Proyecto Final: Parte 2 - Tópicos en español

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

In [123]:
```
import os, pickle, logging
import myutils
from gensim import corpora, models
import gensim.matutils
import numpy as np
from sklearn.decomposition import PCA

logging.basicConfig(format='%(levelname)s : %(message)s', level=logging.DEBUG)
logging.root.level = logging.DEBUG # ipython sometimes messes up the logging se
tup; restore
```

Objetivo

- El objetivo de este cuaderno es tener una primera aproximación a la minería de tópicos en el corpus de los subtítulos, pero solo utilizando los documentos en español y revisando manualmente los parámetros de LDA.

Cargando Datos

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"))

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

```
INFO : loading Dictionary object from ./pickles/dictionary_spa.dict
DEBUG : {'kw': {}, 'mode': 'rb', 'uri': './pickles/dictionary_spa.dict'}
INFO : loaded ./pickles/dictionary_spa.dict
```

In [3]:
```
train_txts_spa, train_labels_spa, test_txts_spa, test_labels_spa = myutils.spli
t_training_txts(txts_spa, training_ratio = .85)
```

In [124]:
```
print("Documentos de entrenamiento:", len(train_txts_spa))
print("Documentos de prueba:", len(test_txts_spa))
```

```
Documentos de entrenamiento: 255
Documentos de prueba: 45
```

In [125]:
```
print("Etiquetas de prueba:") # (0 para acción, 1 para romance / comedia y 2 pa
ra horror)
print(train_labels_spa)
```

```
Etiquetas de prueba:
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2
, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]
```

Explorando los parámetros para el minado de tópicos

En general identifico los siguientes parámetros que podrían afectar al minado de tópicos:

- Número de tópicos
- Número de iteraciones y número de passes
- Tamaños de la segmentación del corpus (chunk_size y overlap_size)
- Palabras utilizadas como stopwords

Además, los parámetros anteriores podrían variar dependiendo de en qué género y en qué lengua se quieran minar los tópicos, y con qué objetivo.

En esta sección mi propósito será variar los parámetros manualmente y escoger los que vea más adecuados para posteriormente utilizarlos en la tarea de clasificación.

Debido a la gran cantidad de variables, me limitaré solamente al corpus en español sin considerar diferencias en el género de las películas

10 topicos, sin segmentar el corpus y sin stopwords

Como es posible observar, todos los tópicos son redundantes (y muy parecidos) ya que al no filtrar stopwords las palabras que son muy comunes acaparan todos los tópicos.

También se alcanza a ver en el log de Gensim que no todos los documentos convergen en el entrenamiento; pero aún con todo se alcanza una perplejidad de 80 (lo cual interpreto con que la redundancia de las palabras que son muy frecuentes ayuda a que el modelo salga bien en esta métrica).

In [6]:
```
corpus = myutils.texts2lists(train_txts_spa, stopwords = None)
corpus = myutils.lists2bow(corpus, dictionary_spa)

%time model_t10_ss_stw0 = models.LdaModel(corpus, id2word=dictionary_spa, num_t
opics=10, passes = 15, iterations = 400)
```

```
INFO : using symmetric alpha at 0.1
INFO : using symmetric eta at 0.1
INFO : using serial LDA version on this node
INFO : running online (multi-pass) LDA training, 10 topics, 15 passes over the
supplied corpus of 255 documents, updating model once every 255 documents, eval
uating perplexity every 255 documents, iterating 400x with a convergence thresh
old of 0.001000
DEBUG : bound: at document #0
INFO : -11.121 per-word bound, 2227.5 perplexity estimate based on a held-out c
orpus of 255 documents with 1601005 words
INFO : PROGRESS: pass 0, at document #255/255
DEBUG : performing inference on a chunk of 255 documents
DEBUG : 5/255 documents converged within 400 iterations
DEBUG : updating topics
INFO : topic #3 (0.100): 0.077*"el" + 0.038*"de" + 0.033*"que" + 0.032*"ser" +
0.030*"no" + 0.024*"uno" + 0.017*"estar" + 0.015*"en" + 0.012*"qué" + 0.012*"ha
ber"
INFO : topic #7 (0.100): 0.000*"el" + 0.000*"ser" + 0.000*"uno" + 0.000*"de" +
0.000*"que" + 0.000*"no" + 0.000*"estar" + 0.000*"en" + 0.000*"qué" + 0.000*"ir
"
INFO : topic #4 (0.100): 0.000*"el" + 0.000*"ser" + 0.000*"de" + 0.000*"no" + 0
.000*"que" + 0.000*"uno" + 0.000*"estar" + 0.000*"en" + 0.000*"me" + 0.000*"hab
er"
INFO : topic #2 (0.100): 0.003*"ser" + 0.002*"que" + 0.002*"no" + 0.002*"de" +
0.002*"qué" + 0.002*"uno" + 0.001*"ia" + 0.001*"sí" + 0.001*"te" + 0.001*"me"
INFO : topic #0 (0.100): 0.000*"el" + 0.000*"ser" + 0.000*"que" + 0.000*"de" +
0.000*"uno" + 0.000*"no" + 0.000*"estar" + 0.000*"en" + 0.000*"haber" + 0.000*"
ir"
INFO : topic diff=6.683697, rho=1.000000
DEBUG : bound: at document #0
INFO : -6.389 per-word bound, 83.8 perplexity estimate based on a held-out corp
us of 255 documents with 1601005 words
INFO : PROGRESS: pass 1, at document #255/255
DEBUG : performing inference on a chunk of 255 documents
DEBUG : 188/255 documents converged within 400 iterations
DEBUG : updating topics
INFO : topic #8 (0.100): 0.046*"el" + 0.039*"no" + 0.039*"ser" + 0.030*"que" +
0.029*"de" + 0.024*"estar" + 0.021*"uno" + 0.016*"ir" + 0.016*"en" + 0.014*"me"
INFO : topic #6 (0.100): 0.032*"el" + 0.024*"que" + 0.019*"de" + 0.019*"ser" +
0.017*"no" + 0.014*"estar" + 0.013*"uno" + 0.009*"me" + 0.009*"sí" + 0.009*"bie
n"
INFO : topic #9 (0.100): 0.004*"ser" + 0.003*"no" + 0.003*"que" + 0.003*"ir" +
0.002*"el" + 0.002*"de" + 0.002*"qué" + 0.002*"estar" + 0.002*"uno" + 0.001*"en
"
INFO : topic #7 (0.100): 0.000*"el" + 0.000*"ser" + 0.000*"uno" + 0.000*"de" +
0.000*"que" + 0.000*"no" + 0.000*"estar" + 0.000*"en" + 0.000*"qué" + 0.000*"ir
"
INFO : topic #5 (0.100): 0.061*"el" + 0.043*"no" + 0.039*"ser" + 0.031*"que" +
0.028*"de" + 0.027*"uno" + 0.022*"estar" + 0.016*"en" + 0.013*"qué" + 0.011*"po
r"
INFO : topic diff=0.343203, rho=0.577350
DEBUG : bound: at document #0
INFO : -6.366 per-word bound, 82.5 perplexity estimate based on a held-out corp
us of 255 documents with 1601005 words
INFO : PROGRESS: pass 2, at document #255/255
DEBUG : performing inference on a chunk of 255 documents
DEBUG : 213/255 documents converged within 400 iterations
DEBUG : updating topics
INFO : topic #3 (0.100): 0.077*"el" + 0.038*"de" + 0.033*"que" + 0.032*"ser" +
0.031*"no" + 0.025*"uno" + 0.017*"estar" + 0.015*"en" + 0.013*"haber" + 0.012*"
qué"
INFO : topic #2 (0.100): 0.001*"ser" + 0.000*"que" + 0.000*"no" + 0.000*"de" +
0.000*"qué" + 0.000*"uno" + 0.000*"ia" + 0.000*"sí" + 0.000*"te" + 0.000*"me"
```

```
CPU times: user 4min 57s, sys: 7min 16s, total: 12min 13s
Wall time: 1min 34s
```

