Nate Andre
Mateo Ibarguen

Complementary Fact-Finder

I. Abstract

We aim to create a tool which provides interesting facts that are highly personalized to the subjects that students write about for their papers. These facts will motivate students to do more in-depth research about topics that would complement their essays. We believe in turn that this additional motivation will help improve the writing proficiency of students.

Our finalized program has the following components:

1) A Latent Semantic Index trained on key words from 5000 Wikipedia articles.
2) After a user-input essay is received, we tokenize the essay and obtain keywords from it. We then compute its tf-idf weight with respect to the Wikipedia corpus.
3) We use cosine-similarity to obtain the articles with the most similar topics to the student's essay.
4) After obtaining these articles, we sum over the tf-idf weights of each sentence in each article in order to obtain the most informative sentence per top article. We consider the most informative sentence of each article as the sentence with the largest sum of its tf-idf weights.

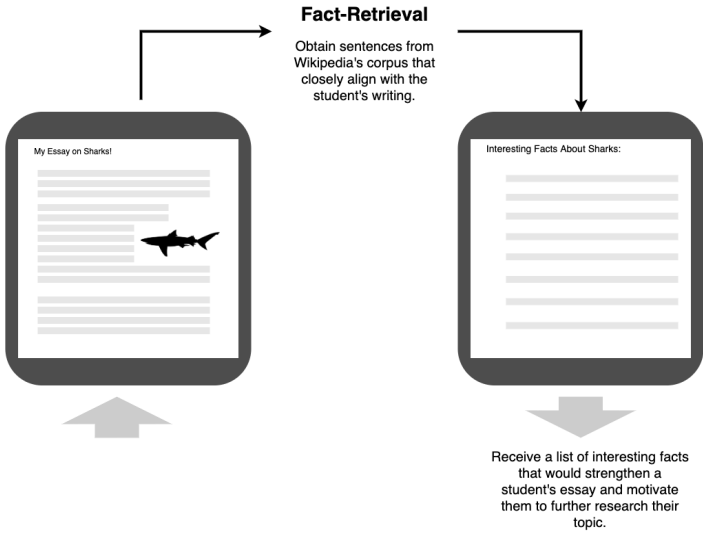
The student will therefore receive informative sentences from articles that are specifically tuned to their content and which students can incorporate in their essays to strengthen their writing skills.

Even though our finalized program uses LSI with the tf-idf matrix as its input, we also experimented with a variety of different models and inputs.

II. Motivation

In order to improve their writing skills, students should include interesting facts and details about their subject matter. However, it is important that these facts to fit well in their essays, and they shouldn't be added without considering the full context of their writing. In order to motivate students to include facts in their writing and improve their researching skills, our tool will provide interesting facts that are specifically suited for the essays they

write. We hope our tool will motivate students to do further research about their topics and give them ideas about further topics to research that would fit well with their essays.



III. Data Extraction

A. Student Corpus

Given the scope of our project and its focus on unsupervised methods to model topic and content information, extracting data from the student corpus simply involved retrieving the raw text from each of the student essay objects.

B. Wikipedia Corpus

You may download the Simple English Wikipedia corpus in the following website: https://dumps.wikimedia.org/simplewiki/latest/. We used Gensim's WikiCorpus class to help us parse the Wikipedia corpus in a quick and efficient manner. After some trial and error, we noticed that our corpus should only contain the abstract section of the Wikipedia articles which is the section before the table of contents. We also decided to exclude articles that only contain lists such as: 'List of Noble Peace Prize Winners' since these articles don't contain complete sentences that would help us in our project.

IV. Data Exploration

A. Data Cleaning

Our approach to data cleaning was different based on whether we were working with the Wikipedia text, which was structured with proper english

notation, or with the student text, which was full of spelling errors, grammatical mistakes, and other issues. With the Wikipedia text, we used the standard NLTK tokenizing workflow and found it to be sufficient. For the student text, we found that the misspellings and improper punctuation made simply using standard NLTK tools difficult.

To tackle the issue with misspellings, we tested out using external packages to determine whether words were misspelled and then dropping them from our vocabulary. We also considered rectifying the misspellings, but found that the correction accuracy was horrendous and might significantly negatively impact out topic modeling. Further, given that many of our topic modeling techniques relied on using bag-of-words models, we felt that decreasing the vocabulary size would be advantageous. However, after testing the misspelling detector, we found that it was flagging many correctly spelled words as invalid, and so the false positive rate was too large for our liking. Thus, we ultimately decided to remove this preprocessing step from our data cleaning pipeline.

