Proyecto Final: Parte 4 - Tópicos multilinguales

- ***Materia:*** Análisis Inteligente de Textos
- ***Maestro:*** Octavio Augusto Sánchez Velázquez
- ***Alumno:*** José Antonio Velázquez Sánchez

In [185]:
```
import pickle
import myutils
import numpy as np
import scipy.spatial
import gensim.matutils
from gensim import corpora, models
```

Objetivos

- El objetivo de este cuaderno es explorar el modelado de tópicos utilizando un corpus multilingüe
- El primer paso es crear el corpus, combinando los subtítulos en las 3 lenguas de cada película
- Después, entrenaré un modelo de LDA para conseguir los tópicos
- Finalmente, utilizaré algún método para ver si LDA es capaz de capturar traducciones individuales de palabras usando las técnicas mencionadas

Cargando datos pasados

In [2]:
```
txts_spa = pickle.load(open("./pickles/txts_spa.pickle", "rb"))
dictionary_spa = corpora.Dictionary.load("./pickles/dictionary_spa.dict")
frecs_spa = pickle.load(open("./pickles/frecs_spa.pickle", "rb"))

txts_eng = pickle.load(open("./pickles/txts_eng.pickle", "rb"))
dictionary_eng = corpora.Dictionary.load("./pickles/dictionary_eng.dict")
frecs_eng = pickle.load(open("./pickles/frecs_eng.pickle", "rb"))

txts_fre = pickle.load(open("./pickles/txts_fre.pickle", "rb"))
dictionary_fre = corpora.Dictionary.load("./pickles/dictionary_fre.dict")
frecs_fre = pickle.load(open("./pickles/frecs_fre.pickle", "rb"))

movieids = pickle.load(open("./pickles/movieids.pickle", "rb"))
titulos = pickle.load(open("./pickles/titulos.pickle", "rb"))
```

In [29]:
```
txts_spa = txts_spa[0] + txts_spa[1] + txts_spa[2]
txts_fre = txts_fre[0] + txts_fre[1] + txts_fre[2]
txts_eng = txts_eng[0] + txts_eng[1] + txts_eng[2]
```

Creando corpus combinando lenguas

Para crear el corpus combinando de las tres lenguas lo más natural es concetenar los 3 archivos de subtítulos que tengo en cada lengua para conseguir un solo archivo por cada película.

Sin embargo, tras meditarlo, consideré que la mejor manera de hacer la concatenación es intercalando palabras de las 3 lenguas. Es decir, en el archivo que combine los 3 idiomas, tendrá primero una palabra en español, luego una en inglés, luego otra en francés y así sucesivamente.

Para ello crée la función "merge_multilingual" en mi módulo "myutils". En caso de que un archivo de una lengua tenga más palabras que las otras dos, se copia directamente lo que resta de dicho archivo sin intercalar las otras lenguas.

Además, antes de hacer la concatenación, aplico el filtro de palabras de paro, tal y como lo he hecho en los anteriores cuadernos, ya que considero que si dejo las palabras más frecuentes solo terminarían siendo ruido e impedirían lograr el objetivo. Existe el problema de que no se filtrarán las mismas palabras en las tres lenguas, haciendo posible que en el corpus permanezcan algunas palabras en una sola lengua (y sin sus respectivas traducciones). Sin embargo, filtro a lo más 300 palabras de cada lengua, y como ya exploré en anteriores cuadernos, las primeras suelen ser verbos auxiliares, pronombres y preposiciones en las 3 lenguas.

In [95]:
```
stopwords_spa = myutils.intersect_most_common(frecs_spa, n = 300)
stopwords_eng = myutils.intersect_most_common(frecs_eng, n = 300)
stopwords_fre = myutils.intersect_most_common(frecs_fre, n = 300)

lists_spa = myutils.texts2lists(txts_spa, stopwords = stopwords_spa)
lists_eng = myutils.texts2lists(txts_eng, stopwords = stopwords_eng)
lists_fre = myutils.texts2lists(txts_fre, stopwords = stopwords_fre)

corpus = myutils.merge_multilingual(lists_spa, lists_eng, lists_fre)
```

A continuación muestro un ejemplo de un documento que combina las tres lenguas. Lo elegí para mostrarlo ya que considero que muestra varias cosas a tener en cuenta:

