Dynamic Topic Modeling in microblogging and literature

Giussani Riccardo

Università degli Studi di Milano, `riccardo.giussani@studenti.unimi.it`

1 Introduction

Topic modeling is a statistical model typically used to describe corpora of textual documents identifying abstract "topics". It has many open-source implementations like the one provided by the gensim library. This model however lacks the temporal dimension: it's based on the assumption that documents are unordered and so it doesn't take into account the fact that in a collection of humanly generated documents the same underlying topic may change over time. This work is based on the idea in Blei and Lafferty (2006) of a dynamic topic modeling, capable of capturing the evolution of topics over time.

2 Research Question and Methodology

2.1 Problem

An implementation of the algorithm presented in Blei and Lafferty (2006) is available in gensim[1]. This work wants to explore the possibility to effectively apply the algorithm to corpora of humanly generated documents and study the capability to identify interesting topics and their evolution.
It is necessary to implement a python pipeline that fits the raw corpus into a format compatible with the interface of gensim models.
The problem of the choice of the number of topics (which is an hyper-parameter to the model) is also taken into account.

2.2 Approach

The corpus is given as input to a custom spacy pipeline[2] that for each document, after the tokenization, lemmatizes the content and then calls the gensim bag-of-words function of the Dictionary object. The gensim dictionary is also created. To each document, a coarse-grained timestamp is assigned (the semantics depends on the original corpus). The output of the pipeline fits the characteristics required by the gensim model interface, so the LdaSeqModel function can be called many times, each time with a different number of topics (in this work three: 5, 10 and 20).

[1] https://radimrehurek.com/gensim/models/ldaseqmodel.html
[2] https://spacy.io/usage/processing-pipelines

3 Experimental Results

3.1 Datasets

I tried to use the model on two datasets of very different nature: one comes from the world of twitter microblogging, specifically the corpus of Donald Trump's tweets; the second is the novel Gone with the Wind. Each tweet of Trump is a document with timestamp the year of publication; each paragraph of Gone with the Wind is a document with timestamp the part (the book is divided into 5 parts). For Trump's tweets it is sufficient to extract the data from the Json downloadable from a dedicated website[3]. Gone with the Wind text is provided by Project Gutemberg Australia[4] and requires some basic html parsing, exploiting the BeautifulSoup library, to fit the corpus to the pipeline.

3.2 Evaluations

In order to choose the proper number of topics, the coherence score of u_mass is taken into consideration for each timestamp. Each topic is then humanly interpreted; the most interesting ones are taken into account for a further analysis of the evolution of the word probability of key words in the topic over time.

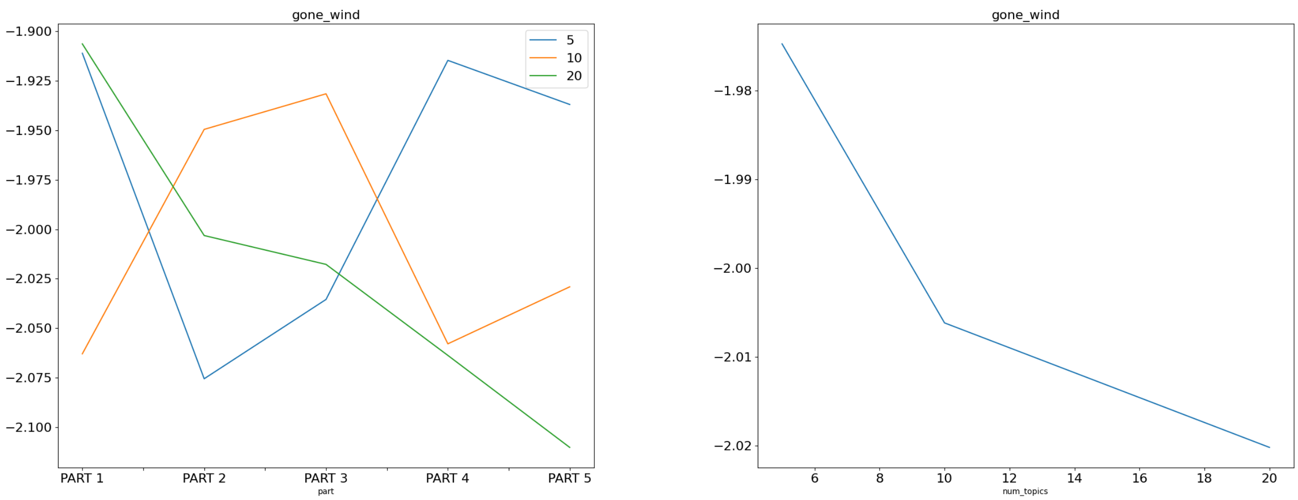
3.3 Results

Coherence

Gone with the Wind Observing coherence over time it can be said that there's no advantage in increasing the number of topics, as coherence always falls in the same restricted interval. So, the model trained on 5 topics is chosen.

