Ranking the ExtraTerrestrial

Klein, Joseph
University of Michigan
S State St, 105
Ann Arbor, Michigan 48109
armennen@umich.edu

Lee, Younghun
University of Michigan
S State St, 105
Ann Arbor, Michigan 48109
younggns@umich.edu

Liu, Jingye
University of Michigan
S State St, 105
Ann Arbor, Michigan 48109
jingye@umich.edu

ABSTRACT
Our project analyzes data from NUFORC (National UFO Reporting Center) website [1] (http://www.nuforc.org) regarding UFO encounters and implements a search engine to efficiently comb through the vast database and identify similar occurrences. We employed various crawling techniques to get the data from the website in a quick, efficient manner that avoided overloading the server and cut down (as much as possible) computation time as we needed to scrape information from over 100,000 pages. Many cutting information retrieval methodologies are used in the search engine, such as inverted index, TF*IDF, Page Ranking algorithm, query expansion and so on. We also have evaluated these technologies based on their performance on our dataset.

CCS Concepts
• Information systems → Information systems applications → World Wide Web → Web searching and information discovery → Web search engines → Page and site ranking

Keywords
Search engine; Crawling; Cleaning; Indexing; Page ranking.

1. INTRODUCTION
Eye-witness accounts of Unidentified Flying Objects (UFOs) have garnered a high level of interest over the last 70 or so years and many people have shared their experiences in a wide range of mediums including online resources like NUFORC (National UFO Reporting Center). NUFORC maintains an active and growing archive numbering over 100,000 eye-witness UFO accounts from all around the world ranging from close encounters of the first kind (seeing a UFO), second kind (perceiving physical changes in the environment or self as a result of a UFO), and even close encounters of the third kind where the witness directly perceives humanoid bodies. The repository at NUFORC is volunteer driven and is the direct result of individuals contacting the agency either via the public hotline or in writing via regular mail or form submission to share their first-hand or second-hand UFO-related experiences.

While the NUFORC website offers a wealth of different UFO encounter testimonies, there is not any easy way of navigating through them for any particular keyword, or to find a group of similar instances other than manually combing through the countless witness accounts categorized by either location, date, or shape where any specific category type will contain hundreds, if not thousands, of accounts. Our group sought an efficient way of classifying similar instances based on an initial query and needed to create an effective search mechanism that indexes the entire set of data as of March 2016.

2. TOOLS / DATA
All data was scraped from the NUFORC website database (http://www.nuforc.org/webreport.html). Then we used libxml2dom[1] to parse html with a high level of efficiency and speed. For creating inverted index and parsing queries, we imported a library called Whoosh[2]. We also used scikit-learn[3] library to vectorize documents and calculate cosine similarities between queries and documents. We also imported NetworkX[4] and matplotlib[5] library to generate graphs, visualize, and calculate PageRank over the results.

3. SYSTEM DESCRIPTION
3.1 Initial Crawling and Data Scraping
The initial crawling and gathering of the NUFORC UFO data was completed via a scraper built in Python that visited every page (as of March 2016) that contained a UFO account. In the NUFORC website, UFO occurrences are organized by date, location, and shape. Only the shape organization produces a manageable number of categories as opposed to the thousands of unique location and dates, so it was chosen to be the seed directory to begin the site crawling. There was not a robots.txt found in the NUFORC web domain but care was taken to make sure that the repeated calls did not overload the website's servers by setting sleeps between each new call and by only parsing each page once as organized by the shape category [2]. This was also enforced by the two tier approached to crawling where the first iteration crawled each shape page and gathered all of the unique <href> links and then concatenated with the domain and path of the seed URL in the second iteration (after ensuring there were no repeats) to parse the individual witness testimonial pages. Each page had a timeout parameter set of a maximum 3 attempts (after which the page was noted and skipped entirely) with an initial timeout set at 8 seconds and each additional attempt after 20 seconds or more.

3.2 Data Cleaning
When the data was initially gathered, there were several data cleaning steps before it was first output to the csv file. First the leading tags for all of the form data was removed (i.e. words such as "Location", "Duration", or "Occurred") and all extraneous whitespace and punctuation was removed. Also, all of the HTML elements such as ampersands and special quotes were either

[1] LibXML2 (http://www.xmlsoft.org/) python bindings.

[2] Python library for indexing text and then searching the index.

[3] Machine learning library in python.

[4] Python language software package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks.