- Inicia con una traducción excelente, palabra por palabra, intercalando entre lenguas: "mutación mutation mutation clave key clé...." es justamente lo que buscaba español-inglés-francés. Esto no sucede en todos los documentos, ya que por lo regular las expresiones usadas en estas tres lenguas suelen dar un orden ligeramente diferente a las palabras. Además, también sucede que en muchos inicios de las películas aparecen textos, y normalmente estos no se incluyen en los subtítulos. Por ejemplo, el inicio de Star Wars es un texto en inglés largo que no se incluye en los archivos de sus subtítulos, pero en los subtítulos en español y en francés sí aparecen las traducciones del texto. Debido a esto existe cierto "desfase" entre los textos que extraje de los subtítulos de las películas de star wars, y esto ocasiona que no pueda suceder lo que aquí sucede de iniciar con traducciones específicas palabra por palabra.
- Conforme se avanza en el texto se comienza a notar un "desfase" en las traducciones. Para encontrar la palabra traducida entre las 3 lenguas hay que comenzar a ver en una "ventana" cada vez más grande. Conforme se avanza en el texto, este desfase se acentúa.
- Hacia el final del documento todas las palabras están en inglés. Esto se explica ya que, por alguna razón, el documento en inglés tenía más palabras (tras filtrar las stopwords) que los documentos de las otras lenguas. Sin embargo esto no es un problema, y lo prefiero así, ya que quizás LDA pueda reafirmar relaciones entre palabras de una misma lengua (de la lengua cuyo documento sea más largo en cada película) y esto a su vez permita conectar palabras entre lenguas distintas.

In [96]:
```
peli = 10
print("Ejemplo de documento combinado:")
print(" ".join(corpus[10]))
```

Ejemplo de documento combinado:
mutación mutation mutation clave key clé evolución evolution évolution permitir enable permettre convertir evolve espèce especie single suprême dominante celle d planète planeta organism processus proceso dominant ordinaire soler species m illier durar planet année mil process parfois mil slow évolution varios normall y géant ciento thousand pologne milenio thousand meridian evolución ofyears mis sissippi salto few avenir adelante hundred proche polonia millennium chute meri dian evolution niagara mississippi leap ensuite futuro forward rocheuses lejano niagara canada catarata falls puis niágara canadian centaine rocosas rockies ki lomètre canadiense few anchorage ciento hundred froid kilómetro mile idiot anch orage anchorage aventure frío cold lycée tonto point université aventura stupid marie cuándo otherwise david instituto adventure peine universidad high toucher marie school ambulance david college effleurer tocar marie approche ambulancia david ambulance tocar touch toucher os ambulance assister acercar touch prémice s ambulancia honey stade toque ambulance évolution asistir touch mutation comie nzo lady apparaître etapa gentleman puberté evolución beginning souvent humano stage déclencher mutación human stress pubertad evolution mlle debido mutation grey periodo manifest instructif estrés puberty apporter emocional often élémen t señorito trigger thème grey period débat educativo heighten mutant obstante e motional dangereux tema stress injuste central ms sénateur reunión grey kelly p alabra quite mauvais peligroso educational conducteur mutante however dangereux injusto fail permettre senador address conduire kelly issue permettre conducir focus sénateur coche hearing mutant peligroso word révéler permiso mutant condi tion conducir dangerous provoquer senador afraid hostilité mutante unfair viole nce admitir question cause condición senator exiger público kelly sénat miedo p erson vote hostilidad behind contre incluso wheel immatriculation violencia dan gerous mutant debido license forcer hostilidad drive mutant senado senator expo ser votar fact exposer contra mutant mutant registro forward cacher mutante rev eal re forzar themselves cacher luz publicly cache público fear liste luz hosti lity mutant público violence identifier esconder present etats mutante hostilit y unis miedo urge sénateur identificar senate kelly esconder vote traverser esc onder against mur lista mutant pénétrer mutante registration salle identificar force coffre estados mutant blanche unidos expose rumeur senador themselves mut ant kelly further capable illinois expose pénétrer cruzar themselves cerveau pa red mutant contrôler evitar community pensée cámara hide priver banco wonder vo lonté blanca afraid américains incluso identify choisir rumor themselves école existir hide élève mutante hide mutant poderoso ms prof penetrar grey mutant me nte list madame controlar identify messieurs pensamiento mutant mutant arrancar united exister libre states bel albedrío senator parmi otorgar kelly surtout pu eblo illinois dont estadounidense walk capable merecer wall eric derecho walk p oser decidir bank dont colegio vault réponse mutante white désespérer profesor house eric mutante house charles caballero senator argument mutante kelly longt emps real rumor race encima ms humain eric grey évoluer pregunta mutant rôder c uyo powerful charles respuesta enter espoir abandonar control retour eric thoug ht mets charles given bâton argumento free roue humanidad american représenter evolucionar deserve futur husmear decide eux dentro whether compter charlar chi ld alberta esperanza school nord esperanza mutant canada cambio teach emmener i nterponer mutant laughlin futuro lady laughlin charles gentleman vu norte truth tel alberta mutant argent canadá among défier laughlin above madame city eric m essieurs laughlin question sauveur city already couille caballero answer permet tre vuestro eric risque dinero charles crétin luchar argument madame caballero ago messieurs salvador mankind vainqueur pelota evolve champion valía since cag e personal sneak wolverine idiota charles pourboires caballero whatever eau gan ador hope bière rey hope ellis jaula return island lobezno future ancien propin a charles port es longer arrivé una matter immigrant ciudad far américain china laughlin ré distinto city ouvre cielo laughlin préparatif agua city sommet cerv eza gentleman nations isla walk unies ellis money terminer punto fight rassembl ement entrada lady chef inmigrante gentleman etats americano savior histoire pu erta whatever chef casi hit etats finalizar ball aborderontl preparación person al économie cumbre lady mondial naciones gentleman surl unidas tonight armement acontecimiento winner mutant prometer king effet mayor cage surla reunión wolve rine scène líder honey mondial mundial stick législateur líder water américain debatir beer concentrer economía ellis mutant mundial island vue tratado arriva