In the end, we decided to keep the misspelled words because we felt that their effect on our topic models would ultimately be limited (apart from the fact that we might be losing information). However, we removed words which were clearly invalid, such as words containing letters and numbers. For the keyword and average word embedding models, the misspelled words wouldn't be considered. For our LDA and LSI models, they would increase the size of the vocabulary, but we felt that their impact on model performance would be minimal.

For both the student and Wikipedia text, we used the lemmatizer to get the roots for all of our tokens, so we could cut down on the overall size of our vocabulary. We found that this step wasn't too computationally expensive. We also decided to remove stopwords to further cut down on the overall size of our vocabulary. We also incorporated other standard techniques into our data processing pipeline such as removing punctuation.

B. Keyword Extraction

We wanted to design a simple keyword extraction method to quickly extract the important words within a given block of text, while minimizing computational complexity. We decided that the easiest way to do this would be to use NLTK's POS tagger to identify nouns and verbs and

subsequently extract them. We chose to ignore modifiers such as adverbs and adjectives, along with other parts of speech, in order to keep our algorithm simple. It's important to note that our keyword extraction pipeline otherwise uses the same steps as our standard data cleaning pipeline.

V. Topic Models

A. Keyword Comparison

To create a baseline for our more advanced unsupervised learning techniques, we created a simple model which compares the keywords between each of the Wikipedia texts and a given student article. A similarity score is calculated based on the sum of unique keywords in common divided by the total number of unique keywords in the wikipedia text. The suggested Wikipedia articles are then selected from those with the highest similarity score. We chose to focus only on unique keywords as we wanted to make this initial model as simple as possible, and doing so would allow us to decrease computational complexity through using Python sets. Further, we scaled by the total number of unique keywords in the wikipedia text so that longer articles were not prioritized over shorter articles.

As mentioned earlier, this approach is relatively immune to there being misspelled words in the student text, as the Wikipedia articles will not contain misspelled words. Overall, this approach is of course, naive, as the topic of a body of text does not only rely on the keywords within the text, but how they interact. Given this, our future modeling techniques attempt to explicitly model the overall topic.

B. Average Word Embeddings (Word2Vec)

To remedy our naive keyword comparison approach which does not account for the overall topic in any way, we created a model based on comparing the similarity between the average word embeddings for a given student text and all the Wikipedia texts. The average word embeddings were calculated by looping through the tokens extracted from a text block and averaging the embeddings for all tokens which were found to have Word2Vec embeddings. This process ignores misspelled words, as they don't have associated embeddings. Euclidean distance was used to compare the similarity between any two sets of average word vectors, as an efficient solution to vectorize the distance computation over the entire wikipedia array of average word embeddings was developed

which removed the need for explicit looping in Python. The suggested Wikipedia articles are then selected from those with the lowest distance scores.

We found that this approach seemed to do better at capturing the underlying topic of the student text, however there were still issues with detecting certain nuances. We attributed these issues to the naive assumption of using average word embeddings, that each of the words within a given block of text equally contribute to the overall topic of the text. Our future modeling is aimed at improving upon this initial attempt at topic and content modeling.

C. LDA

Latent Dirichlet Allocation is an unsupervised machine learning method which allows topics to be inferred from a bag-of-words obtained through a corpora. The general assumption on which LDA is based on is that each document can be represented as a mixture of topics each of which corresponds to a variety of words. If we know the distribution of words in each topic a priori, we can use a Bayesian technique to determine what the highest probability is for a document to belong to a certain topic. Since LDA is a generative probabilistic model, we make some assumptions about the underlying distributions of the data:

a) Each observation is treated as part of a generative probabilistic process.
b) We can infer the hidden structure (in a similar manner to Hidden Markov Chain Models) by using posteriori inference.

