FINDING THEMES IN INDONESIAN TWITTER WITH LSA AND LDA

I. Definition

Project Overview

On 9 July 2014, millions of people across the world's fourth most-populous country went to the polls in a presidential contest widely portrayed as a choice between Indonesia's authoritarian past and a more-democratic and populist future. After 13 days, Indonesia's Election Commission declared Jakarta Governor Joko Widodo, lovingly known by his base as Jokowi, the winner against former general Prabowo Subianto in a 53% to 47% split.

“pak presiden jokowi yaa”

“Mr. President Jokowi, yesss!"

- TWITTER USER

In 2014 as today, Indonesia has consistently ranked among the top 5 countries represented on Twitter, offering a wealth of potential natural language data for analysis. In this project, I'll explore latent topics within a twitter corpus to better understand the Indonesian public's (or rather the young, urban public's) views surrounding the election, as well as the volume of tweets produced by topic.

Problem Statement

The biggest problem with analyzing text data is that it's inherently unstructured. Twitter users misspell words, contract words, repeat letters for emphasis, and all of this is just on the originator's end. The tweets also end up with widely variant lengths and different grammatical structures, making language especially complicated for doing comparative mathematical operations (data science).

For problems like this, so-called bag-of-words models, where a computer makes a “list” of every possible word and then counts the number of times each word appears in a given tweet, provide an easy way to standardize the data into a set format for representing each tweet. Bags-of-words also lend themselves to two widely-used algorithms for discovering topics within tweets and then grouping tweets by those topics: Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA). By running these algorithms against our tweets, we can find latent themes among tweeters, examining the main words in each theme to get a sense for which model better creates understandable groupings.

Latent Semantic Analysis, or LSA, is an algorithm that plots the tweets in multi-dimensions based on the words in each tweet, then tries to draw the most sensible lines to clearly separate the tweets into similar groupings. LDA, by contrast, does a complicated matrix multiplication on the words in the tweets to separate the tweets into topics based on words commonly found in amongst other similar words. LSA is less computationally intensive, but often results in lower-quality topic groupings

II. Analysis

Data Exploration

The corpus of tweets I use, collected during the 2014 Indonesian elections by GitHub user Ali Akbar S., was originally intended for sentiment analysis, and consists of rows of tweets in Indonesian paired with a 1, 0, or -1 depending on the perceived sentiment of the text. It's also limited to only 1,846 tweets, meaning the tweets' short lengths and limited number could adversely affect the algorithms' ability to find themes. Since we're not examining sentiment in this analysis, I chose to drop the final column, keeping only a vector of individual tweets.