Trump's tweets Coherence is much lower for number of topics equal to 20 than others in the first years. Considering the mean coherence, the gain obtained doubling the number of topics from 10 to 20 follows almost the same trend of doubling from 5 to 10. Therefore, 20 topics are selected.

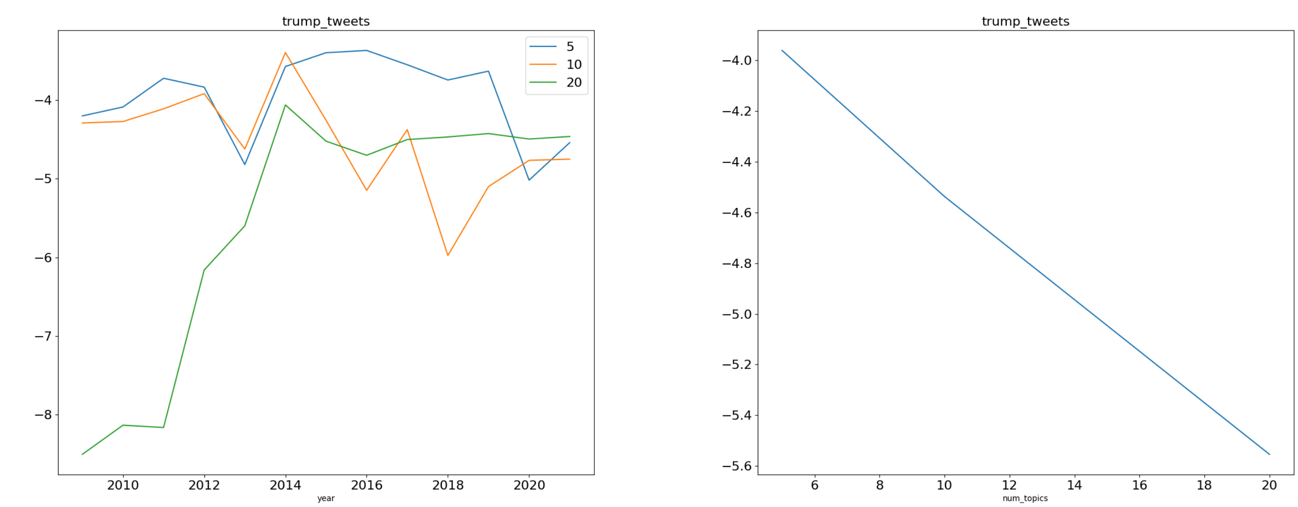
[3] https://www.thetrumparchive.com
[4] https://gutenberg.net.au/ebooks02/0200161h.html



(a) Coherence by part

(b) Mean coherence for each number of topics

Fig. 1: Coherence visualization for Gone with the Wind



(a) Coherence by year

(b) Mean coherence for each number of topics

Fig. 2: Coherence visualization for Trump's tweets corpus

Topic Evolution Topics are visualized as many lists of words (one for each timestamp). Words are ordered by their word probability for that topic at that timestamp. I save for each topic (in each timestamp) the first 30 words. For all the others, a word probability of 0 is assigned.

Gone with the Wind: thoughts and perceptions This topic can be interpreted as the one concerning the protagonist's thoughts and perceptions for the preponderance of the words **know**, **think** and **say**.
Most of the words have no meaning taken out of context but nevertheless it's

interesting to notice for this topic the word probability of **money**, **love** and **war** that are crucial in the story.
The war takes the first three parts, while parts four and five concern the aftermath. It's justified the increasing trend of money as greed and poverty are key elements of parts three and four. Loves comes back in the fifth part, the most sentimental of all parts.