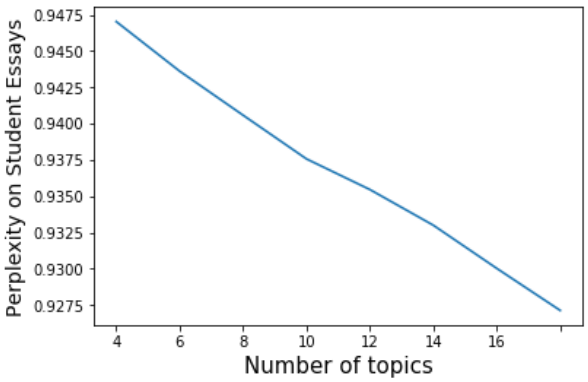
Results

We can evaluate our LDA model with the perplexity of our model trained on the Wikipedia corpus but tested on the student corpus. Wd is the set of unseen documents, Φ is the topic matrix and α is a hyperparameter for the topic distribution of documents.

$$\mathcal{L}(\boldsymbol{w}) = \log p(\boldsymbol{w}|\boldsymbol{\Phi}, \alpha) = \sum_{d} \log p(\boldsymbol{w}_d|\boldsymbol{\Phi}, \alpha).$$

$$\text{perplexity(test set } \boldsymbol{w}) = \exp\left\{-\frac{\mathcal{L}(\boldsymbol{w})}{\text{count of tokens}}\right\}$$

a) Perplexity vs. number of topics on student essays

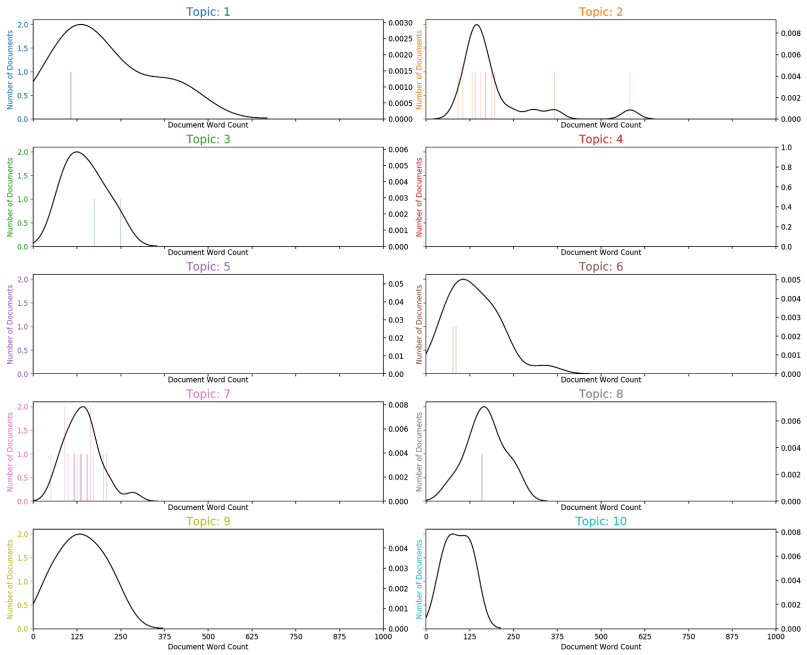


We can visualize how the log-perplexity on the student documents decreases as we increase the number of topics.

Topics visualization

As part of our data exploration, we decided to use LDA to find the top 10 most dominant topics in the Wikipedia corpus. You may find the keywords from these topics in the plot below.

We were curious about the distributions of these topics in the students' essays. We therefore found the dominant topic in each of the student's essays and plotted the frequency distributions of the word counts per dominant topic. It appears as though topic 6 which relates to linguistics, books, and language is not a very popular topic. It also appears that topic 7 which relates to holidays, seasons and events was another topic which was not popular among the student essays.



D. LSI

In order to compute the Latent Semantic Index, we decompose a term-document matrix C into its Singular Value Decomposition. Decomposing the term-frequency matrix into its SVD components allowsedus to obtain a low-dimensionality representation of our term-frequency matrix. We can then use this matrix to compute the cosine similarity of our query and retrieve the documents with the lowest cosine similarities.

The TF-IDF (term frequency-inverse document frequency) is composed by two parts: a normalized term frequency: the bag-of-words representation divided by the number of words in a document, and the Inverse Document Frequency (IDF) which is the logarithm of the number of documents in the corpus divided by the number of documents were the term is present. Therefore, the term-frequency measures how often a term occurs in a document. However, since some terms appear much more often linguistically than others, the term frequency is normalized by the document length.

