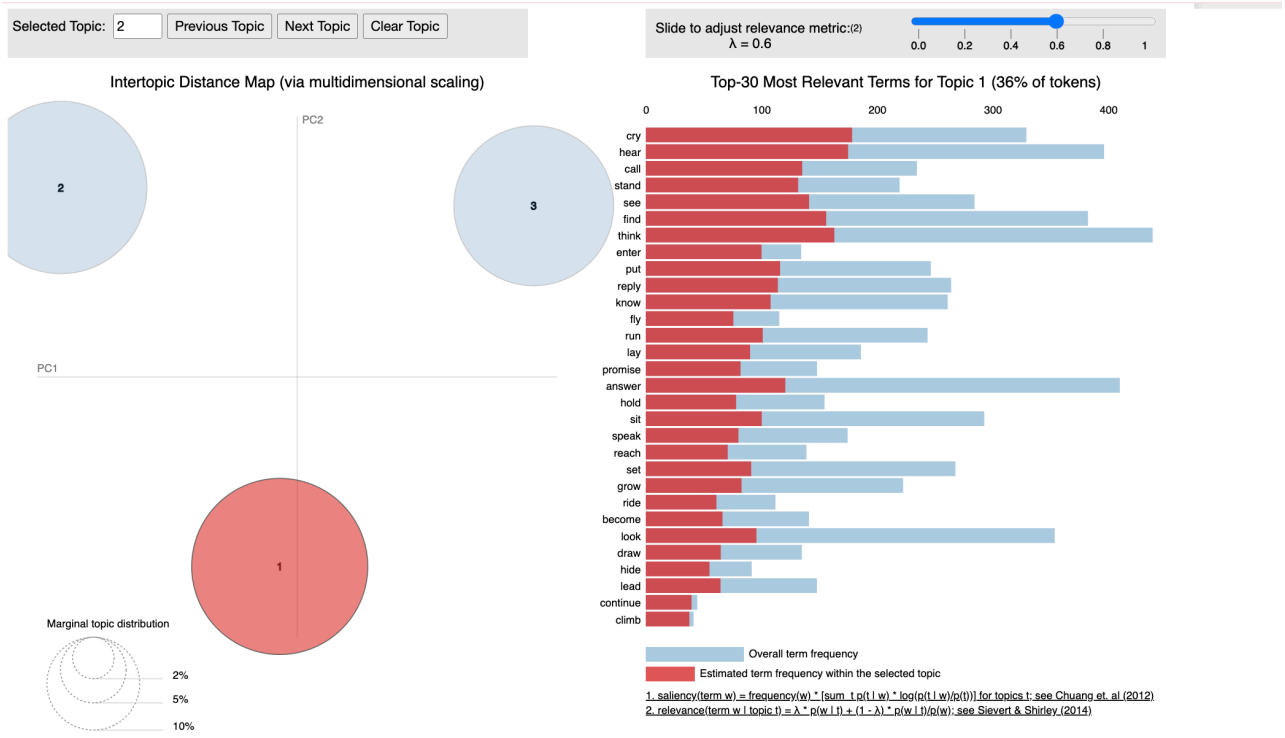
3 Topics for females using events

LDA Topic Modeling for Females using events and 3 topics

```
lda_model_events_females_3 = models.LdaModel(corpus, num_topics=3, id2word=dictionary, passes=15, random_state=42)
```

```
for i, topic in lda_model_events_females_3.show_topics(num_words=20):
    print(f"TOPIC {i+1}: \n\n", topic, "\n\n--------------------------\n")
```

```
TOPIC 1:

 0.028*"send" + 0.020*"live" + 0.018*"answer" + 0.017*"think" + 0.015*"get" + 0.014*"sit" + 0.014*"set" + 0.014*"find" + 0.014*"look" + 0.013*"turn" + 0.013
*"bring" + 0.012*"carry" + 0.011*"leave" + 0.011*"pass" + 0.011*"hear" + 0.010*"return" + 0.010*"felt" + 0.009*"know" + 0.009*"beg" + 0.009*"happen"

--------------------------

TOPIC 2:

 0.020*"cry" + 0.020*"hear" + 0.018*"think" + 0.018*"find" + 0.016*"see" + 0.015*"call" + 0.015*"stand" + 0.014*"answer" + 0.013*"put" + 0.013*"reply" + 0.01
2*"know" + 0.011*"run" + 0.011*"sit" + 0.011*"enter" + 0.011*"look" + 0.010*"set" + 0.010*"lay" + 0.009*"grow" + 0.009*"promise" + 0.009*"speak"

--------------------------

TOPIC 3:

 0.025*"begin" + 0.022*"bring" + 0.019*"look" + 0.019*"answer" + 0.018*"hear" + 0.017*"think" + 0.016*"leave" + 0.016*"run" + 0.016*"turn" + 0.015*"fell" +
0.015*"find" + 0.014*"reply" + 0.014*"cry" + 0.013*"want" + 0.012*"grow" + 0.012*"get" + 0.011*"see" + 0.011*"love" + 0.010*"carry" + 0.010*"know"

--------------------------
```