Tras crear los documentos combinados me dispuse a decidir cómo segmentarlos.

Al igual que en los cuadernos pasados, creo que segmentar cada documento en pequeños documentos resulta clave para que LDA funcione bien, y en esta ocasión creo que la segmentación es particularmente importante para capturar las relaciones de palabras en distintas lenguas.

Originalmente pensaba entrenar distintos modelos de LDA usando diferentes tamaños de segmentación. Pero terminé tomando la decisión de solo entrenar un modelo, usando un corpus que también combinara distintos niveles de segmentación. Particularmente me refiero a segmentar el corpus con un chunk_size de 50, 100 y 200 y combinar esos 3 conjuntos para obtener el corpus que utilizaré para entrenar el modelo.

Además de hacerlo por cuestiones de tiempo, también creo que podría ser beneficioso usar esta técnica ya que así las palabras que aparezcan en el contexto cercano de x palabra en diversas ocasiones tendrán 3 veces el peso que las que no. Por otro lado, así también me ocupo del asunto de los desfases en las traducciones, sin necesitar que 2 palabras traducidas se encuentren consecutivas dentro de los textos.

In [97]:
```
corpus1 = myutils.split_lists(corpus, chunk_size = 100, overlap_size = 50)
corpus2 = myutils.split_lists(corpus, chunk_size = 50, overlap_size = 25)
corpus3 = myutils.split_lists(corpus, chunk_size = 200, overlap_size = 100)
corpus = corpus1 + corpus2 + corpus3
```

In [98]:
```
dictionary_multi = corpora.Dictionary(corpus)
print("Número de tokens en corpus combinado:", len(dictionary_multi))
```

```
Número de tokens en corpus combinado: 76144
```

In [99]:
```
corpus = myutils.lists2bow(corpus, dictionary_multi)
print("Número de documentos tras segmentar:", len(corpus))
```

```
Número de documentos tras segmentar: 122413
```

In [186]:
```
dictionary_multi.save('pickles/dictionary_multi.dict')
```

Entrenando un modelo LDA sobre el corpus combinado

Para entrenar este modelo utilicé las observaciones que hice en anteriores cuadernos: preferí un gran número de iteraciones y un número pequeño de passes, además de preferir 100 tópicos en total.