We found that using the tf-idf, rather than the raw bag-of-words, as input to our LSI model provided better results. After finding the top articles with the best cosine similarity in comparison to our input, we display the sentence with the largest sum of tf-idf weights per article.

Unfortunately, since the LSI model does not provide a distribution of topic probabilities, we could not derive its perplexity to obtain a metric of its performance.

E. Summarize LSI vs. LDA results

After sampling multiple student essays, it was evident that the LSI model far outperformed the LDA model. We therefore decided that our final program would use the LSI model, along with the tf-idf weights to output the most important sentence for each article.

VI. Instructions

In order to run our finalized model, type the following command on the top-level directory.

```
python -m src.main
```

This will output a prompt which will ask you to copy-paste your essay. After doing so, you may press enter and type `--facts` and hit enter in order to obtain the suggested facts. Type `exit` and hit enter in order to escape.

VII. Examples

To show the relative performance between our different topic models, we are displaying three suggested Wikipedia texts for each of out topic modeling techniques to showcase their particular attributes. The suggestions are based on the following student essay:

> The Universe
>
> Introduction
>
> Did you know how the Universe formed ?Keep reading .And ,also my subtopics are Did you know how the Universe is expanding right now , Do you know how a Black hole forms , Do you know how a star forms , Do you know how to travel forward in time …
>
> Did you know how the Universe is expanding right now ?
> Everyone in World knows light is the fast thing in the Universe but many people don't know how far can travel in one sd they say light travels at 1,000 km per sd or more . If you don't exactly how far can light travel in one sd ,it travels.
> The speed of light in a vacuum is 186,282 miles per second (299,792 kilometers per second). "You might have notices this p talks different then the heading "

Mabe light travels very fast ,but the Universe is very very big that million of years to go out of the Universe.
When light travels through the Universe it expands and the space around it also expands . After that the Universe expands forever…

Do you know how a Black hole forms?
Imagine a room of a lot lottery not one shirt will fit in the room, if you try it really hard it will turn into Black hole .There are one ways a Black hole can form . The First way is a really massive star burn really hot and fast . When this progress starts in the stars core there will a little bit iron , it builds up in the core. When the the iron is big enough the star dies in a progress called Supernova. The leftover of the star is the iron it become a Black hole…

Do you know hoe a star forms?
A star form When a lot of hydrogen gas cloud are near to each other . When they are near to each other gravity puts them at high pursue and a star forms.
Do you know that in some ways star can form inside a star ?The First way is ,a really massive star burn really hot and fast . When this progress starts in the stars core there will a little bit iron ,it builds up in the core. When the the iron is big enough the star dies in a progress called Supernova . The leftover of the star is the iron it becomes a Black hole or a Neutron star .These is how star con form inside a star…

Do you know how to travel forward in time?
As you approach the speed of light, your clock runs so slow you could come back 10,000 years in the future.' The theory is based on Einstein's Theory of Special Relativity that states to travel forward in time, an object would need to reach speeds close to the speed of light. These p says if you need to travel forward in time close to speed of light ….

Conclusion
In my book, I thought you about some of the special things in Universe.And again my subtopics are Did you know how the Universe is expanding right now , Do you know how a Black hole forms , Do you know how a star forms , Do you know how to travel forward in time . Now you know some special things in Universe ,now go and became astronaut...

a) **Keyword Comparison:**

Topic: Solar System

It formed by gravity in a large molecular cloud.

Topic: Stars

Stars have a lot of hydrogen.

Topic: Zero

If there are zero things there are no things at all.

b) **Average Word Embeddings (Word2Vec):**

Topic: Calculus

Differential calculus divides things into small different pieces and tells us how they change from one moment to the next while integral calculus joins integrates the small pieces together and tells us how much of something is made overall by a series of changes.

Topic: Black hole

It is made of many millions of millions of stars and planets and enormous clouds of gas separated by a gigantic empty space which is called the universe.

Topic: Earth

Earth also turns around in space so that different parts face the Sun at different times.

c) **LDA**

Topic: History of the Earth

eons to scaleThe history of the Earth describes the most important events and fundamental stages in the development of the planet Earth from its formation to the present day.

