Business & Conflict Barometer NLP Qualitative Analysis

Team DataWorthy

Senait Dafa, Nicholas Ketchum, Olivia Sterling, Mengqi Wu

Winter 2022

Introduction:

Our work will target two key areas to improve the accuracy and efficiency of the conflict barometer through two distinct models: a Fuzzy Matching Models and an Unsupervised Topic Model. The Fuzzy Matching Model aims to address naming inconsistencies in our datasets, while the Unsupervised Topic Model identifies topics within aggregated textual data.

Fuzzy Matching Model

The first project is to create a fuzzy matching model to address inconsistent naming conventions amongst organizations across multiple datasets. Our final deliverable is an iPython notebook that contains code to reconcile alternative organization identifiers. The notebook takes in "official" organization names and returns potential alternative names that refer to the same organization.

Requirements, setup, and usage information is contained within the iPython notebook, as well as the README file contained in the project directory.

- Code and documentation: https://github.com/nketchum/si485-fuzzy-matching.git (*You must be added as a collaborator to access this repository.)*
- Demo video: https://drive.google.com/file/d/1SKS0D7L4ChkmMsverInDZV472-QIYtRR/view?usp=sharing

Input/Output

The fuzzy matching model processes data from two sources: the Orbis database by Bureau van Dijk which contains private company data, and the GDELT knowledge graph, which, in part, contains a database of online news articles that can be sorted by time and location. The tool reads "official" company names registered within a particular country from Orbis, and "unofficial" company names that are mentioned within various online news articles from GDELT. Input data from both databases are stored as CSV and Excel files in an input folder within the project.

The tool outputs several progress files during execution, along with a final output file containing a dictionary of official company names and their possible variants found in online news articles.

- Test input data: https://drive.google.com/drive/folders/1W-YW7AgZFmaURkman-IAp5vNUk56qKGS?usp=sharing
- Test output using data from Sierra Leone: [https://drive.google.com/drive/folders/1mFuDGppvwxO-T09agvrkdHo9sSU-f3ko?usp=

sharing](https://drive.google.com/drive/folders/1mFuDGppvwxO-T09agvrkdHo9sSU-f3ko?usp=sharing)

Methodology

Before making comparisons, all organization names are cleaned by removing punctuation, unusual spaces, and various suffixes such as “inc” and “llc”.

Similarity between names is measured by a few methods. The most effective measurement comes from the Levenshtein distance algorithm, which detects similar words and is the industry standard for measuring word similarity; for example, spell check and autocorrect programs utilize the Levenshtein distance. The Jaro-Wrinkler distance, a similar type of measurement, is another.

The Levenshtein distance total score ranges between 0 and 300, and the Jaro-Wrinkler distance score ranges between 0 and 1. Experiments show Levenshtein scores above 280 combined with Jaro-Wrinkler distance scores above 0.9 provide an overall accuracy of over 95%.

Other measures, including jaccard similarity, cosine similarity, dice similarity are also included but seem much less effective than the Levenshtein or Jaro-Wrinkler distances, but more experimentation is necessary to draw firm conclusions. In addition to these measures, a phonetic approximation of each name is conducted, but little improvement and is not utilized in the final matching.

Overall, more analysis is needed before reaching firm conclusions determining optimal scoring and measurement. Methods to analyze errors should be considered.

Results

This project has successfully concluded with a minimum-viable product that effectively disambiguates organization names. Initially, accuracy looks to be satisfactory as a foundation for further development and improvement to the fuzzy matching tool. Several additional algorithms that calculate name similarity have been partially implemented for future development.