In [7]:
```
model_t10_ss_stw0.show_topics()
```

Out[7]:
```
[(0,
  '0.000*"el" + 0.000*"ser" + 0.000*"que" + 0.000*"de" + 0.000*"uno" + 0.000*"n
o" + 0.000*"estar" + 0.000*"haber" + 0.000*"en" + 0.000*"qué"'),
 (1,
  '0.021*"ia" + 0.020*"ei" + 0.014*"io" + 0.013*"mary" + 0.008*"ted" + 0.007*"w
arren" + 0.007*"ai" + 0.006*"ios" + 0.006*"aigo" + 0.005*"es"'),
 (2,
  '0.000*"ser" + 0.000*"que" + 0.000*"no" + 0.000*"de" + 0.000*"qué" + 0.000*"u
no" + 0.000*"ia" + 0.000*"sí" + 0.000*"te" + 0.000*"me"'),
 (3,
  '0.081*"el" + 0.038*"de" + 0.032*"que" + 0.031*"ser" + 0.030*"no" + 0.025*"un
o" + 0.017*"estar" + 0.016*"en" + 0.014*"haber" + 0.011*"qué"'),
 (4,
  '0.000*"el" + 0.000*"ser" + 0.000*"de" + 0.000*"no" + 0.000*"que" + 0.000*"un
o" + 0.000*"estar" + 0.000*"en" + 0.000*"haber" + 0.000*"me"'),
 (5,
  '0.063*"el" + 0.038*"ser" + 0.037*"no" + 0.034*"que" + 0.033*"de" + 0.026*"un
o" + 0.020*"estar" + 0.015*"en" + 0.014*"qué" + 0.013*"me"'),
 (6,
  '0.014*"peter" + 0.011*"sydney" + 0.010*"zooey" + 0.009*"weir" + 0.007*"justi
n" + 0.007*"miller" + 0.006*"zno" + 0.006*"zpor" + 0.005*"starck" + 0.005*"nave
"'),
 (7,
  '0.000*"el" + 0.000*"ser" + 0.000*"uno" + 0.000*"de" + 0.000*"que" + 0.000*"n
o" + 0.000*"estar" + 0.000*"qué" + 0.000*"ir" + 0.000*"en"'),
 (8,
  '0.055*"el" + 0.040*"ser" + 0.036*"no" + 0.035*"que" + 0.031*"de" + 0.024*"un
o" + 0.022*"estar" + 0.015*"ir" + 0.015*"en" + 0.014*"me"'),
 (9,
  '0.000*"ser" + 0.000*"no" + 0.000*"que" + 0.000*"ir" + 0.000*"el" + 0.000*"de
" + 0.000*"qué" + 0.000*"estar" + 0.000*"uno" + 0.000*"en"')]
```

10 tópicos, segmentado y con muchas stopwords

Para contrastar, a continuación muestro los resultados de segmentar (usando un chunk_size de 200 y un overlap de 100) y de utilizar muchas stopwords (obtenidas de la intersección de las 1000 palabras más frecuentes en los 3 géneros).

Creo que los resultados son mucho mejores que el anterior, pero me da la sensación de que quizás se está sobre-ajustando demasiado. Es decir, que quizás está minando tópicos que pertenecen a una sola película específica; esto lo intuyo ya que muchas de las palabras que aparecen son nombres de personajes directamente.

Por otro lado, en el log de Gensim es posible ver que todos los documentos convergen en tan solo unas pasadas. Por otro lado, la perplejidad es de 525.

In [8]:
```
stopwords = myutils.intersect_most_common(frecs_spa, n = 1000)
corpus = myutils.texts2lists(train_txts_spa, stopwords = stopwords)
corpus = myutils.split_lists(corpus, chunk_size = 200, overlap_size = 100)
corpus = myutils.lists2bow(corpus, dictionary_spa)

%time model_t10_cs_stw1000 = models.LdaModel(corpus, id2word=dictionary_spa, nu
m_topics=10, passes = 15, iterations = 400)
```

```
INFO : using symmetric alpha at 0.1
INFO : using symmetric eta at 0.1
INFO : using serial LDA version on this node
INFO : running online (multi-pass) LDA training, 10 topics, 15 passes over the
supplied corpus of 3263 documents, updating model once every 2000 documents, ev
aluating perplexity every 3263 documents, iterating 400x with a convergence thr
eshold of 0.001000
INFO : PROGRESS: pass 0, at document #2000/3263
DEBUG : performing inference on a chunk of 2000 documents
DEBUG : 946/2000 documents converged within 400 iterations
DEBUG : updating topics
INFO : merging changes from 2000 documents into a model of 3263 documents
INFO : topic #2 (0.100): 0.003*"nave" + 0.003*"príncipe" + 0.002*"luke" + 0.002
*"vera" + 0.002*"cantar" + 0.002*"ataque" + 0.002*"misión" + 0.002*"princesa" +
0.001*"marcus" + 0.001*"ben"
INFO : topic #0 (0.100): 0.003*"nave" + 0.003*"wayne" + 0.002*"programa" + 0.00
1*"planeta" + 0.001*"esperanza" + 0.001*"peter" + 0.001*"destruir" + 0.001*"ene
migo" + 0.001*"objetivo" + 0.001*"maestro"
INFO : topic #7 (0.100): 0.003*"barco" + 0.002*"césar" + 0.002*"pirata" + 0.002
*"vuestro" + 0.002*"tai" + 0.002*"sparrow" + 0.002*"besar" + 0.002*"ass" + 0.00
2*"ok" + 0.002*"kick"
INFO : topic #3 (0.100): 0.002*"ei" + 0.002*"nave" + 0.002*"ia" + 0.002*"greg"
+ 0.001*"cita" + 0.001*"usd" + 0.001*"morfeo" + 0.001*"gato" + 0.001*"vera" + 0
.001*"jake"
INFO : topic #4 (0.100): 0.003*"nave" + 0.003*"pueblo" + 0.003*"general" + 0.00
2*"arca" + 0.002*"jones" + 0.002*"máximo" + 0.002*"roma" + 0.002*"sargento" + 0
.002*"seguridad" + 0.002*"sistema"
INFO : topic diff=7.242285, rho=1.000000
DEBUG : bound: at document #0
INFO : -11.059 per-word bound, 2133.8 perplexity estimate based on a held-out c
orpus of 1263 documents with 228736 words
INFO : PROGRESS: pass 0, at document #3263/3263
DEBUG : performing inference on a chunk of 1263 documents
DEBUG : 1237/1263 documents converged within 400 iterations
DEBUG : updating topics
INFO : merging changes from 1263 documents into a model of 3263 documents
INFO : topic #9 (0.100): 0.003*"april" + 0.002*"alex" + 0.002*"bailar" + 0.002*
"beth" + 0.002*"sexo" + 0.002*"flash" + 0.002*"nancy" + 0.002*"claire" + 0.002*
"cama" + 0.001*"gracioso"
INFO : topic #1 (0.100): 0.003*"carajo" + 0.003*"carolyn" + 0.002*"nave" + 0.00
2*"emily" + 0.002*"josh" + 0.002*"película" + 0.002*"lynn" + 0.002*"claire" + 0
.002*"jane" + 0.002*"walter"
INFO : topic #3 (0.100): 0.003*"carl" + 0.002*"gay" + 0.002*"carajo" + 0.002*"c
ita" + 0.002*"cama" + 0.002*"jamie" + 0.002*"gato" + 0.002*"david" + 0.002*"sex
o" + 0.002*"relación"
INFO : topic #5 (0.100): 0.003*"danny" + 0.003*"jeff" + 0.002*"daniel" + 0.002*
"cama" + 0.002*"maldición" + 0.002*"abraham" + 0.002*"papi" + 0.002*"rachel" +
0.002*"carajo" + 0.002*"liz"
INFO : topic #0 (0.100): 0.007*"vampiro" + 0.003*"pablo" + 0.003*"carajo" + 0.0
02*"peter" + 0.002*"asesino" + 0.002*"cáncer" + 0.002*"nave" + 0.002*"jinete" +
0.002*"morder" + 0.002*"quemar"
INFO : topic diff=1.823339, rho=0.707107
INFO : PROGRESS: pass 1, at document #2000/3263
DEBUG : performing inference on a chunk of 2000 documents
DEBUG : 1995/2000 documents converged within 400 iterations
DEBUG : updating topics
INFO : merging changes from 2000 documents into a model of 3263 documents
INFO : topic #6 (0.100): 0.005*"nave" + 0.002*"brujo" + 0.002*"soldado" + 0.002
*"sistema" + 0.002*"sargento" + 0.002*"planeta" + 0.002*"eggsy" + 0.002*"destru
ir" + 0.002*"george" + 0.002*"misión"
INFO : topic #2 (0.100): 0.003*"príncipe" + 0.002*"ben" + 0.002*"relación" + 0.
002*"acostar" + 0.002*"sexo" + 0.002*"annie" + 0.002*"vera" + 0.002*"baño" + 0.
002*"canción" + 0.002*"enamorar"
```

```
CPU times: user 50.8 s, sys: 181 ms, total: 51 s
Wall time: 50.8 s
```