The 3 topics for females using events are shown as above. Keeping $\lambda = 0.6$ this helps us in selecting words that are relevant to that topic to give more importance to ratio of frequency of word given topic to the overall frequency of word.

- We observe that topic 1, topic 2, topic 3 correspond to 36%, 34.4%, 29.6% of the tokens in the corpus respectively.
- All the topics are clearly separated from one another with large inter topic distances.
- Topic 1 includes top terms like ‘cry’, ‘hear’, ‘call’, ‘stand’ suggesting that the theme is somewhat related to human submissive activities of crying and standing.
- Topic 2 includes top terms like ‘send’, ‘live’, ‘answer’, ‘get’ etc suggesting that the theme is somewhat related to answering and getting.
- Topic 3 includes top terms like ‘begin’, ‘bring’, ‘want’, ‘look’ etc suggesting that the theme is related to starting activities.
- The overall topics are distinct and here we are able to identify some topic themes clearly. There are a few topics that can be distinctly identifies with a key theme. Like topic 1 is related human submissive activities of crying and standing.

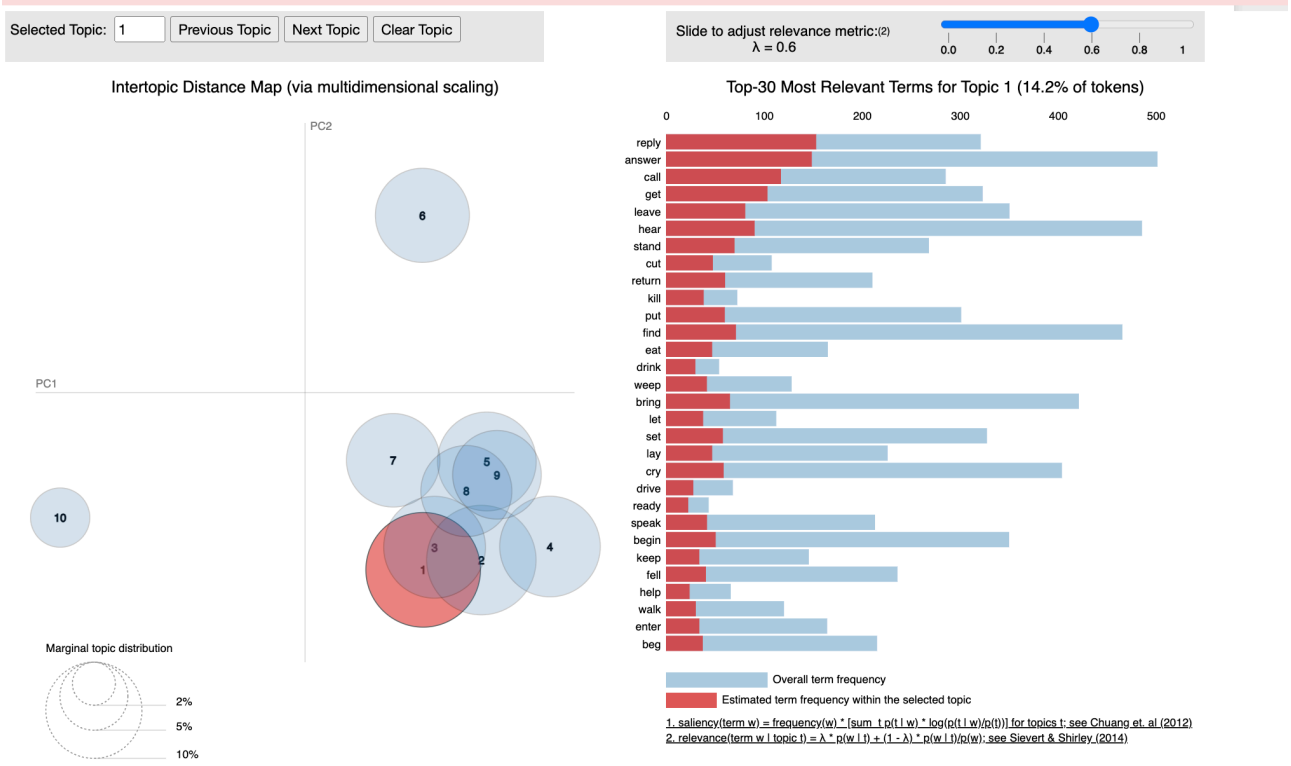
10 Topics for females using events

LDA Topic Modeling for Females using events and 10 topics

```
lda_model_events_females_10 = models.LdaModel(corpus, num_topics=10, id2word=dictionary, passes=15, random_state=42)
```

```
for i, topic in lda_model_events_females_10.show_topics(num_words=20):
    print(f"TOPIC {i+1}: \n\n", topic, "\n\n-------------------------\n")
```

```
TOPIC 1:

 0.031*"get" + 0.025*"send" + 0.020*"answer" + 0.018*"set" + 0.017*"carry" + 0.016*"turn" + 0.014*"bid" + 0.014*"know" + 0.013*"lay" + 0.013*"meet" + 0.012
*"follow" + 0.012*"stand" + 0.012*"wander" + 0.012*"begin" + 0.012*"live" + 0.012*"pass" + 0.011*"bring" + 0.011*"try" + 0.011*"put" + 0.010*"hear"

-------------------------

TOPIC 2:

 0.025*"find" + 0.025*"think" + 0.023*"know" + 0.021*"put" + 0.019*"run" + 0.019*"lay" + 0.018*"hear" + 0.017*"speak" + 0.014*"set" + 0.013*"call" + 0.013*"b
ecome" + 0.012*"show" + 0.010*"enter" + 0.010*"see" + 0.010*"cry" + 0.010*"look" + 0.010*"grow" + 0.009*"declare" + 0.009*"begin" + 0.008*"leave"

-------------------------

TOPIC 3:

 0.032*"leave" + 0.027*"begin" + 0.024*"bring" + 0.017*"turn" + 0.017*"run" + 0.017*"die" + 0.014*"grow" + 0.014*"put" + 0.014*"bid" + 0.014*"throw" + 0.013
*"think" + 0.013*"get" + 0.012*"fell" + 0.012*"follow" + 0.012*"see" + 0.011*"sit" + 0.011*"set" + 0.010*"live" + 0.010*"fill" + 0.010*"seek"