A small testing dataset, confined to Sierra Leone in 2020, provides initial insight into the current performance of the fuzzy matching tool. A list of 2084 companies registered in Sierra Leone was compared against 36,000 online news articles. This represents a total of 75,024,000 processed records. From this dataset, 77 official organization names were matched with 120 potential variants contained in news articles.

name_original_orbis	name_original_gdelt	fuzz_similarity	total_score_name	total_score_metaphone	freq_gdelt	jaro_distance
8 INVESTMENT (SL) LIMITED	Investment Company	95.287958	282	300	2	0.944444
	Investment Corporation	95.287958	282	300	5	0.944444
AFRICA DEVELOPMENT CORPORATION (SL) LIMITED	Africa Development	100.000000	300	300	4	1.000000
	African Development	95.476440	288	284	13	0.982456
AFRICA M,EDIA (SL) LIMITED	Africa Media Corporation	100.000000	300	300	1	1.000000
AFRICAN AIRWAYS LIMITED	Africa Airways	94.957895	287	279	15	0.977778
	African Airways	100.000000	300	300	237	1.000000
	African Airways Corporation	100.000000	300	300	5	1.000000
AFRICAN DEVELOPMENT COMPANY (SL) LIMITED	Africa Development	95.476440	288	284	4	0.982456
	African Development	100.000000	300	300	13	1.000000
AFRICAN MINERALS (SL) LTD	Africa Minerals	94.957895	287	280	3	0.934722
	African Mineral	98.477157	294	290	1	0.979167
	African Minerals	100.000000	300	300	78	1.000000
	African Minerals Limited	100.000000	300	300	7	1.000000
	African Minerals Ltd	100.000000	300	300	22	1.000000
AFRICAN NETWORK (SL) LIMITED	Africa Network	94.957895	287	280	16	0.953968
	African Network	100.000000	300	300	5	1.000000
AFRICAN PETROLEUM (SL) LIMITED	African Petroleum	100.000000	300	300	6	1.000000
AFRICANA AIRWAYS LIMITED	African Airways	94.957895	287	300	237	0.912500
	African Airways Corporation	94.957895	287	300	5	0.912500
AFRICAR LIMITED	Africa Co	95.833333	284	278	3	0.952381
	Africa Plc	95.833333	284	278	26	0.952381

Potential matches between organization names registered in Sierra Leone and organization names mentioned in online news articles.

Next Steps

- Find an optimal match threshold by mixing and adjusting the various scoring measures.
- Determine an efficient method of error analysis.
- Research an effective method to match acronyms with full names by using surrounding contextual information.

Unsupervised Topic Model

The second project is an unsupervised topic model that identifies themes from articles, essays, and other text-based media coverage. Themes of interest include political polarization and controversy, geopolitical instability, public policy, national security, and economics among other possibilities - trends and phenomenons that affect business and industry risk. Our program is split into two parts: content (text) extraction and the topic model code.

Similar to the Fuzzy Matching Model, the requirements, setup, and usage information is contained within two iPython notebooks, as well as the README file contained in the project directory.

- Code and documentation: https://github.com/senaitdafa/LDA-Topic-Model (*You must be added as a collaborator to access this repository.)*

Input /Output

"Content Extraction" ipython notebook: The program inputs a CSV file that contains data from one country, Sierra Leone, populated from our client's UPenn Box databases. The CSV contains a column with links to online articles about current events in Sierra Leone. Each link is input into our web scraping tool, and readable content is output into a new, appended column. The output of the content extraction notebook is the same csv file, but with an additional column for readable content.

"Gensim Topic Model " ipython notebook: The input for the topic model is the output CSV from the content extraction notebook. The aggregated text from each article is processed in the topic model notebook, and the most salient topics are output, which we present in a visualization.

Methodology

The databases hosted by UPenn provide us with web articles organized into a csv, which we have scrapped and formatted to be processed by our model. Our team identified "Goose" as the most error-free and web scraping tool. The team has applied this library to the Sierra Leone dataset, generating output files of extracted article content.

We chose to perform the unsupervised topic model with Python's Gensim natural language processing library. We found previous research work[1] that compared topic model algorithms for short form, article length data, which is also the type of data we are working with. The research provided evidence that the LDA algorithm provided the highest quality and most coherent topics amongst the algorithms studied.