In [55]:
```
model_multi = models.LdaModel(corpus, id2word=dictionary_multi, num_topics= 100
, passes = 6, iterations = 400)
```

```
/home/antoniovs/Escolar/Sem09/anal-txts/proyecto2/environment/lib/python3.6/sit
e-packages/gensim/models/ldamodel.py:1023: RuntimeWarning: divide by zero encou
ntered in log
  diff = np.log(self.expElogbeta)
```

In [58]:
```
model_multi.show_topics(num_topics = 50)
```

Out[58]:
```
[(43,
  '0.089*"bill" + 0.051*"buddy" + 0.039*"bar" + 0.037*"sexy" + 0.024*"fumar" +
0.021*"bonsoir" + 0.021*"fumer" + 0.015*"marie" + 0.013*"brooklyn" + 0.012*"men
teur"'),
 (71,
  '0.080*"normal" + 0.038*"christ" + 0.030*"star" + 0.029*"dick" + 0.026*"golpe
ar" + 0.019*"estrella" + 0.019*"sake" + 0.019*"matt" + 0.018*"loi" + 0.017*"she
riff"'),
 (15,
  '0.016*"verdadero" + 0.015*"coucher" + 0.014*"awesome" + 0.010*"marido" + 0.0
10*"college" + 0.010*"année" + 0.010*"regretter" + 0.009*"act" + 0.009*"espoir"
+ 0.009*"faute"'),
 (8,
  '0.091*"happy" + 0.054*"daughter" + 0.038*"sigh" + 0.032*"birthday" + 0.028*"
anniversaire" + 0.027*"cumpleaños" + 0.024*"anna" + 0.022*"list" + 0.015*"infec
tion" + 0.013*"probablement"'),
 (26,
  '0.027*"number" + 0.018*"travailler" + 0.014*"trabajar" + 0.013*"instant" + 0
.012*"glass" + 0.010*"amoureux" + 0.010*"rico" + 0.009*"anyway" + 0.009*"gros"
+ 0.009*"mas"'),
 (7,
  '0.067*"sang" + 0.054*"sangre" + 0.038*"toucher" + 0.027*"kane" + 0.019*"corp
s" + 0.018*"monstre" + 0.018*"bird" + 0.016*"enlever" + 0.016*"reculer" + 0.014
*"vérité"'),
 (12,
  '0.029*"carl" + 0.027*"maldición" + 0.023*"cambio" + 0.023*"perfect" + 0.022*
"pena" + 0.018*"suicide" + 0.016*"word" + 0.016*"extra" + 0.016*"catch" + 0.016
*"invite"'),
 (98,
  '0.083*"mentir" + 0.055*"walter" + 0.040*"mariage" + 0.037*"arreglar" + 0.032
*"marier" + 0.020*"écrire" + 0.019*"signe" + 0.019*"innocent" + 0.017*"lie" + 0
.017*"épouser"'),
 (75,
  '0.073*"question" + 0.053*"answer" + 0.023*"respuesta" + 0.021*"réponse" + 0.
021*"clothes" + 0.019*"fish" + 0.018*"obviously" + 0.018*"gamin" + 0.016*"foste
r" + 0.014*"memory"'),
 (99,
  '0.035*"londres" + 0.028*"business" + 0.028*"meeting" + 0.027*"conversation"
+ 0.020*"matrimonio" + 0.020*"are" + 0.020*"desastre" + 0.018*"marriage" + 0.01
7*"million" + 0.016*"habiller"'),
 (1,
  '0.074*"play" + 0.058*"lady" + 0.031*"sing" + 0.030*"écrire" + 0.030*"gentlem
an" + 0.029*"tonight" + 0.027*"song" + 0.021*"saltar" + 0.020*"girlfriend" + 0.
020*"vino"'),
 (13,
  '0.070*"house" + 0.050*"annie" + 0.050*"write" + 0.049*"jugar" + 0.048*"juego
" + 0.028*"amanda" + 0.025*"dinner" + 0.023*"pop" + 0.021*"grace" + 0.020*"cena
"'),
 (10,
  '0.019*"arma" + 0.018*"tirer" + 0.018*"ln" + 0.015*"soltar" + 0.015*"atrás" +
0.015*"feu" + 0.014*"arme" + 0.014*"gun" + 0.011*"lâcher" + 0.010*"cuidado"'),
 (37,
  '0.055*"message" + 0.040*"dress" + 0.029*"mensaje" + 0.027*"chuck" + 0.022*"d
emain" + 0.020*"allô" + 0.020*"pastel" + 0.019*"neither" + 0.019*"apartment" +
0.018*"tienda"'),
 (27,
  '0.051*"fight" + 0.028*"promise" + 0.028*"guerra" + 0.027*"war" + 0.024*"vict
oria" + 0.020*"battre" + 0.020*"guerre" + 0.018*"pelear" + 0.014*"luchar" + 0.0
12*"fear"'),
 (18,
  '0.090*"histoire" + 0.058*"music" + 0.050*"raconter" + 0.042*"mm" + 0.034*"po
int" + 0.027*"va" + 0.026*"playing" + 0.026*"punto" + 0.026*"lincoln" + 0.021*"
simplement"'),
```