In [9]:
```
print("Número de stopwords usadas:", len(stopwords))
print("Número de documentos tras segmentar:", len(corpus))

model_t10_cs_stw1000.show_topics()
```

```
Número de stopwords usadas: 666
Número de documentos tras segmentar: 3263
```

Out[9]:
```
[(0,
  '0.007*"vampiro" + 0.004*"wayne" + 0.004*"johnny" + 0.003*"pablo" + 0.003*"pu
eblo" + 0.003*"jake" + 0.003*"espada" + 0.003*"espíritu" + 0.003*"infierno" + 0
.002*"ejército"'),
 (1,
  '0.009*"nave" + 0.004*"sistema" + 0.004*"destruir" + 0.003*"código" + 0.003*"
misión" + 0.003*"seguridad" + 0.003*"robot" + 0.002*"piloto" + 0.002*"base" + 0
.002*"atacar"'),
 (2,
  '0.005*"película" + 0.005*"ben" + 0.004*"annie" + 0.004*"relación" + 0.004*"m
ike" + 0.004*"acostar" + 0.003*"cita" + 0.003*"sexo" + 0.003*"canción" + 0.003*
"pene"'),
 (3,
  '0.004*"carajo" + 0.003*"gay" + 0.003*"sexo" + 0.003*"cama" + 0.003*"carl" +
0.003*"josh" + 0.003*"peter" + 0.002*"fantástico" + 0.002*"larry" + 0.002*"vera
"'),
 (4,
  '0.006*"janet" + 0.005*"peter" + 0.004*"sydney" + 0.004*"caleb" + 0.003*"empe
rador" + 0.003*"zooey" + 0.003*"ed" + 0.003*"thomasin" + 0.003*"victoria" + 0.0
03*"muerto"'),
 (5,
  '0.003*"danny" + 0.003*"charlie" + 0.003*"hospital" + 0.003*"jeff" + 0.003*"a
manda" + 0.003*"accidente" + 0.003*"rachel" + 0.002*"seguridad" + 0.002*"cinta"
+ 0.002*"daniel"'),
 (6,
  '0.005*"nave" + 0.004*"brujo" + 0.003*"stark" + 0.003*"planeta" + 0.003*"sold
ado" + 0.002*"sargento" + 0.002*"princesa" + 0.002*"helen" + 0.002*"tony" + 0.0
02*"steve"'),
 (7,
  '0.005*"ok" + 0.004*"sidney" + 0.004*"barco" + 0.004*"aqui" + 0.004*"hey" + 0
.003*"sutter" + 0.003*"billy" + 0.003*"asi" + 0.003*"vuestro" + 0.003*"michael"
'),
 (8,
  '0.007*"blade" + 0.005*"mutante" + 0.005*"ted" + 0.005*"ei" + 0.005*"ia" + 0.
004*"bianca" + 0.004*"logan" + 0.004*"rod" + 0.004*"lars" + 0.004*"gale"'),
 (9,
  '0.005*"nancy" + 0.003*"april" + 0.003*"claire" + 0.003*"max" + 0.003*"bailar
" + 0.002*"phil" + 0.002*"tina" + 0.002*"johanna" + 0.002*"beth" + 0.002*"flash
"')]
```

10 tópicos, segmentado y con algunas stopwords

Ahora lo ejecuté con los mismos parámetros que en el anterior salvo que usando menos stopwords (tomando solo la intersección de las 300 más comunes de los 3 géneros).

Como presentía, el número de nombres de personajes se redujo en comparación con los anteriores resultados, ya que al haber más palabras en común entre todos los documentos, se da más lugar a la formación de relaciones entre documentos. Sin embargo, el esto también provoca que exista cierta redundancia entre los tópicos (algunas palabras que se repiten en los distintos tópicos). Pese a esto, creo que prefiero estos parámetros a los anteriores.

También me llama la atención el tópico que agrupa interjecciones ('eh', 'oh', 'ay' y 'ah') y el que las haya agrupado con 'carajo' y 'cariño' (¿quizás también las consideró como interjecciones o como palabras que usualmente van junto a ellas?)

Cabe notar que al igual que con los anteriores resultados, según el log de gensim todos los documentos convergieron oportunamente; y la perplejidad es menor en esta ocasión llegando a 367.

