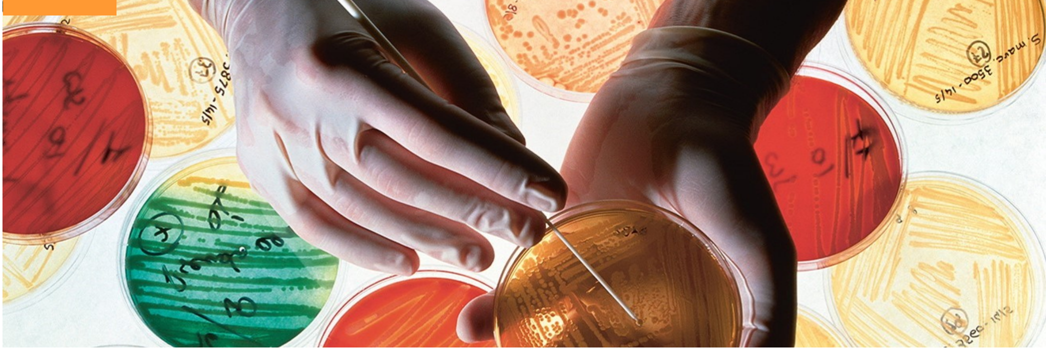

Assigning Grant Applications to Reviewers via Text Analysis

Anne Jorstad, PhD, Data Scientist - 24-25 June, 2020

FNSNF
Swiss National Science Foundation

Overview of SNSF Reviewer-Finding Procedure

Standard process to find reviewers for new grant applications:

- **Humans search web**, via tools such as Google Scholar or Elsevier's Expert Lookup
- A couple hundred applications takes a **couple weeks to process**

Overview of SNSF Reviewer-Finding Procedure

Standard process to find reviewers for new grant applications:

- **Humans search web**, via tools such as Google Scholar or Elsevier's Expert Lookup
- A couple hundred applications takes a **couple weeks to process**

Sometimes we have a **pre-selected pool of reviewers**:

- Want reviewers to be able to **review several applications**
- **(New!)** Download **reviewer publication metadata**, compare to application texts
- **(New!)** Determine the most **similar reviewer-application pairs** automatically

Applications

Spark funding call, March 2020

- "Rapid funding of unconventional ideas"
- All disciplines, often **multidisciplinary**
- 875 applications, 606 reviewers

Coronavirus special funding call, March 2020

- Mostly **biomedicine, public health, economics**
- Vocabulary of all submissions includes "Coronavirus"
- 284 applications, 203 reviewers

Challenge: How to Define Similar Texts

No clear best method.
Ex: NeurIPS (top Machine Learning conference) provides 3 different similarity scores to help Area Chairs find relevant submissions:

Subject Areas		Relevance	TPMS Rank	Affinity Rank
Primary	Secondary			
filter... Clear	filter... Clear	Clear	e.g. < Clear	e.g. < Clear
Deep Learning -> Optimization for Deep Networks	Algorithms -> Missing Data; Deep Learning -> Deep Autoencoders	0.16	3134	7115
Deep Learning -> Analysis and Understanding of Deep Networks	Theory -> Statistical Learning Theory	0.20	165	1376
Deep Learning	Applications -> Sustainability; Deep Learning -> Optimization for Deep Networks; Optimization -> Convex Optimization; Optimization -> Non-Convex Optimization	0.14	1035	1259

Text Data: Grant Applications

- Title
- Keywords
- Abstract
- Main Discipline
- Secondary Disciplines (optional)

Text Data: Reviewers

Scopus: abstract and citation database

- 75 million publication titles (all publishers, peer-reviewed)
- Coverage less good for books, humanities topics, non-English texts

Text Data: Reviewers

Scopus: abstract and citation database

- 75 million publication titles (all publishers, peer-reviewed)
- Coverage less good for books, humanities topics, non-English texts

Download for each reviewer:

- Titles
- Keywords (missing for many publications)
- Abstract (missing for some publications)
- Journal titles
- Subject areas (uses the All Science Journal Classification (ASJC))

Text Similarity Model

Text model options:

- **Raw Texts** (minus stop words): count occurrences of each word
- **TF-IDF** (Term Frequency–Inverse Document Frequency):
  words that appear a lot in one document, but are less frequent in other documents, are more important

Text Similarity Model

Text model options:

- **Raw Texts** (minus stop words): count occurrences of each word
- **TF-IDF** (Term Frequency–Inverse Document Frequency): words that appear a lot in one document, but are less frequent in other documents, are more important

