# Remarkable Scientists: ExpertSearch

## Project Status Report

For this project, we worked on three different tasks:

- Topic Mining of the ExpertSearch professor bios.
- Improving Named Entity Recognition & Extraction of professor names from the bios.
- Improving recognition and extraction of professor e-mail addresses from the bios.

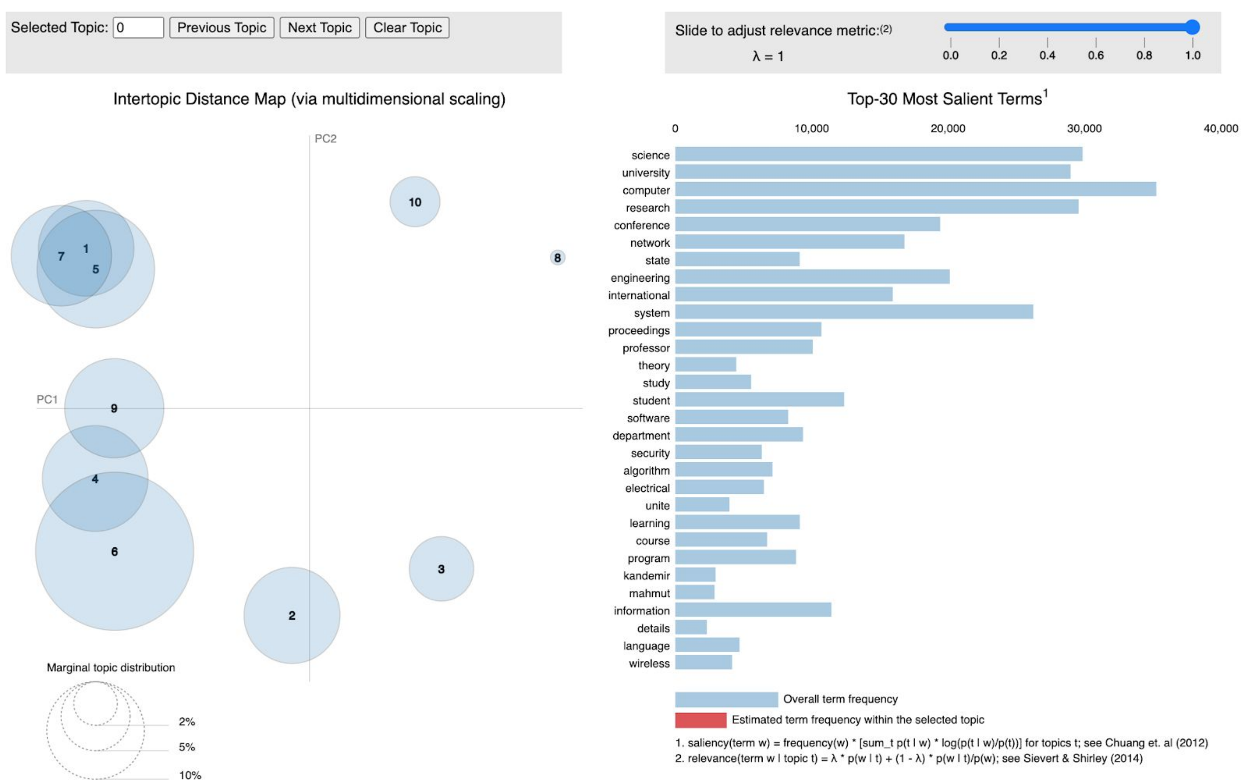
Below is the status report for each.

# Topic Mining of Professor Bios

We were able to combine spaCy, NLTK, and Genism to build an LDA topic model of the professor's bios, using 10 topics. We then used pyLDAvis to visualize the model, and word_cloud to build a word cloud of the 25 highest-weighted terms in each topic.