In [10]:

```
stopwords = myutils.intersect_most_common(frecs_spa, n = 300)
corpus = myutils.texts2lists(train_txts_spa, stopwords = stopwords)
corpus = myutils.split_lists(corpus, chunk_size = 200, overlap_size = 100)
corpus = myutils.lists2bow(corpus, dictionary_spa)

%time model_t10_cs_stw300 = models.LdaModel(corpus, id2word=dictionary_spa, num
_topics=10, passes = 15, iterations = 400)
```

```
INFO : using symmetric alpha at 0.1
INFO : using symmetric eta at 0.1
INFO : using serial LDA version on this node
INFO : running online (multi-pass) LDA training, 10 topics, 15 passes over the
supplied corpus of 4865 documents, updating model once every 2000 documents, ev
aluating perplexity every 4865 documents, iterating 400x with a convergence thr
eshold of 0.001000
INFO : PROGRESS: pass 0, at document #2000/4865
DEBUG : performing inference on a chunk of 2000 documents
DEBUG : 906/2000 documents converged within 400 iterations
DEBUG : updating topics
INFO : merging changes from 2000 documents into a model of 4865 documents
INFO : topic #0 (0.100): 0.005*"capitán" + 0.004*"rey" + 0.003*"nave" + 0.002*"
john" + 0.002*"alto" + 0.002*"guerra" + 0.002*"orden" + 0.002*"ud" + 0.002*"ráp
ido" + 0.002*"tranquilo"
INFO : topic #9 (0.100): 0.003*"arma" + 0.002*"salvar" + 0.002*"detener" + 0.00
2*"guerra" + 0.002*"dinero" + 0.002*"policía" + 0.002*"trabajar" + 0.002*"rápid
o" + 0.002*"regresar" + 0.002*"mover"
INFO : topic #2 (0.100): 0.003*"jack" + 0.003*"nave" + 0.003*"misión" + 0.002*"
salvar" + 0.002*"guerra" + 0.002*"equipo" + 0.002*"arma" + 0.002*"disparar" + 0
.002*"humano" + 0.002*"funcionar"
INFO : topic #8 (0.100): 0.004*"capitán" + 0.003*"jack" + 0.002*"arma" + 0.002*
"baxter" + 0.002*"doctor" + 0.002*"salvar" + 0.002*"nave" + 0.002*"allá" + 0.00
2*"regresar" + 0.002*"humano"
INFO : topic #4 (0.100): 0.004*"sueño" + 0.003*"nave" + 0.003*"dinero" + 0.003*
"rápido" + 0.002*"trabajar" + 0.002*"diablo" + 0.002*"arma" + 0.002*"orden" + 0
.002*"realidad" + 0.002*"funcionar"
INFO : topic diff=8.067354, rho=1.000000
INFO : PROGRESS: pass 0, at document #4000/4865
DEBUG : performing inference on a chunk of 2000 documents
DEBUG : 1929/2000 documents converged within 400 iterations
DEBUG : updating topics
INFO : merging changes from 2000 documents into a model of 4865 documents
INFO : topic #2 (0.100): 0.006*"ted" + 0.004*"jack" + 0.003*"nave" + 0.003*"and
rew" + 0.003*"tío" + 0.002*"pues" + 0.002*"semana" + 0.002*"min" + 0.002*"incre
íble" + 0.002*"adiós"
INFO : topic #7 (0.100): 0.002*"tío" + 0.002*"mover" + 0.002*"pues" + 0.002*"mu
erte" + 0.002*"eh" + 0.002*"trabajar" + 0.002*"regresar" + 0.002*"arma" + 0.002
*"dentro" + 0.002*"tocar"
INFO : topic #1 (0.100): 0.005*"tío" + 0.004*"joder" + 0.003*"pues" + 0.003*"en
cantar" + 0.003*"ben" + 0.003*"ah" + 0.003*"eh" + 0.003*"novio" + 0.002*"casar"
+ 0.002*"amor"
INFO : topic #0 (0.100): 0.007*"ei" + 0.003*"capitán" + 0.003*"barry" + 0.003*"
sidney" + 0.002*"amor" + 0.002*"miller" + 0.002*"nave" + 0.002*"loco" + 0.002*"
rey" + 0.002*"película"
INFO : topic #5 (0.100): 0.004*"oh" + 0.003*"annie" + 0.003*"adiós" + 0.003*"en
cantar" + 0.002*"príncipe" + 0.002*"casar" + 0.002*"amar" + 0.002*"fiesta" + 0.
002*"comer" + 0.002*"puerta"
INFO : topic diff=2.344690, rho=0.707107
DEBUG : bound: at document #0
INFO : -9.837 per-word bound, 914.8 perplexity estimate based on a held-out cor
pus of 865 documents with 159800 words
INFO : PROGRESS: pass 0, at document #4865/4865
DEBUG : performing inference on a chunk of 865 documents
DEBUG : 840/865 documents converged within 400 iterations
DEBUG : updating topics
INFO : merging changes from 865 documents into a model of 4865 documents
INFO : topic #6 (0.100): 0.005*"oh" + 0.005*"diablo" + 0.003*"genial" + 0.003*"
puerta" + 0.003*"comer" + 0.003*"foto" + 0.002*"puta" + 0.002*"sexo" + 0.002*"p
erro" + 0.002*"jay"
INFO : topic #5 (0.100): 0.005*"janet" + 0.004*"oh" + 0.003*"puerta" + 0.003*"c
omer" + 0.002*"príncipe" + 0.002*"libro" + 0.002*"adiós" + 0.002*"auto" + 0.002
*"amar" + 0.002*"loco"
```

```
CPU times: user 1min 9s, sys: 176 ms, total: 1min 9s
Wall time: 1min 9s
```

In [11]:
```
print("Número de stopwords usadas:", len(stopwords))
print("Número de documentos tras segmentar:", len(corpus))

model_t10_cs_stw300.show_topics()
```

```
Número de stopwords usadas: 230
Número de documentos tras segmentar: 4865
```

Out[11]:
```
[(0,
  '0.008*"capitán" + 0.007*"nave" + 0.005*"rey" + 0.005*"john" + 0.004*"guerra"
+ 0.004*"fuerza" + 0.004*"tierra" + 0.004*"barco" + 0.003*"salvar" + 0.003*"ord
en"'),
 (1,
  '0.011*"eh" + 0.011*"tío" + 0.011*"joder" + 0.008*"pues" + 0.007*"ah" + 0.006
*"os" + 0.005*"coche" + 0.005*"blade" + 0.005*"ay" + 0.004*"tucker"'),
 (2,
  '0.005*"correr" + 0.005*"rápido" + 0.005*"puerta" + 0.004*"subir" + 0.004*"al
lá" + 0.004*"mover" + 0.004*"cuidado" + 0.004*"bajar" + 0.003*"señal" + 0.003*"
equipo"'),
 (3,
  '0.003*"diablo" + 0.003*"ud" + 0.003*"caleb" + 0.003*"charlie" + 0.003*"negro
" + 0.003*"dinero" + 0.003*"familia" + 0.002*"thomasin" + 0.002*"pecado" + 0.00
2*"mary"'),
 (4,
  '0.004*"habitación" + 0.004*"sueño" + 0.004*"josh" + 0.004*"cariño" + 0.004*"
comer" + 0.003*"escribir" + 0.003*"miedo" + 0.003*"amor" + 0.003*"sra" + 0.003*
"papi"'),
 (5,
  '0.007*"max" + 0.006*"número" + 0.005*"oh" + 0.005*"príncipe" + 0.005*"victor
ia" + 0.004*"comer" + 0.003*"princesa" + 0.003*"julie" + 0.003*"nueva" + 0.003*
"humano"'),
 (6,
  '0.008*"oh" + 0.005*"genial" + 0.004*"semana" + 0.004*"divertir" + 0.004*"enc
antar" + 0.004*"realmente" + 0.004*"novio" + 0.004*"lindo" + 0.004*"fiesta" + 0
.003*"adiós"'),
 (7,
  '0.005*"sangre" + 0.005*"arma" + 0.005*"puerta" + 0.004*"muerte" + 0.004*"mov
er" + 0.004*"demonio" + 0.004*"luz" + 0.003*"detener" + 0.003*"janet" + 0.003*"
disparar"'),
 (8,
  '0.004*"cantar" + 0.004*"ei" + 0.004*"feliz" + 0.003*"comer" + 0.003*"srta" +
0.003*"barnabas" + 0.003*"navidad" + 0.003*"amar" + 0.003*"doctor" + 0.003*"pel
ícula"'),
 (9,
  '0.005*"juego" + 0.004*"libro" + 0.004*"diablo" + 0.004*"walter" + 0.004*"per
ro" + 0.004*"número" + 0.003*"chris" + 0.003*"alex" + 0.003*"carajo" + 0.003*"e
scribir"')]
```

10 tópicos, sin segmentar, con algunas stopwords

Manteniendo las mismas stopwords que en el experimento pasado, pero esta vez sin segmentar el corpus (es decir, tomando los documentos tal cual están).

Como era de esperarse, aunque admito que me sorprendió, cada tópico se refiere claramente a una película (o a una saga) en específico.