We compared two LDA algorithm implementations in python: the Gensim package, and a popular Gensim wrapper package, Mallet[2], developed by the University of Massachusetts. We found that the Mallet algorithm improved the quality of our topics, which was consistent with our online research[3]. We determined this by calculating coherence scores, which were lower for the model utilizing gensim alone, as well as manually examining output.

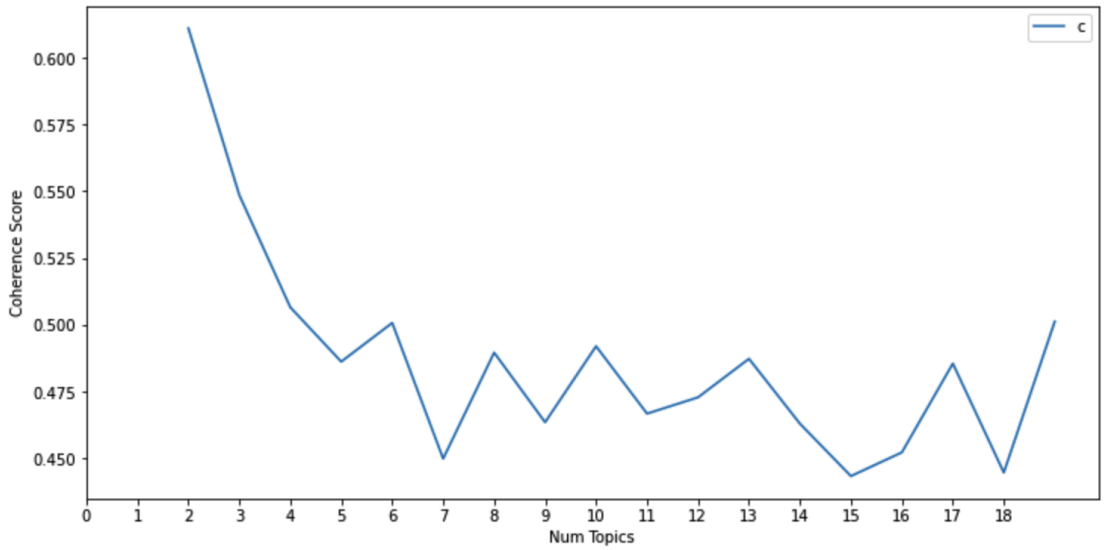
---

[1] https://www.frontiersin.org/articles/10.338u9/frai.2020.00042/full
[2] https://mimno.github.io/Mallet/index
[3] https://www.researchgate.net/publication/331972126_Mallet_vs_GenSim_Topic_Modeling_Evaluation_Report

The topic model does not produce single word/phrase labels for each article. Rather, the output is a keyword list. Our ultimate goal is to produce topic labels for each article, by qualitatively analyzing lists of keyword groups manually. Our results illustrate the topics identified for news articles Sierra Leone in 2020, and the keywords(subtopics) we found to be associated with the topics.

