Keyword Extraction for Expertise Modeling

**Problem Statement**

Widespread expansion of the scientific community has led to an increased requirement for a reliable peer-review system, to allow unbiased assessment of new scientific works. To allow such comparison, the expertise of reviewers needs to be modeled in a way that both provides reviewers with the ability to edit their expertise and allows comparison between this expertise and a new work, without knowing the identity of its author.

In this poster, I present experiments conducted for extracting keywords to summarize scientific papers, which are then used to compare reviewer publications with new scientific works. A keyword based model provides the following advantages over a model that uses complete text:

- Authors can add or remove keywords from their expertise to consider or ignore a particular topic
- Keywords can be easily converted to real-valued vectors suitable for comparison using simple vector space distance functions

**Dataset**

- Query papers - Submissions to the uai 2017 conference
- Reviewer archive - Atleast 5 published works of each reviewer in uai 2017 downloaded from dblp
- Ground Truth - Bids placed by reviewers for each submission
- Kp20k dataset [1] used for training seq2seq model for phrase generation

**Evaluation Criteria**

All models evaluated using thresholded recall score

\[
Recall@M = \frac{|V \cap T|}{|V|}
\]

Where V is the set of relevant reviewers and T is the top M reviewers retrieved by the experimental model.

**Experiments**

1. TfIdf value based keywords
2. SAKE - self attention based keywords extractor trained on re-generating input with an autoencoder
3. SAKE - self attention based keywords extractor trained on abstractive keyphrase generation using seq2seq model
4. Words closest to cluster centers of clustered word representations (extracted from SAKE) as keywords

**Results and Conclusion**

- Keywords extracted from tfidf scores have better recall@M than complete text.
- Keywords extracted from Self attention based autoencoder have worse recall@M than tfidf values but better than complete text. It has better precision@M values than both tfidf and complete text.

**Keyword Extraction**

SAKE - The self-attention model [2] provides an attention weight for each input word to a lstm, tuned according to the desired task. For this project, I experimented with two tasks:

- Self attention mechanism trained with an autoencoder on reviewer abstracts
- Self attention mechanism trained with a seq2seq model for abstractive phrase generation

**Paper Representation**

Final representation for a paper was extracted by averaging the dense representation of keywords extracted using our model. Skip-gram model trained on Google News Dataset was used to extract word embeddings.

\[
E_p = \frac{1}{N} \sum_{n=1}^{N} E_{w_n}
\]

Where \(E_{w_n}\) represents the embedding of the kth word, N is the total number of words present in paper P and \(E_p\) is the resultant embedding of the paper.