In [12]:

```
stopwords = myutils.intersect_most_common(frecs_spa, n = 300)
corpus = myutils.texts2lists(train_txts_spa, stopwords = stopwords)
# corpus = myutils.split_lists(corpus, chunk_size = 200, overlap_size = 100)
corpus = myutils.lists2bow(corpus, dictionary_spa)

%time model_t10_ss_stw300 = models.LdaModel(corpus, id2word=dictionary_spa, num
_topics=10, passes = 15, iterations = 400)
```

```
INFO : using symmetric alpha at 0.1
INFO : using symmetric eta at 0.1
INFO : using serial LDA version on this node
INFO : running online (multi-pass) LDA training, 10 topics, 15 passes over the
supplied corpus of 255 documents, updating model once every 255 documents, eval
uating perplexity every 255 documents, iterating 400x with a convergence thresh
old of 0.001000
DEBUG : bound: at document #0
INFO : -11.689 per-word bound, 3302.4 perplexity estimate based on a held-out c
orpus of 255 documents with 473646 words
INFO : PROGRESS: pass 0, at document #255/255
DEBUG : performing inference on a chunk of 255 documents
DEBUG : 2/255 documents converged within 400 iterations
DEBUG : updating topics
INFO : topic #9 (0.100): 0.003*"capitán" + 0.002*"ciudad" + 0.002*"arma" + 0.00
2*"trabajar" + 0.002*"policía" + 0.002*"salvar" + 0.002*"puerta" + 0.002*"joder
" + 0.002*"jack" + 0.002*"pues"
INFO : topic #5 (0.100): 0.002*"puerta" + 0.002*"ocurrir" + 0.002*"mary" + 0.00
2*"semana" + 0.002*"trabajar" + 0.002*"ia" + 0.002*"rápido" + 0.002*"regresar"
+ 0.002*"cuidado" + 0.002*"mover"
INFO : topic #1 (0.100): 0.002*"puerta" + 0.002*"regresar" + 0.002*"mover" + 0.
002*"guerra" + 0.002*"arma" + 0.002*"correr" + 0.002*"cuidado" + 0.002*"comer"
+ 0.002*"agua" + 0.002*"rey"
INFO : topic #0 (0.100): 0.004*"eh" + 0.003*"jack" + 0.002*"tío" + 0.002*"traba
jar" + 0.002*"ay" + 0.002*"rápido" + 0.002*"correr" + 0.002*"coche" + 0.002*"en
cantar" + 0.002*"cara"
INFO : topic #7 (0.100): 0.002*"oh" + 0.002*"muerte" + 0.002*"comer" + 0.002*"t
rabajar" + 0.002*"amar" + 0.001*"elle" + 0.001*"emperador" + 0.001*"ud" + 0.001
*"genial" + 0.001*"ganar"
INFO : topic diff=5.523079, rho=1.000000
DEBUG : bound: at document #0
INFO : -9.036 per-word bound, 524.8 perplexity estimate based on a held-out cor
pus of 255 documents with 473646 words
INFO : PROGRESS: pass 1, at document #255/255
DEBUG : performing inference on a chunk of 255 documents
DEBUG : 238/255 documents converged within 400 iterations
DEBUG : updating topics
INFO : topic #7 (0.100): 0.003*"oh" + 0.003*"vampiro" + 0.002*"elle" + 0.002*"s
angre" + 0.002*"muerte" + 0.002*"amar" + 0.002*"dragón" + 0.002*"emperador" + 0
.002*"comer" + 0.002*"clase"
INFO : topic #9 (0.100): 0.004*"capitán" + 0.003*"policía" + 0.002*"salvar" + 0
.002*"ciudad" + 0.002*"arma" + 0.002*"muerte" + 0.002*"wayne" + 0.002*"joder" +
0.002*"puerta" + 0.002*"orden"
INFO : topic #4 (0.100): 0.002*"oh" + 0.002*"semana" + 0.002*"amor" + 0.002*"ge
nial" + 0.002*"encantar" + 0.002*"rápido" + 0.002*"loco" + 0.002*"realmente" +
0.002*"feliz" + 0.002*"adiós"
INFO : topic #2 (0.100): 0.003*"príncipe" + 0.002*"semana" + 0.002*"carajo" + 0
.002*"rey" + 0.002*"john" + 0.002*"comprar" + 0.002*"fiesta" + 0.002*"wakanda"
+ 0.002*"regresar" + 0.002*"chris"
INFO : topic #6 (0.100): 0.003*"buddy" + 0.003*"john" + 0.002*"barry" + 0.002*"
oh" + 0.002*"kaiju" + 0.002*"comer" + 0.002*"animal" + 0.002*"srta" + 0.002*"tr
abajar" + 0.002*"mover"
INFO : topic diff=0.739822, rho=0.577350
DEBUG : bound: at document #0
INFO : -8.878 per-word bound, 470.5 perplexity estimate based on a held-out cor
pus of 255 documents with 473646 words
INFO : PROGRESS: pass 2, at document #255/255
DEBUG : performing inference on a chunk of 255 documents
DEBUG : 251/255 documents converged within 400 iterations
DEBUG : updating topics
INFO : topic #8 (0.100): 0.004*"nave" + 0.003*"capitán" + 0.003*"arma" + 0.003*
"fuerza" + 0.002*"puerta" + 0.002*"mover" + 0.002*"salvar" + 0.002*"regresar" +
0.002*"rápido" + 0.002*"luz"
```

```
CPU times: user 2min 59s, sys: 4min 38s, total: 7min 38s
Wall time: 59 s
```

In [13]:
```
print("Número de stopwords usadas:", len(stopwords))
print("Número de documentos tras segmentar:", len(corpus))

model_t10_ss_stw300.show_topics()
```

```
Número de stopwords usadas: 230
Número de documentos tras segmentar: 255
```

Out[13]:
```
[(0,
  '0.004*"eh" + 0.004*"jack" + 0.003*"scott" + 0.003*"coche" + 0.003*"sutter" +
0.002*"tío" + 0.002*"ay" + 0.002*"novio" + 0.002*"misión" + 0.002*"pues"'),
 (1,
  '0.003*"rey" + 0.003*"asgard" + 0.003*"guerra" + 0.003*"regresar" + 0.003*"pu
erta" + 0.003*"thor" + 0.003*"tierra" + 0.002*"sueño" + 0.002*"cerrar" + 0.002*
"correr"'),
 (2,
  '0.004*"john" + 0.004*"príncipe" + 0.003*"chris" + 0.002*"carajo" + 0.002*"re
y" + 0.002*"semana" + 0.002*"wakanda" + 0.002*"comprar" + 0.002*"elegir" + 0.00
2*"princesa"'),
 (3,
  '0.004*"oh" + 0.003*"genial" + 0.003*"cariño" + 0.003*"encantar" + 0.003*"tra
bajar" + 0.002*"tío" + 0.002*"semana" + 0.002*"dinero" + 0.002*"comer" + 0.002*
"jack"'),
 (4,
  '0.003*"oh" + 0.003*"amor" + 0.003*"genial" + 0.003*"encantar" + 0.003*"seman
a" + 0.003*"casar" + 0.003*"feliz" + 0.002*"realmente" + 0.002*"adiós" + 0.002*
"número"'),
 (5,
  '0.003*"ia" + 0.003*"mary" + 0.002*"ei" + 0.002*"io" + 0.002*"brujo" + 0.002*
"ocurrir" + 0.002*"comer" + 0.002*"corazón" + 0.002*"spider" + 0.002*"man"'),
 (6,
  '0.004*"buddy" + 0.003*"barry" + 0.003*"john" + 0.003*"srta" + 0.003*"baxter"
+ 0.002*"animal" + 0.002*"kaiju" + 0.002*"comer" + 0.002*"adiós" + 0.002*"oh"')
,
 (7,
  '0.007*"vampiro" + 0.004*"blade" + 0.004*"diana" + 0.004*"dragón" + 0.003*"sa
ngre" + 0.003*"guerra" + 0.003*"elle" + 0.003*"oh" + 0.003*"akeem" + 0.003*"hic
cup"'),
 (8,
  '0.005*"nave" + 0.003*"arma" + 0.003*"capitán" + 0.003*"fuerza" + 0.003*"puer
ta" + 0.003*"salvar" + 0.003*"mover" + 0.002*"rápido" + 0.002*"regresar" + 0.00
2*"destruir"'),
 (9,
  '0.004*"capitán" + 0.003*"policía" + 0.003*"salvar" + 0.003*"os" + 0.003*"arm
a" + 0.002*"muerte" + 0.002*"barco" + 0.002*"wayne" + 0.002*"ud" + 0.002*"joder
"')]
```

100 tópicos, segmentando el corpus, con algunas stopwords

Usando las mismas stopwords que en los últimos documentos, segmentando y usando 100 tópicos se obtienen los siguientes resultados.