In [60]:
```
model_multi.save("./pickles/modelos/model_multi.model")
```

A decir verdad me sorprendieron los resultados. Pensaba que saldría un caos total en los tópicos, pero incluso con una repasada rápida es posible ver algunas palabras que se agrupan con sus traducciones.

Explorando distancias entre palabras

Para poder agrupar las traducciones de distintas palabras se me ocurrió utiliza la representación aprendida de LDA para cada una de ellas y simplemente obtener la distancia euclidiana entre distintas palabras, para ver si son traducciones o no.

En esta sección expongo un pequeño experimento donde comparo los vectores de "money" "dinero" "argent" contra los de "dog" "perro" "chien".

In [101]:
```
topics_mat = model_multi.get_topics().T # matriz donde cada palabra es un rangl
ón de 100 columnas
```

In [102]:
```
# Obtengo los IDs de las distintas palabras
a = dictionary_multi.token2id['money']
b = dictionary_multi.token2id['dinero']
c = dictionary_multi.token2id['argent']

e = dictionary_multi.token2id['dog']
f = dictionary_multi.token2id['perro']
g = dictionary_multi.token2id['chien']
```

In [103]:
```
# Obtengo los vectores de las distintas palabras (de tamaño 100 cada uno)
a = topics_mat[a,:]
b = topics_mat[b,:]
c = topics_mat[c,:]

e = topics_mat[e,:]
f = topics_mat[f,:]
g = topics_mat[g,:]
```

In [104]:
```
# Obtengo las distancias de cada uno
distancias = scipy.spatial.distance.pdist([a,b,c,e,f,g])
distancias = scipy.spatial.distance.squareform(distancias)
```

In [105]:
```
print(distancias)
```

```
[[0.         0.00279951 0.01271675 0.07157819 0.07881712 0.06311633]
 [0.00279951 0.         0.00991723 0.06991249 0.07730756 0.06122084]
 [0.01271675 0.00991723 0.         0.06464502 0.07257879 0.05512906]
 [0.07157819 0.06991249 0.06464502 0.         0.00888158 0.01111013]
 [0.07881712 0.07730756 0.07257879 0.00888158 0.         0.01999171]
 [0.06311633 0.06122084 0.05512906 0.01111013 0.01999171 0.        ]]
```

Aclarando que "money", "dinero" y "argent" ocupan las posiciones 0, 1 y 2, y que "dog" "perro" y chien" ocupan la 3, 4, 5 se puede ver en la matriz de distancias que claramente las primeras tres palabras son más cercanas entre sí que con el otro grupo de palabras, y viceversa.

Obteniendo palabras más cercanas

Tras el breve experimento anterior, me dispuse a encontrar las palabras más cercanas a una palabra dada. Lo hago utilizando el método de fuerzo bruta, donde simplemente calculo la distancia de la palabra dada a todas las demás palabras y obtengo las 25 palabras más cercanas.

Aunque antes de implementarlo me preocupaba que fuera demasiado lento, resulta ejecutarse de inmediato. Al final, supongo que el vocabulario multilingüe de 72,000 palabras sigue siendo suficientemente pequeño como para permitirlo.

In [167]:
```
def palabras_mas_cercanas(palabra, topics_mat, n = 25):
    w_id = dictionary_multi.token2id[palabra]
    vec = topics_mat[w_id,:]
    distancias = scipy.spatial.distance.cdist([vec], topics_mat)[0]

    w_ids_cercanos = distancias.argsort()[:n]
    dists = distancias[w_ids_cercanos]

    for i, w_id in enumerate(w_ids_cercanos):
        w = dictionary_multi[w_id]
        print(w, "- {0:.5f}".format(dists[i]))
```

Primero comparo "money" "dinero" y "argent" corroborando que se encuentran muy cercanas entre sí. Palabras relacionadas en las 3 lenguas también aparecen con sus traducciones: sell y vendre así como pagar y pay. Y varias otras palabras que también aparecen aunque sin sus traducciones (caja, dollar, usd, buy) y algunas otras palabras que quizás no están muy relacionadas pero se encuentran muy cercanas (photo y foto, phantôme, ...)