## LDA Visualization

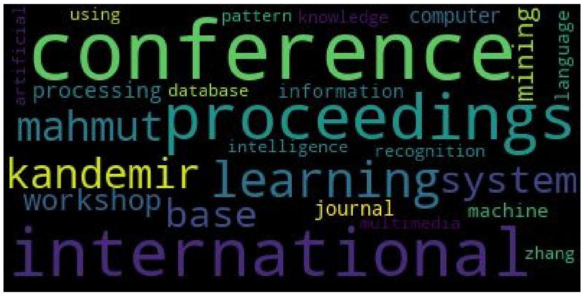
The following was created using pyLDAvis, which is visualizing a 10-topic LDA model:



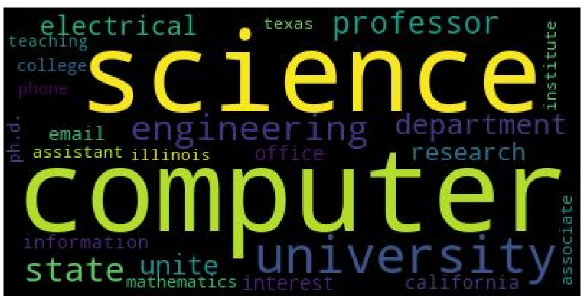
## Word Clouds

The following word clouds show the top 25 words in each of the 10 LDA topics:

### Topic 1 Word Cloud

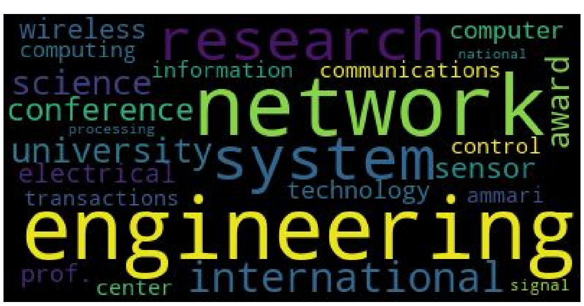



### Topic 2 Word Cloud

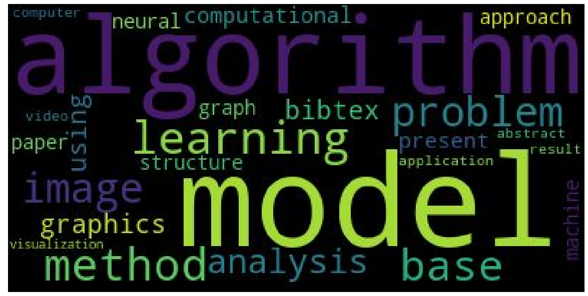



### Topic 3 Word Cloud

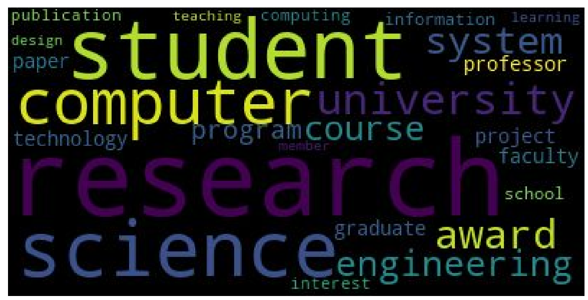

### Topic 4 Word Cloud



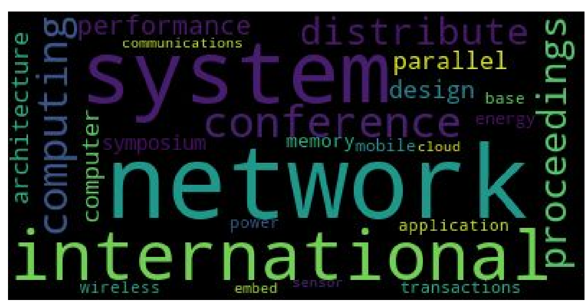

### Topic 5 Word Cloud



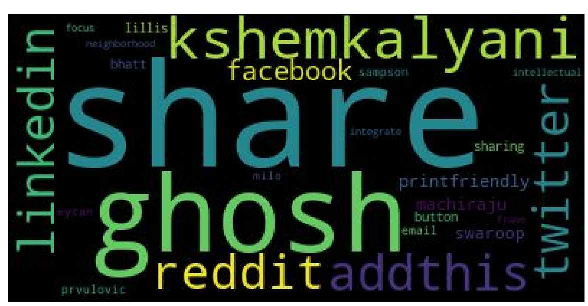

### Topic 6 Word Cloud



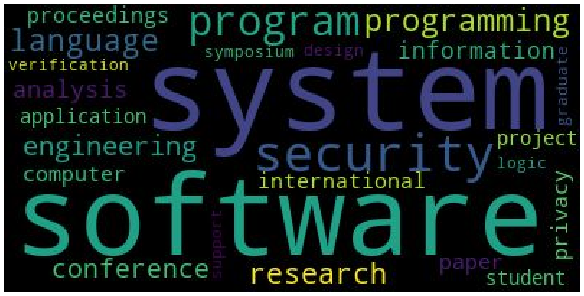

### Topic 7 Word Cloud



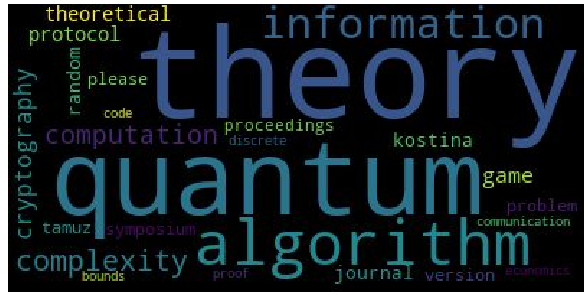

### Topic 8 Word Cloud



Topic 9 Word Cloud




Topic 10 Word Cloud




## Next Steps

We intend to experiment with different topic counts in the LDA model to determine the best results.

The last step for this section would be to provide the word cloud of the most relevant topic in the search results. In short, we would split the main search result body into two parts:

- The left ⅔ of the body would contain the search result text.
- The right ⅓ of the body would contain the topic word cloud.

We are also interested in making the bios more browsable by clicking on one of the topic word clouds in the home page, and viewing the bios by decreasing relevance to that topic. However, this is not a core functionality of metapy, so it is unclear if we would be able to build this additional functionality in the time that we have.

## Challenges

We are not familiar with developing Flask apps in Python using Gunicorn, so unwinding the existing project structure and figuring out how to extend it has been a challenge thus far.

# Named Entity Extraction of Professor Names

We have conducted several experiments on NER(Named Entity Extraction) solutions in the market. The best open source solution would be Standford NER Tagger, which is also provided along with ExpertSearch project for version 2018. The provided tagger and compiled result are incomplete as it eliminates some name information during the tagging phase, part of missing information is critical to extract the main context from the text. We have done two improvements for solving this problem, firstly, we update NER tagger to the newest version which is 2020-11 online, and test on 3-class model and 4-class model separately. Secondly, we implemented a filtered mechanism from tagging results, filtering all names with more than one word, capturing all tokens tagged as ‘Person’ in a row, and building a two-layer tagging system(3-class model and 4-class model) to cross-validation. We use the 4-class model as the main model to capture ‘PERSON’ entities, and check each produced token whether it states in the 3-class model. Now the system we created is able to capture most names from context files, filtering names are irrelevant partially. For example, for the first compiled bio text, we are able to retrieve ‘**Tarek F. Abdelzaher Professor**’ as full name rather than **‘Tarek’**, second one ‘**Sarita V. Adve**‘ as correct name rather than ‘**Sarita V. Adve Richard T. Cheng**’ in the provided file