Haciendo una inspección manual y superficial de los tópicos, creo que son los tópicos que más me han gustado hasta ahora. Esto me ha sorprendido, ya que por alguna razón pensaba que iba suceder un sobreajuste donde cada tópico se refiriera a una película en específico. Pero quizás debido a que la segmentación es adecuada con los parámetros usados, es posible capturar relaciones entre palabras que no dependan de la separación específica entre películas individuales.

Lo que me gusta de estos tópicos es que no parece haber tantos nombres de personajes, las palabras más probables de los tópicos parecen referirse a conceptos generales, sí parece haber una coherencia semántica en los tópicos, y no hay casi redundancia (repetición) en los tópicos. Creo que todas estas características serían buenas para el propósito de clasificación por género.

El log de gensim deja ver que todos los documentos convergieron, incluso desde los primeros pases. Sin embargo, se alcanzó una perplejidad de 360,640.

In [14]:
```
stopwords = myutils.intersect_most_common(frecs_spa, n = 300)
corpus = myutils.texts2lists(train_txts_spa, stopwords = stopwords)
corpus = myutils.split_lists(corpus, chunk_size = 200, overlap_size = 100)
corpus = myutils.lists2bow(corpus, dictionary_spa)

%time model_t100_cs_stw300 = models.LdaModel(corpus, id2word=dictionary_spa, nu
m_topics=100, passes = 15, iterations = 400)
```

```
INFO : using symmetric alpha at 0.01
INFO : using symmetric eta at 0.01
INFO : using serial LDA version on this node
INFO : running online (multi-pass) LDA training, 100 topics, 15 passes over the
supplied corpus of 4865 documents, updating model once every 2000 documents, ev
aluating perplexity every 4865 documents, iterating 400x with a convergence thr
eshold of 0.001000
INFO : PROGRESS: pass 0, at document #2000/4865
DEBUG : performing inference on a chunk of 2000 documents
DEBUG : 1482/2000 documents converged within 400 iterations
DEBUG : updating topics
INFO : merging changes from 2000 documents into a model of 4865 documents
INFO : topic #83 (0.010): 0.006*"capitán" + 0.005*"nave" + 0.004*"barco" + 0.00
3*"cuidado" + 0.003*"sueño" + 0.003*"misión" + 0.003*"subir" + 0.003*"futuro" +
0.003*"arma" + 0.003*"muerte"
INFO : topic #82 (0.010): 0.004*"arma" + 0.004*"capitán" + 0.003*"salvar" + 0.0
03*"azúcar" + 0.003*"ud" + 0.003*"daphne" + 0.003*"wayne" + 0.003*"regresar" +
0.003*"hydra" + 0.002*"puerta"
INFO : topic #26 (0.010): 0.008*"arma" + 0.004*"fred" + 0.004*"disparar" + 0.00
3*"bastar" + 0.003*"victor" + 0.003*"soltar" + 0.003*"meter" + 0.003*"lula" + 0
.003*"mae" + 0.003*"joder"
INFO : topic #18 (0.010): 0.008*"código" + 0.006*"nave" + 0.004*"baxter" + 0.00
4*"tonto" + 0.004*"ascensor" + 0.003*"capitán" + 0.003*"wistrom" + 0.003*"conse
jo" + 0.003*"dobisch" + 0.003*"salto"
INFO : topic #77 (0.010): 0.011*"rey" + 0.006*"perro" + 0.005*"challa" + 0.004*
"arma" + 0.004*"woojin" + 0.004*"wakanda" + 0.004*"país" + 0.003*"prueba" + 0.0
03*"hierba" + 0.003*"corazón"
INFO : topic diff=90.836571, rho=1.000000
INFO : PROGRESS: pass 0, at document #4000/4865
DEBUG : performing inference on a chunk of 2000 documents
DEBUG : 2000/2000 documents converged within 400 iterations
DEBUG : updating topics
/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)
INFO : merging changes from 2000 documents into a model of 4865 documents
INFO : topic #44 (0.010): 0.019*"fiesta" + 0.010*"galleta" + 0.009*"banda" + 0.
009*"cinta" + 0.008*"vaso" + 0.006*"queso" + 0.006*"maní" + 0.006*"conseguir" +
0.006*"maggie" + 0.006*"divertir"
INFO : topic #6 (0.010): 0.017*"cumpleaños" + 0.013*"pingüino" + 0.010*"bernie"
+ 0.009*"feliz" + 0.007*"cariño" + 0.007*"hugo" + 0.007*"diablo" + 0.007*"reír"
+ 0.005*"wendy" + 0.005*"gracioso"
INFO : topic #33 (0.010): 0.007*"oliva" + 0.007*"genial" + 0.006*"baño" + 0.006
*"pulgar" + 0.006*"jake" + 0.005*"pervertir" + 0.005*"taylor" + 0.005*"fusión"
+ 0.005*"doctorado" + 0.005*"semana"
INFO : topic #36 (0.010): 0.008*"blair" + 0.008*"ardilla" + 0.004*"fuerza" + 0.
004*"juego" + 0.004*"cavar" + 0.004*"quemar" + 0.003*"paular" + 0.003*"ganar" +
0.003*"detonación" + 0.003*"lengua"
INFO : topic #66 (0.010): 0.045*"sueño" + 0.014*"cantar" + 0.013*"soñar" + 0.01
3*"escritor" + 0.011*"seth" + 0.010*"escribir" + 0.010*"librería" + 0.009*"asi"
+ 0.009*"rod" + 0.007*"café"
INFO : topic diff=inf, rho=0.707107
DEBUG : bound: at document #0
INFO : -19.639 per-word bound, 816638.6 perplexity estimate based on a held-out
corpus of 865 documents with 159800 words
INFO : PROGRESS: pass 0, at document #4865/4865
DEBUG : performing inference on a chunk of 865 documents
DEBUG : 865/865 documents converged within 400 iterations
DEBUG : updating topics
INFO : merging changes from 865 documents into a model of 4865 documents
INFO : topic #67 (0.010): 0.011*"latín" + 0.010*"tránsito" + 0.006*"doctor" + 0
.006*"scott" + 0.005*"cronómetro" + 0.005*"proteger" + 0.005*"ud" + 0.005*"cohe
```

```
CPU times: user 14min 5s, sys: 18min 2s, total: 32min 8s
Wall time: 4min 25s
```

In [21]:
```
print("Número de stopwords usadas:", len(stopwords))
print("Número de documentos tras segmentar:", len(corpus))

model_t100_cs_stw300.show_topics(num_topics = 30)
```

```
Número de stopwords usadas: 230
Número de documentos tras segmentar: 4865
```