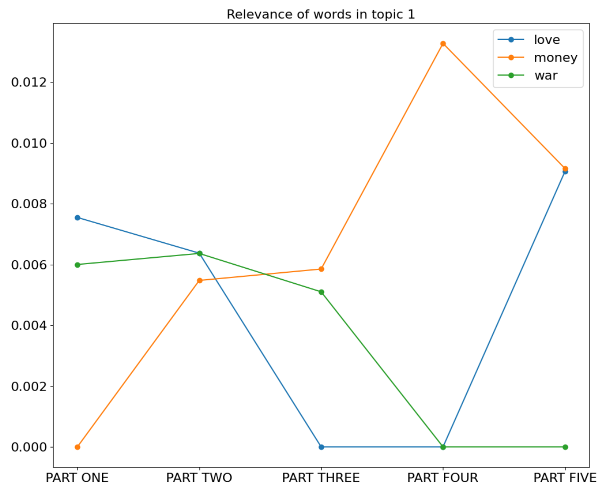


Fig. 3: word probability of the words love, money and war.

Gone with the Wind: characters This topic contains the names of many important characters in the story, so I find it interesting to evaluate if the word probability of the name of a character has a correlation with their relevance in the story (the protagonist Scarlett has of course the highest word probability).

The model captures the decline of Ashley in favour of Rhett, both in the plot and

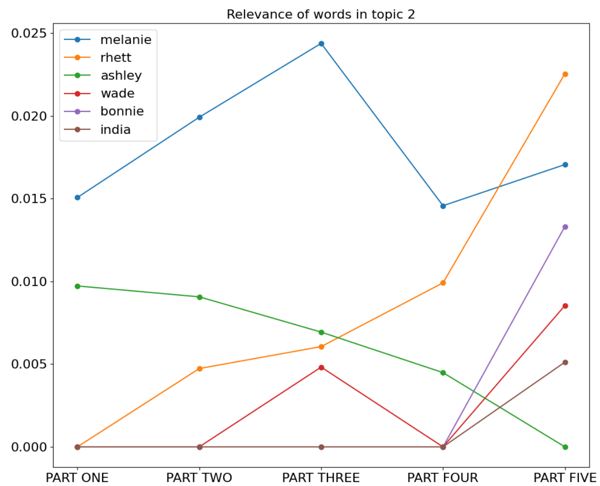


Fig. 4: word probability of some characters

in Scarlett's affections. The model is also able to detect as a character Bonnie even though she was not present in previous parts.

Trump's tweets: Bad things This topic is dominated by the word **Obama** and negative words such as **bad**, **disaster** and **weak**. This topic shows Trump's main preoccupations over time such as ebola in 2014, Libya before 2013 and administration (of the state) during it's years of presidency. It's also evident an ever-lasting worry about Iran and a moderate one for nuclear energy.
It must be noticed that before 2014 there were no tweets about ebola, but the probability is still high because the underlying model applies continuous modifications to the probabilities. This is a case of oversmoothing. The existence of this phenomenon was also pointed out by Blei in a Google Talk[5]

[5] https://www.youtube.com/watch?v=7BMsuyBPx90&t=2672s

2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021
obama	obama	obama	obama	obama	obama	obama	medium	fake	fake	fake	fake	fake
iran	iran	iran	attack	attack	ebola	medium	obama	medium	medium	medium	medium	medium
attack	attack	attack	iran	iran	iran	iran	report	obama	report	administration	report	voter
policy	policy	policy	policy	wrong	say	say	say	report	obama	report	administration	report
release	release	release	release	disaster	attack	report	bad	administration	administration	obama	obama	administration
economic	economic	economic	report	bad	medium	bad	attack	election	bad	say	voter	obama
nuclear	nuclear	nuclear	foreign	say	bad	attack	weak	say	say	try	americans	month
libya	libya	libya	medium	medium	west	weak	policy	bad	phony	iran	iran	americans
americans	americans	americans	bad	report	report	write	iran	phony	election	witness	month	iran
month	month	foreign	month	release	disaster	result	fake	weak	month	bad	say	election
foreign	foreign	disaster	disaster	mistake	result	policy	virginia	iran	iran	policy	election	say
disaster	disaster	bad	economic	month	write	americans	write	month	weak	month	policy	policy
bad	bad	report	say	public	release	disaster	phony	policy	policy	election	try	try
report	report	month	americans	policy	weak	month	voter	virginia	result	phony	witness	lamestream
medium	medium	medium	credit	write	public	virginia	result	attack	write	americans	result	witness