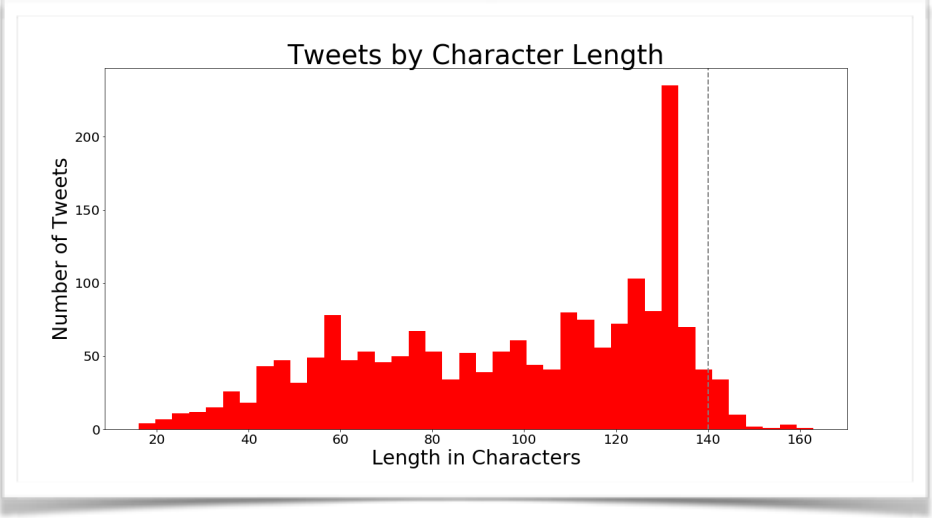
First 5 Observations in Data

	isi_tweet	sentimen
0	tidak setuju jokowi jadi cawapres capres jokow...	1
1	capres jokowi wacapres abraham samad gubernur ...	1
2	capres prabowo dan cawapres jokowi dan gubdki ...	1
3	jadi skenarionya gini 2014 biar prabowo jadi p...	1
4	sby mantan tni dan calon presiden prabowo subi...	1

1846 tweets in dataset

A "bag-of-words" model creates a dictionary object counting the number of times a word appears in a sentence. For example, "Ada asap ada api" would be stored as {ada: 2, asap: 1, api: 1}

"Vectors" are really just charts with only one column. Storing my tweet data this way allows me to access individual tweets as separate data points.



I was surprised to find some Tweets over 140 characters, the limit until 2017. But examining the first 5 of these Tweets didn't reveal anything unusual. Instead, I saw some regular problems with Twitter data, like Tweeters forgetting to type spaces between words, or users concatenating (sticking together) numbers and letters.

First 5 Tweets over 140 Characters

```
----------
Tweet 5:
jadi skenarionya gini 2014 biar prabowo jadi presiden jokowi tetepgubernur kalau jakarta berhasil tidak usah nunggu 2
019 buat gantiin prabowo
----------
Tweet 4:
saat jadi meneg bumn dahlan iskan bisa sedikit ambil bagian dalam kep strategis pln saat jadi presiden jokowi bisa ju
ga melakukan hal yang sama untuk jakarta
----------
Tweet 3:
capres arb banyak beriklan mengeluarkan kata2 yang membuai rakyat kemudian membuat statement elektabilitasnya naik ja
ngan hnya segitunya bung
----------
Tweet 2:
sebel lihat capres aburizal bakrie tidak tahu kenapa dia ya nyalon atas dasar apa juga tidak ngrti mau ngurusin indon
esia ngurusin lapindo saja tidak kelar2
----------
Tweet 1:
eneggg sangat saya lihat iklan capres arb di iklan ngomong a sok bijak bnr sebener a dia punya kaca tidak sii di ruma
hnya lapindoajablomkelar
```

Algorithms & Techniques

English has long dominated the tech field, and most existing natural language processing (NLP) packages are created to augment English language data for analysis. Since the twitter data is in Indonesian, we have to get a little creative with our preprocessing.

To deal with punctuation, numbers, and numbers concatenated with words, I imported regular expressions, or RegEx,to drop everything but alphabetical characters. I imported an NLP package, Gensim, to easily build models, and NumPy to set a random seed for reproducible results. A package called Sastrawi converts Indonesian words to their root forms without prefixes or suffixes and lowercases them, and the CSV package allowed me to read in a separate list from Kaggle to remove the most common Indonesian words from the corpus (the first time I ran LDA, several topics included words like dan ("and") or di ("at")). I also added an index column to use later to identify specific tweets.

RegEx is a way to manually tell the computer "I'm interested in these particular characters, or sequence of characters, and not others." It allows us to do things like remove anything that's not an alphabetical character, such as numerals and punctuation.

To analyze text, we usually have to convert words to their stems (so words like "ran", "run", and "runs" all count as the same concept), lowercase them (so "Run" and "run" are identical), and remove overly common words, called "stop words" (so we don't end up with a list of words like "and and "the".