-------------------------

TOPIC 4:

 0.036*"reply" + 0.035*"answer" + 0.027*"call" + 0.024*"get" + 0.021*"hear" + 0.019*"leave" + 0.017*"find" + 0.016*"stand" + 0.015*"bring" + 0.014*"return" +
0.014*"put" + 0.014*"cry" + 0.014*"set" + 0.012*"begin" + 0.011*"cut" + 0.011*"lay" + 0.011*"eat" + 0.010*"speak" + 0.010*"weep" + 0.009*"fell"

-------------------------

TOPIC 5:

 0.062*"live" + 0.057*"bring" + 0.033*"steal" + 0.030*"send" + 0.025*"call" + 0.018*"answer" + 0.017*"lose" + 0.016*"shut" + 0.015*"sit" + 0.015*"spin" + 0.0
14*"drive" + 0.013*"reply" + 0.013*"reach" + 0.012*"begin" + 0.012*"consult" + 0.011*"stand" + 0.011*"cause" + 0.011*"hear" + 0.011*"cry" + 0.011*"get"

-------------------------

TOPIC 6:

 0.026*"find" + 0.025*"lead" + 0.020*"promise" + 0.020*"fly" + 0.020*"ride" + 0.018*"set" + 0.016*"think" + 0.016*"mount" + 0.015*"leave" + 0.015*"sit" + 0.0
15*"live" + 0.013*"hear" + 0.013*"carry" + 0.012*"place" + 0.012*"return" + 0.012*"reach" + 0.012*"draw" + 0.011*"stand" + 0.011*"reply" + 0.010*"open"

-------------------------

TOPIC 7:

 0.046*"look" + 0.035*"think" + 0.026*"find" + 0.020*"eat" + 0.018*"see" + 0.018*"grow" + 0.014*"sit" + 0.013*"bring" + 0.013*"stand" + 0.012*"begin" + 0.011
*"beg" + 0.011*"want" + 0.011*"live" + 0.010*"rise" + 0.010*"turn" + 0.010*"leave" + 0.010*"arrive" + 0.009*"seem" + 0.009*"call" + 0.008*"seize"

-------------------------

TOPIC 8:

 0.054*"send" + 0.027*"see" + 0.023*"cry" + 0.017*"appear" + 0.016*"fell" + 0.014*"think" + 0.014*"happen" + 0.013*"hear" + 0.013*"answer" + 0.012*"set" + 0.
012*"put" + 0.012*"fetch" + 0.011*"bring" + 0.010*"sit" + 0.010*"beg" + 0.009*"receive" + 0.009*"arrive" + 0.009*"carry" + 0.008*"throw" + 0.008*"grow"

-------------------------

TOPIC 9:

 0.040*"cry" + 0.039*"look" + 0.039*"answer" + 0.037*"hear" + 0.019*"turn" + 0.019*"run" + 0.018*"love" + 0.017*"begin" + 0.017*"know" + 0.013*"carry" + 0.01
3*"sit" + 0.012*"kiss" + 0.012*"hide" + 0.011*"beg" + 0.011*"catch" + 0.011*"open" + 0.010*"fell" + 0.010*"bring" + 0.010*"touch" + 0.010*"return"

-------------------------

TOPIC 10:

 0.047*"think" + 0.024*"turn" + 0.022*"hear" + 0.021*"sit" + 0.017*"find" + 0.014*"wish" + 0.014*"bring" + 0.013*"send" + 0.012*"carry" + 0.012*"promise" +
0.012*"look" + 0.011*"set" + 0.011*"dress" + 0.011*"answer" + 0.011*"die" + 0.011*"see" + 0.010*"hold" + 0.010*"put" + 0.009*"get" + 0.009*"leave"

-------------------------
```