Topic: Black hole

Simulation of gravitational lensing by a black hole which distorts the image of a galaxy in the background larger animation History In 1783 an English clergyman called John Michell wrote that it might be possible for something to be so heavy you would have to go at the speed of light to get away from its gravity.

Topic: Alice Springs

Alice Springs is a city in the Northern Territory of Australia.

d) **LSI**

Topic: Universe

It is made of many millions of millions of stars and planets and enormous clouds of gas separated by a gigantic empty space which is called the universe.

Topic: Black hole

Simulation of gravitational lensing by a black hole which distorts the image of a galaxy in the background larger animation. In 1783 an English clergyman called John Michell wrote that it might be possible for something to be so heavy you would have to go at the speed of light to get away from its gravity.

Topic: Galaxy

The observable Universe contains more than 2 trillion 1012 galaxies and overall as many as an estimated stars, more stars than all the grains of sand on planet Earth.

Application: We hope that the student who wrote this paper about the universe and black holes is given more ideas about further topics to research. If I were a student in this position, I would be thrilled to learn that there are “more stars than all the grains of sand on planet Earth. ” I would be awed that there are “2 trillion” galaxies and I would be eager to incorporate these facts into my essay and do further research on stars and galaxies.

VIII. Conclusion and improvements

It went pretty well, but given that we only used a small subset of all available wikipedia articles due to computational concerns, there are times where there might not be enough content for the topic that a particular student chose, and as such sometimes there are weird suggestions that show up. In order to address this concern, we increased the number of articles in our corpus from 1000 articles to 5000 articles. We believe that this increment in the size of our corpus allowed us to obtain higher quality facts.

Considering this was an unsupervised learning assignment, it was difficult to exactly quantify how well our model was performing at extracting the most related articles to the student essays. We had to rely on sampling various student essays and using our intuition to determine which model was performing better.

We had one particular occurance of a student essay about “rocks” which talks about “minerals,” “gold, ” “platinum” and “silver”. Unfortunately, articles about “rock and roll” also mention “gold albums, ” “platinum albums, ” and “silver albums.” Thus, we obtained facts mostly about “rock and roll” when we tested this student’s essay. We experimented training the model with bigrams instead of unigrams and this seamed to fix this issue. However, the model would cease to perform well on other student essays.

As a possible future idea, we could ask users to rate the facts we provide to them. This will allow us to get an idea of how well our model is performing at delivering facts that are suitable for our target demographic. We can cluster similar students and use this information to output facts that we believe a specific student would be more interested in based on the scores collected from previous students.

IX. Project Breakdown

- Nate built the data cleaning pipeline for extracting tokens and features from the Wikipedia corpus and student text, along with building the keyword extractor, developing the keyword topic model, and developing the average word embedding topic model.
- Mateo used Gensim's WikiCorpus module to create a class which would parse the Wikipedia corpus to extract the information that was relevant to our project. He also wrote the LSI and LDA topic models pipelines so that they would display the most informative sentences of the most relevant Wikipedia articles for a student's essay.

X. Tools/Packages

1. NLTK: Word and sentence tokenizer, along with POS tagger.
2. Spacy: Wrapper for Word2Vec vectors.
3. Gensim: NLP tools.
4. Numpy: Numerical computation.
5. Json/Pickle: Efficient data storage.

XI. Citations

[1] Mikolov, T., Et. al (2013). Distributed Representations of Words and Phrases and their Compositionality. *NIPS*. Retrieved from https://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf

[2] Blei, D., Et. al (2003). Latent Dirichlet Allocation. *JMLR*. Retrieved from http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf

[3] Deerwester, S. (1990). Indexing by Latent Semantic Analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*. Retrieved from https://www.crss.ucsc.edu/Papers/deerwester-jasis90.pdf

[4] (n.d.). Retrieved from http://qpleple.com/perplexity-to-evaluate-topic-models/.

[5] McAllister, A., Naydenova, I., & Quang. (2013). Building a LDA-based Book Recommender System. Retrieved from

https://humboldt-wi.github.io/blog/research/information_systems_1819/is_lda_final/.
[6] idf :: A Single-Page Tutorial - Information Retrieval and Text Mining. (n.d.). Retrieved from http://www.tfidf.com/.
[7] Complete Guide to Topic Modeling - NLP-FOR-HACKERS. (2018, February 6). Retrieved from https://nlpforhackers.io/topic-modeling/.