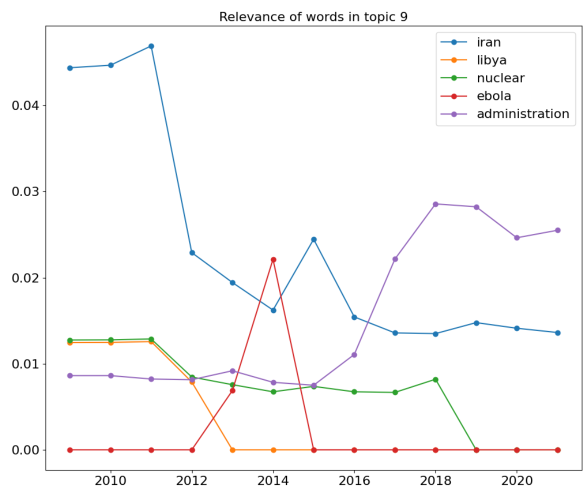


Fig. 5: word probability of negative things, according to Trump

Trump's tweets: Opponents This topic collects Trump's political opponents. In each timestamp we can see a peak in word probability for the specific adversary of that time, may that be an actual person (like Clinton, Cruz, Obama or Bush) or a dangerous event (as impeachment or collusion with Russia).

2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021
high	cont	cont	obama	thing	thing	bush	clinton	clinton	collusion	impeachment	impeachment	happen
cont	high	price	price	money	happen	jeb	cruz	campaign	happen	happen	happen	impeachment
price	price	high	cont	bad	bad	thing	campaign	thing	campaign	collusion	thing	thing
money	money	money	high	happen	bush	clinton	bad	happen	clinton	thing	low	2020
obama	obama	obama	thing	obama	energy	bad	say	high	thing	low	2020	low
campaign	campaign	campaign	gas	lose	money	campaign	happen	low	low	high	john	election
thing	thing	thing	money	high	lose	happen	thing	bad	high	2020	election	john
gas	gas	gas	bad	lot	high	lose	jeb	john	john	campaign	campaign	campaign
unemployment	unemployment	unemployment	campaign	price	obama	money	lose	collusion	election	john	high	price
bad	bad	bad	unemployment	didn't	campaign	say	bush	lose	bad	clinton	price	high
low	low	low	happen	john	didn't	cruz	john	say	level	time	catch	catch
opec	opec	opec	energy	energy	lot	john	low	election	unemployment	election	energy	energy
energy	energy	energy	mitt	give	say	high	high	level	comey	bad	call	call
waste	waste	waste	rise	bush	john	low	money	cruz	time	price	bad	bad
rise	rise	rise	lose	campaign	low	didn't	didn't	time	price	unemployment	promise	promise

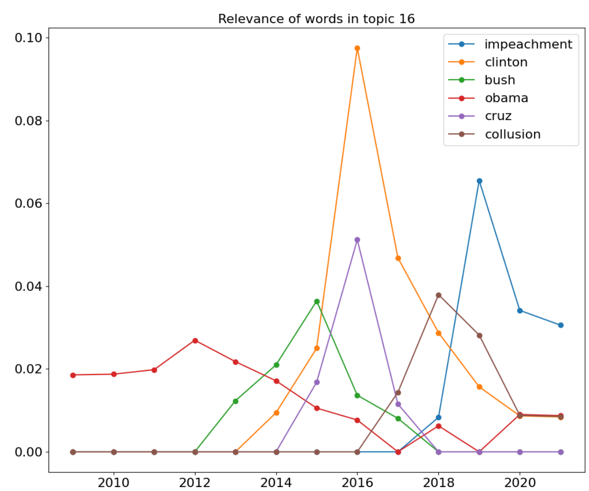


Fig. 6: word probability of some political opponents

4 Concluding Remarks

Dynamic topic modeling provides an interesting and useful tool to extend the classical LDA model on the temporal dimension. A possibility for improvement can be a more specific evaluation of coherence to decide the optimal number of topics. A drawback of this model is that it is not capable to capture the birth and death of new topics through time. Overall, this simple experiment provides a way to interpret results and points out that, for some topics, a humanly understandable interpretation is possible.

Bibliography

Blei, D. M. and Lafferty, J. D. (2006). Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120.