[5] Python 2D plotting library

replaced or removed in the first pass of the cleaning. The body of the text had stop words removed and lowercased for the cosine similarity and PageRank algorithms though the original text is preserved for user readability.

rowNumber	OccurrenceD	Time	Location	loc_key	City	State	Country	LatLong	Shape	Duration	Description
103942	6/24/96	0:30	Aurora, CO	204410	Aurora	CO	US	[39.7294319, -104.8319195]	Changing	1 hour	Referred to Marilyn
962	3/27/06	23:59	Baie D'urfe ((	400001	Baie-D'Urfe	QC	CA	[45.416456, -73.9160797]	Changing	2 seconds	Large fast pulsating
1269	6/13/02	23:59	Los Angeles (	400002		CA	US	[36.778261, -119.4179324]	Changing	15min	I WAS TRAVELING FR
1656	2/4/01	23:58	Vancouver (C	205743	Vancouver	BC	CA	[49.2827291, -123.1207375]	Changing	10 sec	What looked like a fl
344	10/14/06	23:56	Beverly Hills,	400003	Beverly Hills	FL	US	[28.9169245, -82.45815360(	Changing	about ten mi	Went outside for a l
317	6/17/02	23:56	Niagra-on-th	400004	Niagara-on-t	ON	CA	[43.2549888, -79.0772616]	Changing	approximate	four balls of light tha
347	10/8/07	23:55	Newport, NC	400005	Newport	NC	US	[34.7865497, -76.8591057]	Changing	almost 1 hou	Bright blinking rows
200	9/5/07	23:55	Roswell, NM	205664	Roswell	NM	US	[33.3942655, -104.5230242]	Changing	10-15 secon	dim lights moving in
909	3/6/98	23:55	Southamptor	400006	Southampton		GB	[50.9411176, -1.4230311]	Changing	20 minutes	we had been out an
1123	10/30/10	23:50	Broken Arrov	200819	Broken Arrov	OK	US	[36.060949, -95.7974526]	Changing	30s	Seven unexplained l

Figure 1. Sample data input collected by web crawler.

3.3 Building a Search Engine

Our team imported Whoosh for building basic structure of search engine. Whoosh is a fast, stand-alone Python library that doesn't require any other libraries or compilers to run. Whoosh creates inverted index with pre-defined schemas and lets users to specify fields they want to integrate into the index. In this project, we came up with using document number, description of each observation, observed location (name of the state) and time. With this schema, Whoosh can perform search with simple queries.

3.4 Creating a Baseline Set

By default, Whoosh uses Okapi BM25 function for sorting documents so that it retrieves results from the bag of words by using TF*IDF to simply look at frequencies. However, we thought that the result from Whoosh's ranking itself is not enough for our project. Because we assumed that users are going to put simple queries to search by stating a couple of keywords, and we need to expand their queries in order to retrieve more relevant sets of results. Therefore, we decided to consider the sorted result from Whoosh to be a baseline set for comparing with other result sets.

3.5 Retrieving Relevant Set by Using PageRank

We implemented the Python module NetworkX library to calculate PageRank and examine the inter-relationships between documents which enables us to produce more relevant results. Since PageRank computes importance of each node based on incoming links in the graph, we had to create a graph from the documents. These documents are not actually connected by direct links, but we assumed that similar documents tend to be related to each other so we create edges by calculating cosine similarity between all of the similar documents over a specified threshold. This returns a graph and a PageRank for each node. We consider the set of top results from PageRank as a result set #1.

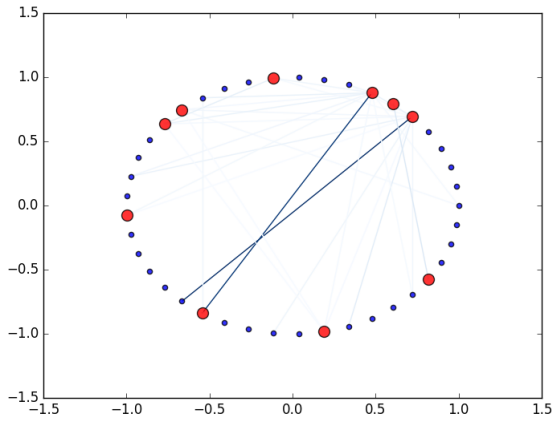

Figure 2. Cosine similarity network on search result for the query term 'little green alien'. Red dots mean top relevant documents, and blue dots are the other documents. The color of each edge maps to the similarity between documents – if the edge is thicker, their similarity value is higher than others.

3.6 Query Expansion

As mentioned above, we decided to expand the original query and use a built-in library in Whoosh. Whoosh provides query expansion feature by extracting key terms which are frequent in the given documents, but relatively infrequent in the indexed collection as a whole. These key terms are not actually based on Natural Language Processing methods or any other Artificial Intelligence algorithms but statistically generated. However, it still guarantees a high level of similarities with the original query since key terms are retrieved from top N results sorted by Whoosh. With these extracted key terms, we got different set of results by searching expanded query. We consider this set as a result set #2.