Results

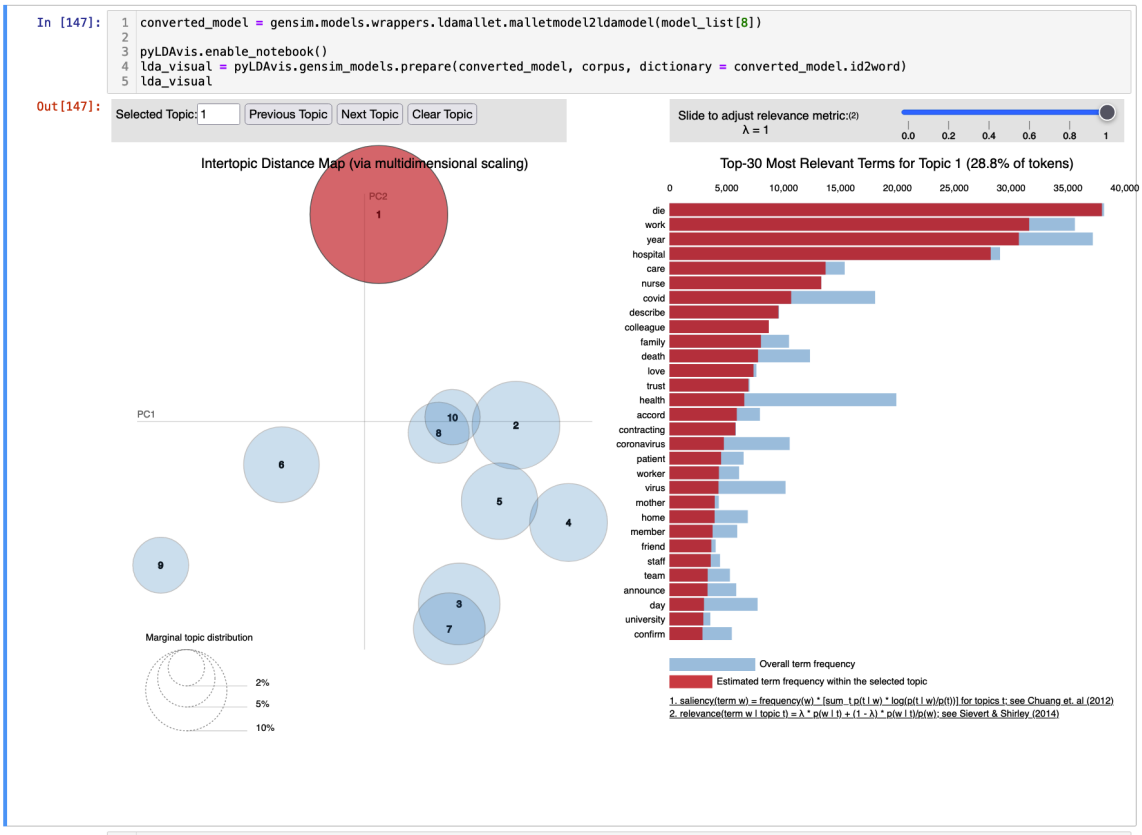
The quality of the topics changes with the number of topics that we choose to identify for the model. Our goal was to find high quality, coherent topics that don't have overlap or repeat. We produced and compared the coherence score of 20 models, which is depicted in the graph below. We identified 3 models with high coherence scores: models with 6, 10, and 19 topics.



After examining the three models, we found that a model of 10 topics has the best coherence. We were also able to find the "dominant topic" for each article in the dataset. Each topic is a collection of keywords, from this, we can manually infer the topic name. Below is the output of the optimal model: it displays each topic, the keywords that describe the topic, the number of documents described by the dominant topic, and the percentage representation of the topic weight in the dataset.

Dominant_Topic	Keywords	Document Number	Document Percent Contribution
0	people, time, make, woman, year, call, day, pl...	842	0.1552
1	case, country, people, virus, coronavirus, hea...	737	0.1358
2	die, work, year, hospital, care, nurse, covid,...	243	0.0448
3	country, travel, include, due, announce, day, ...	275	0.0507
4	government, state, country, international, afr...	561	0.1034
5	health, covid, support, country, include, resp...	477	0.0879
6	school, child, education, student, girl, peopl...	225	0.0415
7	company, business, project, market, food, year...	501	0.0923
8	woman, health, people, country, study, disease...	377	0.0695
9	page, find, access, website, court, press, pol...	1189	0.2191