After parsing the tweets into a list of individual preprocessed words and using Gensim to create a bag-of-words, I ran my LSA model and two LDA models on the data to see which model produced the clearest and most-coherent topics. LSA is a simpler algorithm and is less computationally expensive to run, but LDA typically results in more sensible topics. LDA can be run either on the bag of words immediately, or after performing an algorithm called term frequency/inverse document frequency (Tf-idf) which amplified the differences between tweets. Post-Tf-idf LDA is an industry standard, but can result in proper nouns being overrepresented, which might result in less-clear categorizations, depending on the corpus. Testing all three, first to discern general topics over the corpus, then to categorize a chosen tweet, then finally to categorize a fake tweet I will compose, should give me a good idea of which method works best in this instance.

III. Methodology

Data Preprocessing

Sastrawi's stemmer worked impressively well against some highly-conjugated example sentences I entered in Indonesian, and had the additional benefit of lowercasing my words. The stop words list included various conjugations of common words, but since the list was not exhaustive, I removed stop words only after stemming and removing non-alphabetical characters.

Stemming Tests

```
-----
Original: Mereka meniru-nirukannya

Stemmed: mereka tiru
-----
Original: Saya pembantu

Stemmed: saya bantu
-----
Original: Dia dipanggil oleh wanita tercantik

Stemmed: dia panggil oleh wanita cantik
```

To run my models in Gensim, I needed to create a dictionary of the word in my corpus, and chose to remove words that appeared less than 40 times or in more than 30% of tweets. This removed words that could muddle underlying categories by being too common or rare. I also ran my Tf-idf algorithm at this point for later use in my model.

Implementation & Refinement

I experimented with a range of numbers of topics, and landed on 5 as a reasonable number on which to run my LSA, BOW LDA and Tf-idf LDA models. I also adjusted my parameters for filtering my Gensim dictionary to the final parameters of 40 <= total number of times a word is observed in the corpus <= 30% of the corpus from my original parameters of 15 <= total observations in the corpus <= 50% of the corpus, which on first glance seemed to give clearer topics.

“dalam angan berkata betapa hebatnya indonesia jika punya presiden prabowo dengan wakilnya jokowi.”

“It’s said in wishes how great Indonesia would be with a President Prabowo and Vice President Jokowi.”

- TWITTER USER

LSA Topics

```
Topic: 0
Words: 0.421*"megawati" + 0.395*"wakil" + 0.373*"mantan" + 0.354*"jusuf" + 0.354*"kalla" + 0.300*"jk" + 0.212*"prabow
o" + 0.120*"nyata" + 0.119*"gubernur" + 0.110*"tokoh"


Topic: 1
Words: -0.897*"megawati" + 0.199*"jusuf" + 0.199*"kalla" + 0.184*"wakil" + 0.166*"jk" + 0.119*"prabowo" + 0.107*"mant
an" + 0.070*"nyata" + 0.057*"bincang" + 0.057*"tokoh"


Topic: 2
Words: -0.904*"prabowo" + 0.151*"jusuf" + 0.149*"kalla" + 0.146*"mantan" + -0.145*"wiranto" + -0.114*"arb" + 0.107*"w
akil" + -0.092*"indonesia" + -0.086*"cawapres" + -0.077*"calon"


Topic: 3
Words: -0.850*"dahlan" + -0.407*"iskan" + 0.155*"prabowo" + -0.133*"konvensi" + -0.077*"pilih" + -0.076*"maju" + -0.0
74*"arb" + 0.070*"mantan" + 0.070*"jusuf" + 0.068*"kalla"


Topic: 4
Words: 0.561*"arb" + 0.411*"hatta" + 0.340*"wiranto" + -0.240*"prabowo" + -0.187*"dahlan" + 0.179*"cawapres" + 0.168*
"ketua" + 0.162*"pan" + 0.162*"buka" + 0.150*"evaluasi"
```