4. EVALUATION

In short, we came up with three sets of results from each query.

1) Baseline: Search for the original query. Use Okapi BM25 function to sort result set.

2) Result set #1: Use Baseline to create a graph. Calculate PageRank from the graph and retrieve top results

3) Result set #2: Use Baseline to expand original query. Search for expanded query.

As a last step of the project, we wanted to evaluate the system by comparing each result set. Therefore, we conducted manual human judgments and calculated average value of each set over everyone in the group. To do this, we had to define the scale system and we came up with the five levels of scaling with definitions described below.

Table 1. Scale systems used in the evaluation

Scale	Description
0	Result document is totally irrelevant from the general topic (UFO).
1	Result document is about the general topic(UFO), but doesn't have query terms in it.
2	Result document is about the general topic and includes

	query terms, but has different context from their original meaning.
3	Result document is about the general topic, includes query terms, and has same context with their original meaning, but at least one of the query terms in the document is directing different object from other terms.
4	Result document is about the general topic, includes query terms, has same context with their original meaning, and all query terms in the document are directing same object.

First of all, the reason why we came up with scale of zero is that we cannot make sure about the quality of input document set. All input documents grabbed from NUFORC website have to be related to UFO observation, but there are also possibilities of having fake documents in the website like a random post. And we wanted to differentiate fake results from other UFO observation documents even if they are not related to query terms.

As an approach to define neutrality of the document, we came up with the situation that the retrieved document has the query terms but their context is different from original meaning. For example, if the query is 'chasing disc', the user might want to get results like 'I saw the red light with chasing disc', not 'I saw the UFO when my dog was chasing the Frisbee disc'. We categorized these documents as a scale of two.

The documents which have scale of three and four are all related to query terms, but the difference between scale three and four is the relationship between query terms. For example, if there are two retrieved documents from the query 'little green alien'. One document is 'I saw a little green alien' and the other is 'I saw a green alien coming from the little UFO'. In this example, all two documents have all query terms and same context. But the query terms in the first document are all related to each other and pointing the same object (alien), but in the second document, only two query terms out of three are related to each other - 'green' and 'alien' are pointing the same object (alien), while 'little' is pointing another object (UFO). We assumed that most of users might want to search phrases if they are searching for multiple terms, so we came up with the idea of differentiating each term's pointing object.

5. DEMO OUTPUTS

We did our demo with searching three different queries - 'fireball fast', 'little green alien', and 'area 51'. First of all, we used data crawled from NUFORC website and get three sets of results - baseline, results with calculating PageRank, and results with expanded queries.

```
JINGYEs-MacBook-Pro:project LIU$ python search_by_query.py sample_queries.txt

You are searching:
little green alien
number of hits
41
Do you want search?
desc:her desc:me desc:little desc:green desc:alien
(desc:her AND desc:me AND desc:little AND desc:green AND desc:alien)
Top 10 ranking by nx pageranking
[u'99049', u'20282', u'85824', u'20371', u'343', u'82786', u'49080', u'39465', u'22100', u'84749']
Top 10 Ranked Document by TFIDF (our baseline)
[23299, 6037, 68782, 22209, 41310, 66042, 42347, 25218, 1606, 89864]
Top 10 Ranked Document by Query Expansion with Query Expanded to (desc:her AND desc:me AND desc:little AND desc:green AND desc:alien)
[u'92440', u'81879', u'59942', u'39465', u'25447', u'98541', u'21735', u'82786', u'22100', u'20371']
```

Figure 3. Query searching interface.

We grabbed top ten documents from each result set and all of our team members conducted judgment by using the scale system described in the Evaluation section. After doing this activity, we could come with the average relevance score of each result set and were able to determine the efficiency of methods.

Table 2. Search result about 'fireball fast'

Model	Average Score	Top Result Document
Baseline (TF*IDF)	3.9	Fireball flying fast east over West Boca Raton. Red-orange fireball, flying east in straight line, fast.
PageRank	3.47	White fireball moving across the sky I was walking my dog around 4:00 am and I noticed a really bright looking fireball move across the sky. It was not that fast moving.
Query Expansion	3.43	Bright orange fireball seen traveling across night sky from Cherry Hill, NJ, traveling at pretty fast speed. No sound involved. We were sitting outside in our yard, when I noticed what appeared to be a huge orange fireball sailing across the sky, it appeared to be traveling at the speed of an airplane or faster, but it was not an airplane or anything like that. (…)

Table 3. Search result about 'little green alien'