## Next Steps

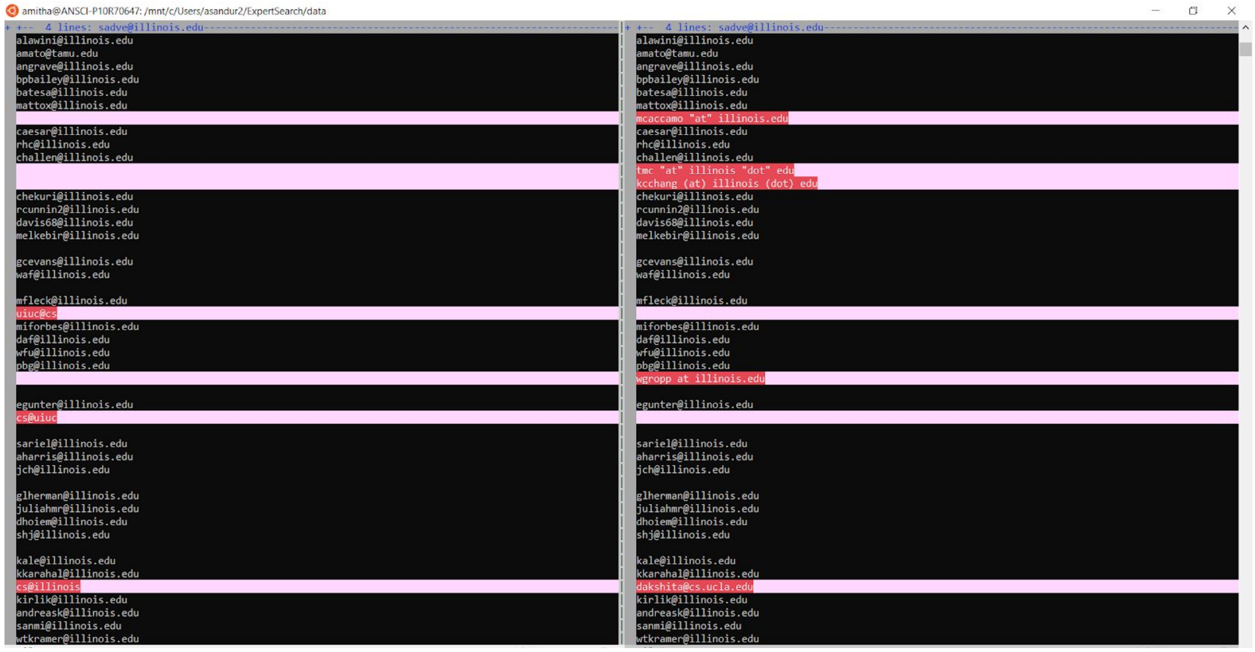
We will continue working on a 2-layer tagging system to cross validation results between two models. Any results show or partial show on both models state they are ‘Person/Name’ entities with high possibility. Additionally, we will continue on filtering results, with introducing a counting/score system. Any name tokens appearing in high frequency should have a high score indicating its the main context of text.

## Challenges

The biggest challenge we are facing is lots of famous names and confusing information shown in the bio files. For example, ‘Ann Arbor’ can be a name or location. The tagging system is not able to distinguish between them. Also ‘Kennedy’ as a famous name, sometimes can occur multiple times in the bio page, the system even with the scoring system is not able to know ‘Kennedy’ is not the main context. We will introduce another non-standard model can tag ‘locations’ entity, then do another cross-validation between our model and the third party model result, it should lead us to a potential solution finding the main person of the context, and better filter out non-relevant names/location.

# Extraction of Professor E-Mail

We have worked on improving the existing regex based extraction of email ids from faculty bios. The original code was extracting the usual format of email ids (user@illinois.edu) and also some false positives such as cs@uiuc. Our new code is able to extract many of the alternative email id formats, for example: "**yang.r.yang at yale.edu**" "**denisew (at) uw.edu**" "**moli96 at uw.edu**". It is also able to capture email ids that were originally missing from the first output and remove some false positives. Below is a comparison of the outputs from existing code shown on the left and output from new code shown on the right.




As observed from the figure above, the new code is able to catch the "user at illinois dot edu" types of format email ids.

We have covered cases for the following email formats:

- user at illinois.edu
- user at illinois dot edu
- user "at" illinois "dot" edu *or* user "at" illinois.edu
- user (at) illinois (dot) edu
- We removed some erroneous outputs like website urls/sentences/repeated special characters

## Next Steps

We plan to convert the email ids in “user at illinois dot edu” format into the normal "user@illinois.edu" format as one of our next steps.

Although the new regex based code is a significant improvement over the existing code, we feel that this method is laborious. And even if we spend a lot more time trying to cover all the exhaustive number of cases, the gain in improvement for the tool as a whole wouldn’t be much, as the number of cases are a lot. Hence we plan to look into machine learning based approaches to extract email ids.

## Challenges

The challenge here is to convert the various different email formats into the usual email id format. Although we may consider keeping it as is if we discover more new email formats along the way and it becomes too heterogeneous, because we have also seen a few websites having email ids in this alternative format.