LDA on BOW Topics

```
Topic: 0
Words: 0.123*"kalla" + 0.118*"jusuf" + 0.116*"wakil" + 0.110*"mantan" + 0.041*"nyata" + 0.040*"jk" + 0.040*"tokoh" +
0.035*"gubernur" + 0.035*"sosok" + 0.035*"populer"


Topic: 1
Words: 0.153*"dahlan" + 0.132*"prabowo" + 0.088*"wiranto" + 0.083*"iskan" + 0.057*"indonesia" + 0.055*"ya" + 0.047*"c
awapres" + 0.037*"ahok" + 0.033*"gubernur" + 0.025*"dukung"


Topic: 2
Words: 0.159*"jk" + 0.142*"arb" + 0.080*"wakil" + 0.065*"prabowo" + 0.053*"pilih" + 0.051*"calon" + 0.047*"risma" + 0
.034*"mantan" + 0.031*"mahfud" + 0.027*"wapres"


Topic: 3
Words: 0.083*"konvensi" + 0.072*"pdip" + 0.068*"pilih" + 0.064*"partai" + 0.060*"golkar" + 0.055*"mahfud" + 0.049*"su
rvey" + 0.045*"menang" + 0.041*"rakyat" + 0.041*"md"


Topic: 4
Words: 0.150*"hatta" + 0.119*"megawati" + 0.072*"ketua" + 0.066*"buka" + 0.065*"pan" + 0.060*"pencapresan" + 0.060*"e
valuasi" + 0.059*"radjasa" + 0.047*"cawapres" + 0.039*"calon"
```

LDA on Tf-idf Topics

```
Topic: 0
Words: 0.162*"dahlan" + 0.080*"pdip" + 0.076*"pilih" + 0.071*"iskan" + 0.063*"maju" + 0.063*"dukung" + 0.057*"mega" +
0.056*"gubernur" + 0.046*"jk" + 0.043*"pasang"


Topic: 1
Words: 0.125*"hatta" + 0.124*"megawati" + 0.083*"partai" + 0.071*"jk" + 0.053*"ya" + 0.050*"ketua" + 0.047*"cawapres"
+ 0.046*"buka" + 0.043*"pan" + 0.041*"evaluasi"


Topic: 2
Words: 0.117*"jakarta" + 0.087*"dahlan" + 0.085*"orang" + 0.069*"konvensi" + 0.065*"wapres" + 0.051*"iskan" + 0.051*"
megawati" + 0.036*"sby" + 0.031*"md" + 0.030*"mahfud"


Topic: 3
Words: 0.182*"arb" + 0.117*"calon" + 0.086*"iklan" + 0.067*"survey" + 0.046*"risma" + 0.041*"sby" + 0.039*"prabowo" +
0.038*"cawapres" + 0.038*"indonesia" + 0.035*"jk"


Topic: 4
Words: 0.122*"prabowo" + 0.079*"wakil" + 0.071*"mantan" + 0.067*"kalla" + 0.065*"wiranto" + 0.065*"jusuf" + 0.051*"me
gawati" + 0.043*"rakyat" + 0.032*"jk" + 0.031*"indonesia"
```

IV. Results

Model Evaluation & Validation

Computers tend to begin counts with zero rather than one, as a quirk or the way binary systems work. Although my models have five topics, they're numbered 0 through 4.

The topics derived from LSA seemed pretty unclear, with a lot of overlapping words. Topic zero seemed roughly to be about Megawati Sukarnoputri's support for Presidential Candidate and Jakarta Governor Jokowi and his running mate, Jusuf Kalla (also known as JK) against Former General Prabowo Subianto. Topic one was almost identical, substituting the word for "talks" with the word for "governor." Topic three was interesting, discussing Presidential Candidate Prabowo and rival Vice Presidential Candidate Kalla, as well as two dropouts from the presidential race, Aburizal Bakrie and Hanura Wiranto, as well as the words for candidate and Vice Presidential candidate. Perhaps these tweets were regarding a "dream team" of Prabowo as President and Kalla as Vice President. Topic three once again discussed Prabowo and Kalla, with mention of a convention, Aburizal Bakrie, and the words "election", "enter", and "forward". The final topic once again included Golkar Chairman Aburizal Bakrie, Hatta, Wiranto, Prabowo, Dahlan, Prabowo's PAN party, and the words "vice president", "open", and "evaluation". Overall, these groupings don't appear particularly useful or enlightening.