Out[21]:
```
[(49,
  '0.082*"kevin" + 0.054*"gato" + 0.015*"plan" + 0.014*"ronan" + 0.011*"hallowe
en" + 0.009*"vegetariano" + 0.009*"bestia" + 0.008*"círculo" + 0.008*"bomba" +
0.007*"orbe"'),
 (70,
  '0.092*"nueva" + 0.089*"york" + 0.018*"albert" + 0.017*"escribir" + 0.014*"le
er" + 0.012*"director" + 0.010*"semana" + 0.010*"presidente" + 0.009*"fred" + 0
.008*"ciudad"'),
 (20,
  '0.054*"buddy" + 0.026*"baxter" + 0.021*"dreyfuss" + 0.019*"acompañar" + 0.01
8*"sonda" + 0.015*"kubelik" + 0.014*"sheldrake" + 0.014*"fuerza" + 0.013*"west"
+ 0.012*"alianza"'),
 (83,
  '0.016*"kaiju" + 0.012*"scotty" + 0.009*"atmósfera" + 0.009*"cerebro" + 0.009
*"puente" + 0.008*"categoría" + 0.008*"sobrevivir" + 0.008*"elegir" + 0.008*"si
stema" + 0.007*"hannibal"'),
 (17,
  '0.024*"walter" + 0.019*"número" + 0.013*"adam" + 0.013*"libro" + 0.010*"leer
" + 0.009*"chuck" + 0.009*"encantar" + 0.009*"quizás" + 0.008*"tom" + 0.008*"ro
jo"'),
 (34,
  '0.024*"luz" + 0.019*"josh" + 0.018*"billy" + 0.015*"puerta" + 0.012*"morder"
+ 0.010*"cuidado" + 0.009*"abajo" + 0.008*"perro" + 0.007*"lastimar" + 0.007*"s
ubir"'),
 (29,
  '0.029*"brujería" + 0.019*"palmer" + 0.017*"mei" + 0.017*"lee" + 0.012*"daesu
" + 0.010*"reloj" + 0.010*"fruto" + 0.009*"levantar" + 0.008*"orificio" + 0.008
*"black"'),
 (61,
  '0.033*"mía" + 0.024*"elizabeth" + 0.015*"funcionar" + 0.013*"vd" + 0.012*"ja
ne" + 0.009*"reconsiderar" + 0.009*"mail" + 0.008*"malekith" + 0.008*"aether" +
0.007*"orgullo"'),
 (57,
  '0.044*"barco" + 0.037*"blade" + 0.023*"capitán" + 0.015*"os" + 0.013*"pirata
" + 0.012*"vuestro" + 0.010*"enemigo" + 0.010*"turner" + 0.008*"plata" + 0.008*
"perla"'),
 (55,
  '0.095*"lucy" + 0.052*"andrew" + 0.031*"ba" + 0.030*"celoso" + 0.020*"gusto"
+ 0.018*"estatal" + 0.014*"aguardar" + 0.013*"monique" + 0.012*"próspero" + 0.0
10*"cantera"'),
 (72,
  '0.015*"llave" + 0.015*"asesino" + 0.014*"puerta" + 0.014*"miedo" + 0.011*"fa
milia" + 0.010*"habitación" + 0.010*"víctima" + 0.010*"prueba" + 0.009*"muerte"
+ 0.009*"policía"'),
 (18,
  '0.022*"toby" + 0.021*"groot" + 0.019*"heather" + 0.019*"quill" + 0.016*"veci
no" + 0.015*"código" + 0.014*"yondu" + 0.014*"grito" + 0.013*"wallace" + 0.013*
"ethan"'),
 (60,
  '0.041*"chris" + 0.031*"tony" + 0.010*"stark" + 0.009*"chistoso" + 0.008*"gan
ar" + 0.008*"traje" + 0.008*"mark" + 0.007*"howard" + 0.007*"banner" + 0.006*"t
ocar"'),
 (33,
  '0.079*"jake" + 0.027*"judy" + 0.014*"dígito" + 0.013*"candar" + 0.010*"mg" +
0.009*"religioso" + 0.009*"suavemente" + 0.009*"wg" + 0.008*"dinero" + 0.008*"c
omputador"'),
 (6,
  '0.021*"cumpleaños" + 0.021*"mills" + 0.017*"reír" + 0.015*"bernie" + 0.012*"
twinkies" + 0.010*"pingüino" + 0.008*"césar" + 0.008*"traje" + 0.008*"will" + 0
.008*"stuart"'),
 (85,
  '0.090*"danny" + 0.031*"you" + 0.013*"dolph" + 0.013*"repasar" + 0.013*"derec
ha" + 0.012*"ah" + 0.011*"carrera" + 0.011*"jackson" + 0.010*"conducir" + 0.010
```

Ploteando resultados

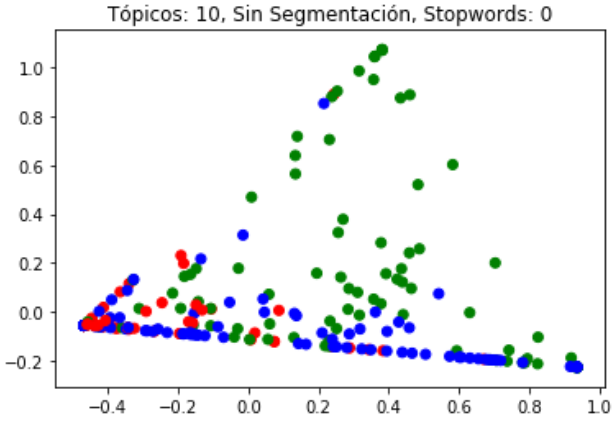
A continuación, por cada modelo previamente entrenado, obtengo representaciones vectoriales de todos los subtítulos y las proyecto a 2D usando PCA. Los ploteo como puntos, donde cada uno tiene un color dependiendo del género del que provienen (rojo es acción, verde es romance/comedia y azúl es horror).

In [120]:
```
def plot_results(txts, txts_colors, dictionary, frecs, model, n_topics, segment
ed, n_stw):
    if n_stw == 0:
        stopwords = []
    else:
        stopwords = myutils.intersect_most_common(frecs, n = n_stw)

    corpus = myutils.texts2lists(txts, stopwords = stopwords)
    corpus = myutils.lists2bow(corpus, dictionary)

    corpus = [model.get_document_topics(doc) for doc in corpus]
    numpy_matrix = gensim.matutils.corpus2dense(corpus, num_terms=n_topics).T
    pca = PCA(n_components=2)
    embeddings2D = pca.fit_transform(numpy_matrix)

    segmented_text = "Con Segmentación" if segmented else "Sin Segmentación"
    titulo = "Tópicos: {0}, {1}, Stopwords: {2}".format(n_topics, segmented_tex
t, n_stw)
    myutils.plot_data(embeddings2D, colores = txts_colors, titulo = titulo)
```

In [89]:
```
txts = train_txts_spa + test_txts_spa
colores = myutils.index2color(train_labels_spa + test_labels_spa, ['red', 'gree
n', 'blue'])
```

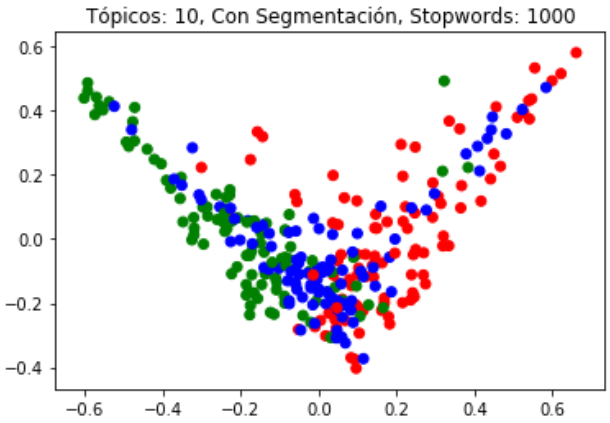
In [90]:
```
plot_results(txts, colores, dictionary_spa, frecs_spa, model = model_t10_ss_stw
0,
             n_topics = 10, segmented = False, n_stw = 0)
```

```
DEBUG : update_title_pos
DEBUG : update_title_pos
DEBUG : update_title_pos
DEBUG : update_title_pos
```



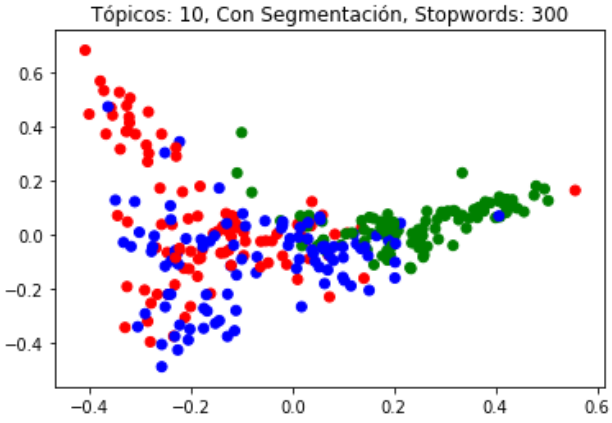
In [91]:
```
plot_results(txts, colores, dictionary_spa, frecs_spa, model = model_t10_cs_stw
1000,
             n_topics = 10, segmented = True, n_stw = 1000)
```

```
DEBUG : update_title_pos
DEBUG : update_title_pos
DEBUG : update_title_pos
DEBUG : update_title_pos
```