In [168]:
```
palabras_mas_cercanas('money', topics_mat, n = 25)
```

```
money - 0.00000
dinero - 0.00280
argent - 0.01272
pagar - 0.01345
pay - 0.01371
photo - 0.01373
foto - 0.02197
payer - 0.02376
vendre - 0.02471
sell - 0.02497
wayne - 0.02669
dollar - 0.02681
vender - 0.02709
usd - 0.03019
caja - 0.03054
señorito - 0.03360
buy - 0.03399
talent - 0.03489
herself - 0.03544
fantôme - 0.03559
appreciate - 0.03616
criminal - 0.03628
fric - 0.03677
examen - 0.03689
clown - 0.03755
```

In [169]:
```
palabras_mas_cercanas('dinero', topics_mat, n = 25)
```

```
dinero - 0.00000
money - 0.00280
argent - 0.00992
pagar - 0.01065
photo - 0.01093
pay - 0.01122
foto - 0.01917
payer - 0.02099
vendre - 0.02191
sell - 0.02217
wayne - 0.02389
dollar - 0.02401
vender - 0.02429
usd - 0.02739
caja - 0.02774
señorito - 0.03081
buy - 0.03128
talent - 0.03209
herself - 0.03264
fantôme - 0.03279
appreciate - 0.03337
criminal - 0.03348
fric - 0.03397
examen - 0.03409
clown - 0.03475
```

In [170]:
```
palabras_mas_cercanas('argent', topics_mat, n = 25)
```

```
argent - 0.00000
pagar - 0.00074
photo - 0.00101
pay - 0.00570
foto - 0.00926
dinero - 0.00992
payer - 0.01131
vendre - 0.01200
sell - 0.01225
money - 0.01272
wayne - 0.01397
dollar - 0.01409
vender - 0.01438
usd - 0.01747
caja - 0.01782
señorito - 0.02097
buy - 0.02183
talent - 0.02217
herself - 0.02272
fantôme - 0.02287
appreciate - 0.02350
criminal - 0.02356
fric - 0.02406
examen - 0.02418
clown - 0.02483
```

Con "dog" "perro" y "chien" sucede algo parecido. Solo que ahora aparece mucho vocabulario asociado con lo escolar (school, école, escuela, lycée, verano, ...)

In [171]:
```
palabras_mas_cercanas('dog', topics_mat, n = 25)
```

```
dog - 0.00000
carajo - 0.00657
perro - 0.00888
chien - 0.01111
school - 0.01149
high - 0.01377
escuela - 0.01637
école - 0.02053
max - 0.02260
bobby - 0.02836
clair - 0.02883
grunt - 0.02948
miles - 0.04385
student - 0.04385
toute - 0.04399
façon - 0.04431
rid - 0.04459
mata - 0.04476
verano - 0.04626
lycée - 0.04644
cáncer - 0.04732
attraper - 0.04741
liste - 0.04828
estudiante - 0.04879
huge - 0.04908
```

In [172]:
```
palabras_mas_cercanas('perro', topics_mat, n = 25)
```

```
perro - 0.00000
school - 0.00261
dog - 0.00888
carajo - 0.01545
chien - 0.01999
high - 0.02173
escuela - 0.02526
école - 0.02941
max - 0.03148
bobby - 0.03724
clair - 0.03771
grunt - 0.03836
miles - 0.05273
student - 0.05273
toute - 0.05287
façon - 0.05310
rid - 0.05347
mata - 0.05363
verano - 0.05514
lycée - 0.05531
cáncer - 0.05620
attraper - 0.05627
liste - 0.05714
estudiante - 0.05768
huge - 0.05794
```

In [173]:
```
palabras_mas_cercanas('chien', topics_mat, n = 25)
```

```
chien - 0.00000
carajo - 0.00454
escuela - 0.00526
high - 0.00763
école - 0.00943
dog - 0.01111
max - 0.01149
bobby - 0.01725
clair - 0.01772
grunt - 0.01837
perro - 0.01999
school - 0.02260
miles - 0.03274
student - 0.03274
toute - 0.03288
façon - 0.03340
rid - 0.03348
mata - 0.03367
verano - 0.03515
lycée - 0.03533
cáncer - 0.03621
attraper - 0.03633
liste - 0.03720
estudiante - 0.03768
huge - 0.03801
```

Lo mismo con "amor" y "amour" donde aparecen palabras relacionadas a lo familiar en las tres lenguas.