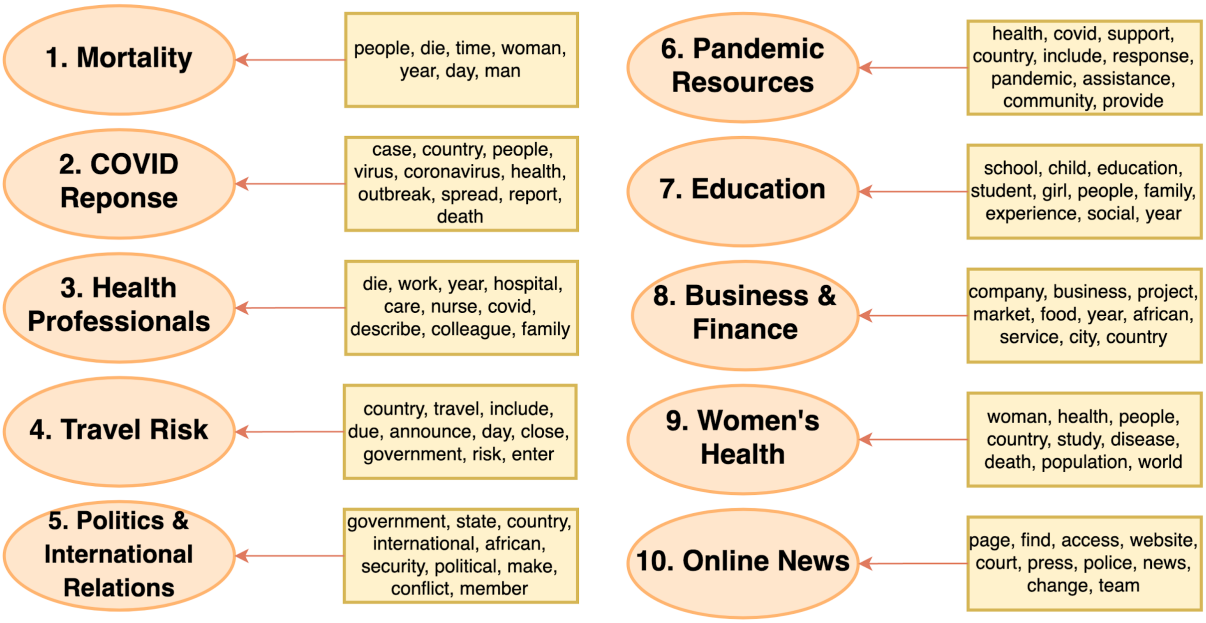
The graph below is interactive, and displays the importance of each keyword in each topic, which allows us to better understand what the label for the topic should be:



On the left, each bubble represents the topic and its weight in the dataset as described by its size. The keywords are on the left. In the screenshot above, topic one is selected. Each topic label can be inferred from the story that its keywords describe. The most important keywords for topic one

are “die” “work”, “nurse”, “hospital”, “hospital”, “colleague”. This topic describes the healthcare industry and healthcare workers.

Finally, the visualization below is our qualitative interpretation of the topic labels, inferred from the keywords and their weights. The orange ovals represent the topic labels, and the yellow squares contain keyword/subtopics generated by our model.



Below is the word cloud visualization generated by the associated LDA wordcloud library for this data. Larger words have more representation in the dataset, and is helpful for determining the topic labels:


Topic 1
play
make
people
call
woman
man
year
day
time
back
Topic 2
health
death
country
people
spread
case
virus
report
coronavirus
outbreak
Topic 3
work
care
colleague
die
covid
year
family
hospital
nurse
describe
Topic 4
announce
day
due
include
risk
travel
enter
government
country
close
Topic 5
member
international
government
african
conflict
state
security
political
country
make
Topic 6
country
include
covid
provide
pandemic
health
response
support
community
assistance
Topic 7
people
family
experience
year
child
education
school
girl
student
social
Topic 8
company
year
city
service
african
food
business
country
project
market
Topic 9
world
study
health
woman
make
people
disease
population
country
death
Topic 10
court
police
page
find
team
news
change
access
press
website

Next Steps

- Separate the interface and the implementation for the topic model code to allow for ease of abstraction
- Test implementation viability on larger datasets such as longer time periods or multiple countries.

Given that our contributions to ACDS's conflict barometer have been split into two separate projects, the team has been successful in achieving the minimum viable product for both models.