1. How have you defined a trend? How can you separate it from background noise and/or spurious relationships?

We defined a trend as a word or multi-word phrase that shows increasing relative frequency over time. We use linear regression to calculate the rate of change of relative frequency (slope) and choose those words with the highest slope. By doing this, we focus on words with the biggest increase in relative frequency, which should be separated from background noise and/or spurious relationships.

In LDA models background noise was reduced by joining statistically collocated terms and filtering out the most common words. Here, trends were defined as topic weights normalized across years (eg. the combined topic distribution / year was 1) and then ranked by their linear coefficients to find rising trends.

1. What are the main techniques you have used and how have you tailored them for this problem?

The first main technique we have used is linear regression with relative frequency of words as a function of time. Specifically we used CountVectorizer for preprocessing and frequencies counting. We extracted both unigrams and bigrams, and analyzed both abstracts and full texts. We then normalized the frequency of each word to the length of the article as the relative frequency and calculated the average relative frequency of each word for each year. We conducted a linear regression with average relative frequency of words as a function of time and selected words with highest positive slope. These words show increasing relative frequency over time and thus represent a trend.

Additionally, LDA was used to determine trending topics. It was tailored to this problem in a few ways. The parameters were optimized over the given corpus using grid search and LDA models incorporating time as an additional variable were used.

1. What was your strategy for finding multi-word phrases versus single words?

To start with, we simply set ngram = 2 in CountVectorizer to find bigrams. Then we manually curated bigrams with increasing relative frequency to select bigrams that represent real phrases.

Next, collocated terms were found using NLTK’s bigram and trigram finding algorithms, incorporating their statistical scoring functions to determine collocated words. Later, gensim’s Phrases models were used to conduct a similar approach, joining statistically likely bigrams and trigrams.

1. What approach(es) did you use to separate one subfield from others?

This is a difficult problem. We used LDA to attempt to separate trends into fields. However, with LDA, each document contains a distribution of topics. One approach that was experimented with was assigning a document to belong to the topic that had the highest weight.

1. What parts of the document did you use and why?

We used both abstracts and full texts and integrate hits from both sources. However, we give higher weights to words identified from the abstract. An abstract summarizes the main idea of an article in a concise and accurate manner, and therefore provides much higher fraction of key words that may represent a trend. On the other hand, full texts include more information, which sometimes dilute out the signal of key words that represent a trend.

Using LDA, models were run on all the sections individually as well as the combinations of them combined. However, the final models used the titles concatenated to the abstracts since using the entire texts didn’t seem to have a lot of added benefit. This is unsurprising since the texts were messier and contained extraneous sections like ‘Acknowledgements’, some references that were surely missed in preprocessing, and other junk.

1. How did you normalize the results against the growth of the conference, lengths of documents, etc.?

For the linear regression approach, we normalized the frequency of ngrams to the length of articles and take the average of relative frequency of ngrams for each year. We think normalization to the lengths of articles and averaging relative frequencies of words for each year should minimize confounding effects from the growth of the conference and lengths of documents.

Using LDA, weights of topics were always normalized by year. Additionally, topic models incorporating time were used, eg. LdaSeq and DTM. When calculating n-grams, it was done on a per document basis, hopefully minimizing the skewed article distribution.

1. We know that you can look back and find trends but how would you find the next trend with your method? Be specific.

The words that represent trends all exhibit increasing relative frequency over time, so we should be able to use time series to forecast their future relative frequency. If they still show increasing relative frequency, they probably also represent trends in the (near) future.

The LDA models can be trained on new corpuses, eg. online training, and trends updated accordingly. With both scikit-learn and gensim models there are already builtin methods to fit and update parameters according to new documents. Since this is an iterative algorithm, this is not too hard to do. To find new trends you could update models and look for words/topics whose distributions changed the most.

1. Plot of the final top 10 normalized trends as a function of time.

Please refer the the attached notebook. There are plots showing our trends using relative word counts, multiple LDA implementations, and DTM.

Objectively, as far as the LDA models are concerned, the very last table/plot seems to represent the most coherent topics with subjects like bayesian, latent, stochastic, gaussian\_process, various neural network related fields, and gradient\_descent.