LDA with a bag of words seemed better, with topics such as "Kalla", "Leader(ship)", "Governor (Jokowi)", "Real", and "Popular" (Topic zero: Jokowi), "Dahlan", "Prabowo", "Wiranto", "Fill in", "Yes", "Support", "Ahok", and "Governor" (Topic one: Famous endorsements for Jokowi against Prabowo), "Kalla", "Aburizal Bakrie", "Prabowo" "Tri Rismaharini", and "Mahfud MD" (Topic 2: Famous Endorsements of Prabowo against the Jokowi-Kalla ticket, or in Rismaharini's case, a fake endorsement), "convention", "PDI-P party", "Golkar party", "Mahfud MD", "survey", "People's/public", "win" (Topic three: polling and party comparison), and "Megawati", "PAN", "Presidential candidacy", and "Hatta Radjasa" (Topic four: Prabowo). The topics also had a nice symmetry, with one topic per major candidate, one topic of famous supporters of each candidacy, and a topic on polling and party comparison.

"jadi kapan kopdar capres prabowo nya hahaha"

"So when will you join (in supporting) Presidential Candidate Prabowo? hahaha!"

- TWITTER USER

Tf-idf is an algorithm that magnifies the weight of words that are less common among all tweets and shrinks the weight of words that are more common thought the corpus. The result is that the things that make each tweet most unique are amplified, which can help clustering in some cases, but may also result in overrepresentation of proper nouns that are too unique and prevent grouping with other tweets.

LDA with Tf-idf gave interesting topics such as "Dahlan", "PDI-P", "Forward", "Support", "Mega(wati)", "Governor", "Kalla" (Topic zero: Jokowi), "Hatta", "Megawati", "Party", "Kalla", "Vice President", "PAN" and "Evaluation" (Topic one: Vice Presidents), "Jakarta", "Dahlan", "Convention", "Vice President", "Megawati", "Yudhoyono", "Mahfud MD" (Topic two: national political players), "Aburizal Bakrie", "Candidate", "Ad", "Survey", "Tri Rismaharini", "Yudhoyono", "Prabowo", "Vice Presidential Candidate", and "Kalla" (Topic three: unclear, possibly political advertisements and polling), and "Prabowo", "Vice", "Kalla", "Wiranto", "Megawati", "People", and "Indonesia" (Topic five: unclear, more famous political players). However, the topics seemed less clearly-defined than LDA with the bag of words.

Justification & Spot Checking

For this particular corpus, and perhaps this is related to the small total number of tweets and the limited words in each tweet, LDA performed better when not weighted with Tf-idf, so that is the model and topic grouping I chose to proceed with.

I ran the LDA with BOW algorithm on a tweet about Jokowi—"kalau jadi presiden jokowi tetep jadi gubernur jakarta tidak"—which matched 73% with the first topic: Jokowi. A fake Indonesian tweet I wrote supporting Jokowi, PDI-P, and Kalla—"Saya mendukung JK dan Kalla! PDI-P selamanya!"—also scored as an 80% match with the Jokowi topic.