Model	Average Score	Top Result Document
Baseline (TF*IDF)	2.17	Alien Tests Soil I saw an alien and a UFO in Sawyer Michigan in 1990. I was eight years old. There is a couple from Chicago who own a little cottage who are barely ever there...probably why the alien chose that spot. (…)It was testing the soil with something that looked like those lights that commonly light walkways up to a person's house only the light from it was neon green (…)
PageRank	2.3	(…) I saw some things moving in the brush. Little objects scurrying about very quick like that of coyote's or something. My heart began to thump. (…) On the left bottom story in the windows were aliens that I first saw. There was a female. (…) Their eyes were shaped like the pictures but they were white with irises. Blue eyes, the girl had green eyes. The one with the draped skull on the second story was standing and holding a clipboard against himself in his white robe.
Query Expansion	2.6	(…) Me: What do you mean you went to a space ship? Her: Some aliens came into my room and took me to their space ship? Me: But when mommy came into your room you were sitting on your bed. (…) Me: They had no mouth? Did they have a nose? Her: really little. But they had big heads Me: How tall were they? Her: A little taller than

		me. (…) She interupts me and says, It hurt a little when they poked me but they were nice. and I think they were green or something. I then asked again if she could draw me some pictures and she repeated the I don't know how you'll have to show me answer.

Table 4. Search result about 'area 51'

Model	Average Score	Top Result Document
Baseline (TF*IDF)	3.47	Possible spy satelite over Area 51. From FL390 we saw a star like object moving high above. We asked Oakland Approach what was flying above. He said "Nothing on my radar." I concluded that it was a possible spy satelite passing over Area 51. As the sun was rising it made for a perfect exposure and remained lit while proceding to and from the view of our aircraft.
PageRank	1.9	(…) I also couldn't see any moving white lights, only stars. About 5-8 miles away, down the dirt road toward area 51 I did spot a lot of vehicle lights that appeared to be on high beams. I had noticed no traffic before this time so I decided to move on in case they were base security.
Query Expansion	3.7	One hoax "UFO," and one possible test plane, over Hiko/Ash Springs/Alamo area. Multiple sightings between November 2013 and January 2015 (when this report was filed) Multiple sightings of a confirmed hoax craft; And a single sighting of one possible test craft from Area 51 or Nellis AFB Location: State Route 318 from Lund to Hiko and continuing south along US 93 to just south of Alamo.

From these results, we could come with overall average scores about each methodology.

Table 5. Search result about 'fireball fast'

Model	Average Score
Baseline (TF*IDF)	3.18
PageRank	2.56
Query Expansion	3.23

6. Discussion

While the entries in the witness database often have information such as time and place of the event and other information, the crux of each testimony - a free-flowing text area - was not consistently encoded or transcribed. The scraping process incurred multiple tracebacks due to the wide variety of encodings used for the target web pages.

Initial scraping was completed with the Beautiful Soup module for Python but lxml offered a significantly faster parsing speed (though admittedly some of the encoding errors could have been avoided possibly had we stuck with Beautiful Soup).

7. Conclusion

Initial scraping was completed with the Beautiful Soup module for Python but lxml offered a significantly faster parsing speed (though admittedly some of the encoding errors could have been avoided possibly had we stuck with Beautiful Soup).

The NUFORC dataset as it exists on the web is difficult to navigate through a browser and even more cumbersome to scrape due to the inconsistencies (e.g. encoding, HTML layout, content) from page to page. Through a series of trial and errors, we were able to crawl the entire database (numbering over 100,000 pages) and return a normalized dataset. Additional processing steps included adding latitude/longitude coordinates for each of the coordinates along with removing stopwords for more efficient indexing (though the original text was retained as well for the search results).

After indexing the dataset, we then created a search engine to query the entire dataset to identify the most relevant documents (per the given query), as determined by manually scored sample result sets. Through our evaluation we found that the search engine returned the most relevant (on average) results when we implemented query expansion as opposed to PageRank or the baseline (which is solely TF*IDF).

The NUFORC dataset as it exists on the web is difficult to navigate through a browser and even more cumbersome to scrape due to the inconsistencies (e.g. encoding, HTML layout, content) from page to page. Through a series of trial and errors, we were able to crawl the entire database (numbering over 100,000 pages) and return a normalized dataset. Additional processing steps included adding latitude/longitude coordinates for each of the coordinates along with removing stopwords for more efficient indexing (though the original text is returned in its entirety for readability purposes in the search results).

After indexing the dataset, we then created a search engine to query the entire dataset to identify the most relevant documents (per the given query), as determined by manually scored sample result sets. Through our evaluation we found that the search engine returned the most relevant (on average) results when we implemented query expansion as opposed to PageRank or the baseline (which is solely TF*IDF).

8. REFERENCES

[1] NUFORC online database, http://www.nuforc.org/webreport.html, Retrieved from April 15, 2016.

[2] Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to information retrieval (Vol. 1, No. 1, p. 496). Cambridge: Cambridge university press.