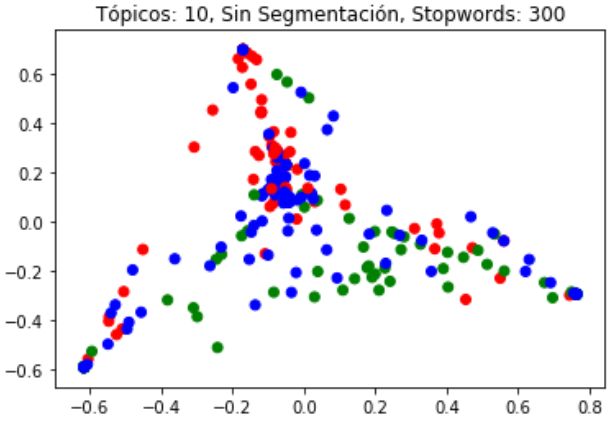
In [92]:
```
plot_results(txts, colores, dictionary_spa, frecs_spa, model = model_t10_cs_stw
300,
             n_topics = 10, segmented = True, n_stw = 300)
```

```
DEBUG : update_title_pos
DEBUG : update_title_pos
DEBUG : update_title_pos
DEBUG : update_title_pos
```



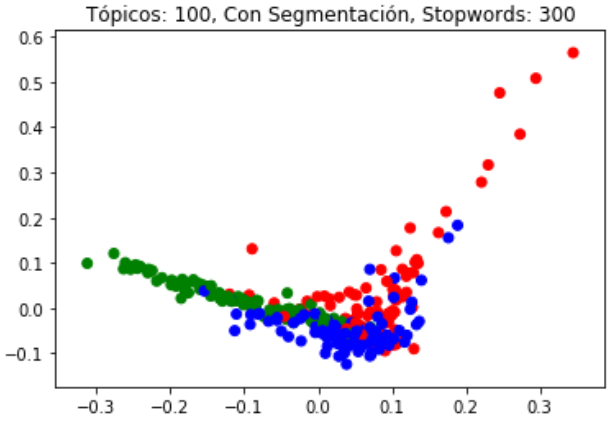
In [93]:
```
plot_results(txts, colores, dictionary_spa, frecs_spa, model = model_t10_ss_stw
300,
             n_topics = 10, segmented = False, n_stw = 300)
```

```
DEBUG : update_title_pos
DEBUG : update_title_pos
DEBUG : update_title_pos
DEBUG : update_title_pos
```



In [94]:
```
plot_results(txts, colores, dictionary_spa, frecs_spa, model = model_t100_cs_st
w300,
             n_topics = 100, segmented = True, n_stw = 300)
```

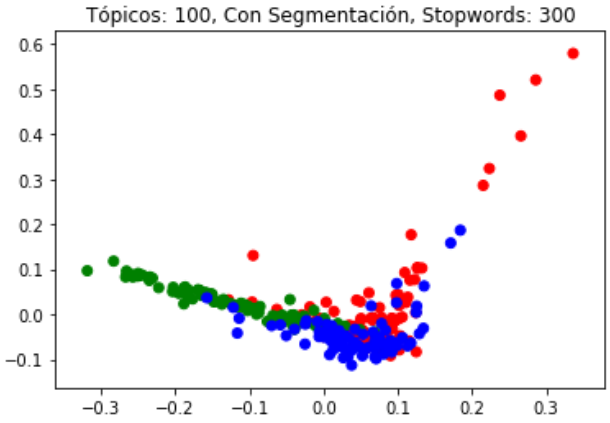
```
DEBUG : update_title_pos
DEBUG : update_title_pos
DEBUG : update_title_pos
DEBUG : update_title_pos
```



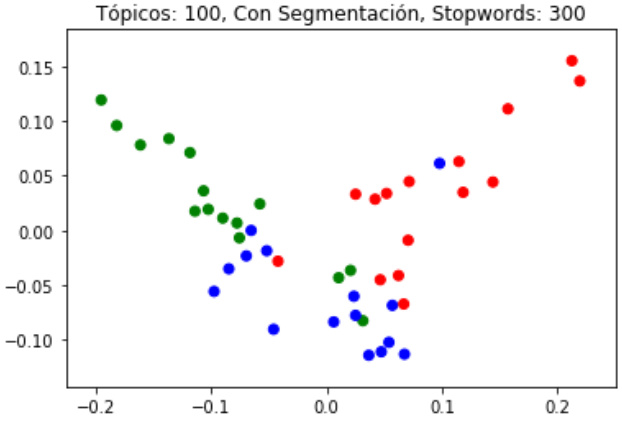
Y solamente por curiosidad, a continuación separo el último plot en los puntos provenientes del conjunto de entrenamiento del de conjunto de prueba. Se puede observar que siguen la misma forma, aunque los documentos del conjunto de prueba parecen tener mayor varianza con respecto a las tendencias mostrada en el conjunto de entrenamiento.

In [122]:
```
p = len(train_txts_spa)
plot_results(train_txts_spa, colores[:p], dictionary_spa, frecs_spa, model = mo
del_t100_cs_stw300,
             n_topics = 100, segmented = True, n_stw = 300)
plot_results(test_txts_spa, colores[p:], dictionary_spa, frecs_spa, model = mod
el_t100_cs_stw300,
             n_topics = 100, segmented = True, n_stw = 300)
```

```
DEBUG : update_title_pos
DEBUG : update_title_pos
DEBUG : update_title_pos
DEBUG : update_title_pos
```




```
DEBUG : update_title_pos
DEBUG : update_title_pos
DEBUG : update_title_pos
DEBUG : update_title_pos
```



Conclusiones y Comentarios

- Como ya he comentado, los resultados en los distintos experimentos tuvieron algunas sorpresas, aunque en general se esperaban tener los resultados obtenidos.
- Algo que aún no he comentado es sobre la variabilidad en la perplejidad dependiendo del tamaño del corpus utilizado y del número de tópicos a modelar. Por alguna razón no tenía muy presente que la perplejidad podría variar tanto dependiendo de estos factores, y que entonces uno no puede tomarla despreocupadamente como una medida de desempeño universal para comparar dos modelos, sino que es muy necesario considerar el contexto en el que se obtuvo.
- Por su parte, creo que los plots reflejan varios de los aspectos que ya comenté sobre los tópicos obtenidos. Aquellos tópicos donde sentía que no había mucha diferenciación, resulta que tienen un plot donde los documentos se mezclan bastante. En general, a mi gusto, el último plot (así como el último modelo) muestra una representación más separable de los documentos (y de los tópicos).
- Sin embargo, los plots también tuvieron sorpresas en que no me esperaba figuras tan geométricas, particularmente triángulares. Lo primero que pensé al verlas fue asociarlas con el Simplex de la distribución de dirichlet, pero mi instinto me hace dudar sobre ello (¿ o será que sí hay relación ?). En general no sé bien cómo interpretar las figuras triángulares y las rectas que se forman en las orillas y en el interior de los triángulos... considerando que PCA representa los datos en las direcciones de mayor varianza, el tener líneas rectas formando triángulos significa que los datos varían mucho, en la misma proporción, en distintas direcciones (¿?) no lo sé.
- Finalmente solo queda decir que quizás si tuviera más oportunidad de experimentar pudiera llegar a ver parámetros más óptimos. Sin embargo en el siguiente cuaderno me dispondré justamente a automatizar el proceso de selección de parámetros. Por lo pronto, me parece que si hay algo seguro es que prefiero usar stopwords, y segmentar los documentos del corpus en documentos más pequeños.

```
In [ ]:
```