The 10 topics for males using events are shown as above. Keeping $\lambda = 0.6$ this helps us in selecting words that are relevant to that topic to give more importance to ratio of frequency of word given topic to the overall frequency of word.

- We observe that topic 1, topic 2, topic 3, topic 4, topic 5, topic 6, topic 7, topic 8, topic 9, topic 10 correspond to 14.2%, 12.9%, 11.1%, 10.9%, 10.5%, 9.6%, 9.4%, 9%, 8.5%, 3.8% of the tokens in the corpus respectively.
- Topics 2 and 6 have clear distinction from other topics and have no overlaps.
- Other topics overlap with each other and are within one another a lot.
- Topic 1 includes top terms like ‘reply’, ‘answer’, ‘call’, ‘get’ etc suggesting that the theme is related to answering questions and summoning.
- Topic 2 includes top terms like ‘look’, ‘think’, ‘eat’, ‘find’ etc suggesting that the theme is related to human activities.
- Topic 3 includes top terms like ‘lead’, ‘ride’, ‘fly’ etc suggesting that the theme is somewhat related to spearheading in tasks.
- Topic 4 includes top terms like ‘queen’, ‘children’, ‘land’, ‘bride’ etc does not indicate any key theme.
- Topic 5 includes top terms like ‘cry’, ‘look’, ‘answer’, ‘hear’ etc suggesting that the theme is somewhat related to human emotional activities.
- Topic 6 includes top terms like ‘send’, ‘see’, ‘fetch’, ‘appear’ etc suggesting that the theme is somewhat related to looking and getting.
- Topic 7 includes top terms like ‘get’, ‘send’, ‘wander’ etc suggesting that the theme is somewhat related to being a hippie and roaming.
- Topic 8 includes top terms like ‘think’, ‘turn’, ‘sit’, ’wish’ etc suggesting that the theme is related to human cognitive activites.