In [174]:
```
palabras_mas_cercanas('amor', topics_mat, n = 25)
```

```
amor - 0.00000
familia - 0.00514
amour - 0.01109
famille - 0.01474
family - 0.02148
amar - 0.02360
malade - 0.03714
triste - 0.04029
wedding - 0.04141
souffrir - 0.04648
honest - 0.04892
extrañar - 0.04904
gentil - 0.04907
pitié - 0.04976
you - 0.04983
doux - 0.05141
toilette - 0.05145
además - 0.05246
papa - 0.05270
jamás - 0.05371
weyland - 0.05383
deserve - 0.05431
fix - 0.05434
envie - 0.05437
lover - 0.05445
```

In [175]:
```
palabras_mas_cercanas('amour', topics_mat, n = 25)
```

```
amour - 0.00000
familia - 0.00641
famille - 0.00758
amor - 0.01109
family - 0.01355
amar - 0.01365
malade - 0.02629
triste - 0.02945
wedding - 0.03058
souffrir - 0.03567
honest - 0.03810
extrañar - 0.03825
gentil - 0.03871
pitié - 0.03897
you - 0.03901
doux - 0.04060
toilette - 0.04063
además - 0.04176
papa - 0.04188
weyland - 0.04302
jamás - 0.04314
deserve - 0.04352
fix - 0.04363
lover - 0.04363
moneda - 0.04374
```

Con "victoire" "victoria" "victory" sucede algo ligeramente distinto. En esta ocasión las 3 palabras se encuentran todas relacionadas a temas relacionado con lo bélico; pero "victory" se encuentra particularmente relacionada a los nombres de personajes de star wars, mientras que "victoria" y "victoire" se relacionan a palabras negativas relacionadas con la lucha y la guerra.

In [180]:
```
palabras_mas_cercanas('victoire', topics_mat, n = 25)
```

```
victoire - 0.00000
mankind - 0.00128
pendiente - 0.00128
worthy - 0.00128
choper - 0.00128
temor - 0.00128
gwen - 0.00129
claquer - 0.00130
initial - 0.00130
illegal - 0.00130
warrior - 0.00132
traitor - 0.00132
referencia - 0.00133
noirs - 0.00135
formule - 0.00135
conquistar - 0.00137
generation - 0.00138
abro - 0.00139
anger - 0.00139
voto - 0.00140
traidor - 0.00144
sénateur - 0.00144
opposer - 0.00145
fácilmente - 0.00145
counter - 0.00146
```

In [182]:
```
palabras_mas_cercanas('victory', topics_mat, n = 25)
```

```
victory - 0.00000
cuisiner - 0.00002
forbid - 0.00003
dooku - 0.00006
strain - 0.00008
république - 0.00012
padmé - 0.00027
complot - 0.00027
naboo - 0.00035
alderaan - 0.00036
jabba - 0.00039
restore - 0.00039
decente - 0.00042
rebelión - 0.00044
droide - 0.00047
insupportable - 0.00048
atorar - 0.00050
comte - 0.00052
rebellion - 0.00054
grievous - 0.00054
droïde - 0.00059
pit - 0.00060
sabre - 0.00060
defraudar - 0.00062
ruine - 0.00063
```

In [183]:
```
palabras_mas_cercanas('victoria', topics_mat, n = 25)
```

```
victoria - 0.00000
war - 0.00330
guerra - 0.00417
pelear - 0.00539
guerre - 0.00610
promise - 0.00814
battre - 0.00834
luchar - 0.00939
fear - 0.01323
strong - 0.01341
destino - 0.01372
lay - 0.01461
throat - 0.01564
vencer - 0.01644
choc - 0.01653
fate - 0.01687
gift - 0.01697
soldier - 0.01700
colère - 0.01730
noble - 0.01730
prometer - 0.01735
lèvre - 0.01757
intervenir - 0.01761
create - 0.01772
aprovechar - 0.01778
```

Y finalmente me dispongo a ver palabras al azar en las 3 lenguas. Encontrando una fuerte influencia del corpus que utilicé: 'door' y 'sangre' se asocian con cosas de horror. Y por alguna razón que no me explico "fiesta" y "fête" se encuentran muy relacionadas a "sick" (¿quizás por expresiones como "that party was sick"?)