Model on Sample Tweet

```
Score: 0.729947566986084
Topic: 0.150*"hatta" + 0.119*"megawati" + 0.072*"ketua" + 0.066*"buka" + 0.065*"pan" + 0.060*"pencapresan" + 0.060*"e
valuasi" + 0.059*"radjasa" + 0.047*"cawapres" + 0.039*"calon"

Score: 0.06830424070358276
Topic: 0.123*"kalla" + 0.118*"jusuf" + 0.116*"wakil" + 0.110*"mantan" + 0.041*"nyata" + 0.040*"jk" + 0.040*"tokoh" +
0.035*"gubernur" + 0.035*"sosok" + 0.035*"populer"

Score: 0.06825393438339233
Topic: 0.153*"dahlan" + 0.132*"prabowo" + 0.088*"wiranto" + 0.083*"iskan" + 0.057*"indonesia" + 0.055*"ya" + 0.047*"c
awapres" + 0.037*"ahok" + 0.033*"gubernur" + 0.025*"dukung"

Score: 0.06682510673999786
Topic: 0.159*"jk" + 0.142*"arb" + 0.080*"wakil" + 0.065*"prabowo" + 0.053*"pilih" + 0.051*"calon" + 0.047*"risma" + 0
.034*"mantan" + 0.031*"mahfud" + 0.027*"wapres"

Score: 0.06666915863752365
Topic: 0.083*"konvensi" + 0.072*"pdip" + 0.068*"pilih" + 0.064*"partai" + 0.060*"golkar" + 0.055*"mahfud" + 0.049*"su
rvey" + 0.045*"menang" + 0.041*"rakyat" + 0.041*"md"
```

Model on Synthetic Tweet

```
LDA with BOW Model:

Score: 0.7960905432701111
Topic: 0.123*"kalla" + 0.118*"jusuf" + 0.116*"wakil" + 0.110*"mantan" + 0.041*"nyata" + 0.040*"jk" + 0.040*"tokoh" +
0.035*"gubernur" + 0.035*"sosok" + 0.035*"populer"

Score: 0.05283946916460991
Topic: 0.159*"jk" + 0.142*"arb" + 0.080*"wakil" + 0.065*"prabowo" + 0.053*"pilih" + 0.051*"calon" + 0.047*"risma" + 0
.034*"mantan" + 0.031*"mahfud" + 0.027*"wapres"

Score: 0.0505364686625068665
Topic: 0.153*"dahlan" + 0.132*"prabowo" + 0.088*"wiranto" + 0.083*"iskan" + 0.057*"indonesia" + 0.055*"ya" + 0.047*"c
awapres" + 0.037*"ahok" + 0.033*"gubernur" + 0.025*"dukung"

Score: 0.05044544115662575
Topic: 0.083*"konvensi" + 0.072*"pdip" + 0.068*"pilih" + 0.064*"partai" + 0.060*"golkar" + 0.055*"mahfud" + 0.049*"su
rvey" + 0.045*"menang" + 0.041*"rakyat" + 0.041*"md"

Score: 0.05008801072835922
Topic: 0.150*"hatta" + 0.119*"megawati" + 0.072*"ketua" + 0.066*"buka" + 0.065*"pan" + 0.060*"pencapresan" + 0.060*"e
valuasi" + 0.059*"radjasa" + 0.047*"cawapres" + 0.039*"calon"
```

V. Conclusion

Reflections

Despite my small corpus and limited vocabulary in each tweet, LSA and LDA helped me quickly suss out topics within the dataset and see sensible topic clusterings. A simple LDA with a bag of words gave me the most sensible clusterings of the two presidential candidates, famous endorsements or supporters of each ticket, and public polling. The model also seemed to perform well on a sample tweet and synthetically-created tweet.

Improvement

To really shine, these models should be applied to a much larger corpus. The corpus should also be cleaned of characters repeated three or more times as well as English words, which I observed at times within the corpus. With these conditions met, the models would be more robust and better represent the Indonesian twitter population during the 2014 Indonesian election. A more complete corpus could also enable me to map the sizes of each topic cluster more meaningfully to answer questions such as whether Jokowi or Prabowo seemed to generate more tweets, or how many tweets seem to have been about endorsements or supporters rather than focusing on the actual candidates. There's lots of room to expand this basic code framework and hopefully improve insights around Indonesian political twitter for the 2019 election.