- Topic 9 includes top terms like ‘leave’, ‘begin’, ‘die’ etc suggesting the ides of starting or ending things.
- Topic 10 includes top terms like ‘live’, ‘steal’, ‘bring’ etc suggesting no common theme.
- The overall topics are distinct but the model is worst as many topics could not be identified with a central theme due to repetition of a lot of words across the topics.

Based on this, we can understand that the number of topics need to be optimal for identifying distinct topics in LDA topic modelling. Too few topics may lead to underfitting and too many topics can lead to overfitting. Underfit models tend to group unrelated words together and overfit models can produce topics with many irrelevant words.

- In our case based on my assumption, the LDA model for females with events with 3 topics was better than 5 or 10 topics. The reason being, I was able to atleast identify the themes associated with the topics distinctly.
- The model with 5 topics was also good in terms of segregating the different themes with respect to the topics.
- The model with 10 topics was the worst as it suffered from overfitting and there were words in topics that did not make complete sense.

Performing LDA topic modeling on the male and female characters using events involved steps and we got some interesting topics corresponding to the male and female characters.

- We understood that optimal number of topics helps in identifying clear set of themes for the topics. - In our case it was 5 topics LDA model for males that stood out and 3 topics LDA model for females.
- Using them we were able to get separate ideas for the topics and terms.
- Even though we separated the dataset for male and female characters, we obtained words from both genders for both the male and female topics. This is obvious considering he fact that male and female characters are related to each other and then sentences consisted of some relationships between them.
- We obtained a set of terms specifically related to the events for male and female characters.
- For males, I observed that some terms included were ‘ask’, ‘get’, ‘seize’, ‘begin’, ‘lead’ and they suggest some masculine behaviour of being dominant and the fairy tales might have been written in this manner to consider male characters as somewhat authoritative and having an urge to exhibit masculine behaviour to get things they want.
- For females, I observed that some terms included were ‘answer’, ‘reply’, ‘cried’, ‘wish’, ‘think’ and they suggest some feminine behaviour of being submissive and the fairy tales might have been written in this manner to consider female characters as somewhat recluse and having an exhibiting feminine traits to react emotionally in situations.
- Overall, we have a decent quality of topics that is not of very high quality for males and females as the words do not clearly help us in identifying the key ideas. LDA model with loess topics have a broader domain and can successfully separate the ideas.
- The LDA models with more number of topics tend to perform poor due to overfitting because a lot of words get repeated in multiple topics, thus losing the essence of a central idea specific to the topic.

Task 5: Brainstorm other ways to identify gender differences in the narratives of female and male characters

Identifying gender differences in narratives of male and female characters is important. Here are some additional ways to explore these differences:

- **Sentiment Analysis:** Analyze the sentiment of text associated with male and female characters separately. If we can find differences in emotional tones and attitudes attributed to different genders.

- **Dialogue Trends:** Dialogue analysis of male and female characters can help in understanding different types of conversations and emotions. There may be some patterns.

- **Power Dynamics:** We need to examine the power dynamics between male and female characters. Men are generally depicted as dominant while females as submissive or vice versa. This can highlight gender-based inconsistencies in story.

- **Character Descriptions:** We need to study how male and female characters are described within the narrative. They roles or characters they portray can also highlight the gender-based differences.

References:

1) geeksforgeeks.com

2) Canvas Class Slides: https://canvas.illinois.edu/courses/37566/pages/course-schedule

3) https://neptune.ai/blog/pyldavis-topic-modelling-exploration-tool-that-every-nlp-data-scientist-should-know#:~:text=Each%20bubble%20represents%20a%20topic,used%20words%20will%20be%20displayed

4) www.google.com

5) https://we1s.ucsb.edu/research/we1s-tools-and-software/topic-model-observatory/tmo-guide/tmo-guide-pyldavis/