In [176]:
```
palabras_mas_cercanas('door', topics_mat, n = 25)
```

```
door - 0.00000
open - 0.00816
puerta - 0.01115
cerrar - 0.03042
ouvrer - 0.03855
arriba - 0.04003
close - 0.04242
abajo - 0.04299
acá - 0.04543
fermer - 0.04587
ouvrir - 0.04613
video - 0.04662
lock - 0.04664
camera - 0.04702
allá - 0.04714
cámara - 0.04774
demonio - 0.04776
andar - 0.04932
caméra - 0.05004
entrer - 0.05012
mover - 0.05030
afuera - 0.05038
outside - 0.05075
hurry - 0.05196
calm - 0.05203
```

In [177]:
```
palabras_mas_cercanas('sangre', topics_mat, n = 25)
```

```
sangre - 0.00000
sang - 0.01388
toucher - 0.01721
kane - 0.02781
monstre - 0.03599
bird - 0.03601
corps - 0.03635
reculer - 0.03830
enlever - 0.03879
vérité - 0.04036
laisse - 0.04097
pecho - 0.04142
sein - 0.04383
oscuridad - 0.04394
supe - 0.04487
magic - 0.04492
interior - 0.04511
peor - 0.04564
mapa - 0.04578
arracher - 0.04581
résister - 0.04590
créature - 0.04622
elección - 0.04625
sas - 0.04653
hunter - 0.04750
```

In [179]:
```
palabras_mas_cercanas('sick', topics_mat, n = 25)
```

```
sick - 0.00000
despierto - 0.00472
fête - 0.00481
rock - 0.00492
abuelo - 0.00519
chaud - 0.00543
habitación - 0.00547
dîner - 0.00617
chocolate - 0.00650
suck - 0.00668
infectar - 0.00708
caliente - 0.00711
fiesta - 0.00731
pill - 0.00733
sensible - 0.00737
baño - 0.00745
leche - 0.00746
trago - 0.00765
vaya - 0.00778
party - 0.00785
déconner - 0.00797
wake - 0.00800
froid - 0.00804
warm - 0.00805
winter - 0.00805
```

In [178]:
```
palabras_mas_cercanas('fiesta', topics_mat, n = 25)
```

```
fiesta - 0.00000
mmm - 0.00119
fête - 0.00393
habitación - 0.00531
party - 0.00601
wake - 0.00725
despierto - 0.00729
sick - 0.00731
rock - 0.00767
abuelo - 0.00947
sleep - 0.00949
chaud - 0.00983
dîner - 0.01015
suck - 0.01111
chocolate - 0.01128
hermoso - 0.01138
baño - 0.01160
tocar - 0.01179
bag - 0.01181
tomorrow - 0.01197
infectar - 0.01199
caliente - 0.01200
pill - 0.01230
sensible - 0.01231
vaya - 0.01243
```

Conclusiones y trabajo futuro

- La verdad los resultados superaron por mucho mis expectativas. Al ver cómo resultaban los documentos que combinaban las 3 lenguas perdí mucho la fe en esta idea, pero al final creo que sí se muestra que LDA logró capturar relaciones interesantes entre el vocabulario de las 3 lenguas.
- Al igual que en los otros cuadernos, creo que de tener más tiempo y recursos computacionales se podría experimentar con corpora más grande, y con distintos parámetros (número de documentos, parámetros de la segmentación, stopwords, etc...).
- Uno de los problemas de esta parte del proyecto fue no poder tener una métrica sencilla de evaluar los resultados. Lo mejor que se me podría ocurrir es hacer manualmente (o quizás con ayuda de google translator o diccionarios online) una lista de traducciones posibles entre palabras y usar eso para evaluar qué tan cercanas se encuentran las palabras de las que deberían ser sus traducciones.
- Por su parte, algo que también me hizo falta explorar es la idea de minar tópicos en las 3 lenguas distintas y encontrar una manera de encontrar los tópicos que son equivalente, o las palabras que son traducciones entre sí. Una forma que se me ocurrió requeriría también de una lista de traducciones y de encontrar los tópicos donde más palabras traducidas haya de una palabra original. Hacerlo para todo el vocabulario y hacer "match" entre los tópicos que más parezcan tener relaciones entre los distintos idiomas.

In [ ]:
```
```