- **Document embeddings** (ex: BERT): future work
- **"Concept extractor"** from Dimensions.ai
  - Database similar to Scopus, but developed by Digital Science and funded by research organizations rather than publishers
  - Input text $\rightarrow$ vector of "concepts"
  - Initial testing results: interesting but not yet useful
  - Future work: collaborate with Dimensions to improve results of this tool

Text Similarity Model

Text model output: term frequency **vector** representation of each text

$S_{a,r}$ = **similarity** between **application** vector a and **reviewer** vector r

$\text{argmax}_r \, S_{a,r}$ = best reviewer for application a

Text Similarity Model

Text model output: term frequency **vector** representation of each text

$S_{a,r}$ = **similarity** between **application** vector a and **reviewer** vector r

$\text{argmax}_r\, S_{a,r}$ = best reviewer for application a

Vector similarity options:

- **Cosine similarity** $\leftarrow$ best results (agrees with literature)
- Correlation
- Others

Text Similarity Model

How to compare these text similarity models?

- No ground truth!
- Test on applications and **manually assigned reviewers** from a previous funding call
- Get **input from domain experts** when found matches were different

Text Similarity Model: Domain Expert Feedback

Raw texts:

- Meaningful with some mistakes
- Different matches: reasonable 67% of the time
- Some reviewers used a lot more than others

Text Similarity Model: Domain Expert Feedback

Raw texts:

- Meaningful with some mistakes
- Different matches: reasonable 67% of the time
- Some reviewers used a lot more than others

TF-IDF:

- Best overlap with known manual assignment
- Different matches: reasonable 90% of the time
- More even distribution of reviewers to applications

Text Similarity Model: Domain Expert Feedback

Raw texts:

- Meaningful with some mistakes
- Different matches: reasonable 67% of the time
- Some reviewers used a lot more than others

TF-IDF:

- Best overlap with known manual assignment
- Different matches: reasonable 90% of the time
- More even distribution of reviewers to applications

Conclusion: Use TF-IDF (agrees with literature)

Text Similarity Model: Further Tweeks based on Feedback

Application texts:

- Repeat discipline names and keywords a couple times

Reviewer texts:

- Limit to first/last author publications
- Limit to past 10 years
- Include mapping from Scopus subject areas to SNSF discipline list (imprecise)

Results: Coronavirus

Biomedical expert feedback:

"The examples I checked were **really good matches**! Very balanced with regard to the research topics covered in the grant proposal abstract. The underlying algorithm and keyword criteria for the biomedical part are very suitable."

Social Sciences and Humanities (SSH) expert feedback:

"We were **not that impressed** with the results of the automatic matching, the method does not seem to work in favour of our disciplines."

Results: Coronavirus SSH example

"Crisis Management of the Covid 19 Epidemic in coercive contexts"

- **discipline:** social work
- **keywords:** settings for juveniles; coercive settings; therapeutic settings; prisons; health in prisons

Reviewer #1: *good match*

- **top disciplines:** health; infectious diseases
- **top keywords:** health workers; human resources for health; post-conflict; infectious diseases

Reviewer #2: *bad match*

- **top disciplines:** immunology/immunopathology; biochemistry
- **top keywords:** hiv; antiretroviral therapy; china; sexual transmission

Results: Coronavirus SSH example

"Crisis Management of the Covid 19 Epidemic in coercive contexts"

Misleading terms that lead to the bad match:

- therapeutic/therapy
- China
- HIV (in the context of vulnerable groups)

Results: Coronavirus SSH example

Conclusion:

- **Biomedical terminology easily interferes with SSH topics.**
- SSH texts include less specific vocabulary, so are harder to match than the precise vocabulary found in biomedical texts.

Modified algorithm:

- Biomedical reviewers $\nrightarrow$ SSH applications ✗
- SSH reviewers $\rightarrow$ biomedical applications ✓ (sometimes there is an SSH component)

Results: Spark Multidisciplinary Example

*"Using **Deep Neural Networks** to Bridge Clinical and Quantitative Analysis of Intracranial **EEG** in **Epilepsy**"*

Ideally: Assign one reviewer for each main topic

Reality: Good luck!

- For now, reviewer assignment must be manually adjusted

Future work:

- Detect main topics
- Require representative reviewers from each

Balancing Number of Applications per Reviewer

Constraints:

- max A applications per reviewer (usually 5-10)
- exactly R reviewers per application (usually 2-3)

Linear programming objective function:

$$\max_{M} \sum_{a,r} S_{a,r} M_{a,r}$$

$S_{a,r} =$ similarity between application a and reviewer r
$M_{a,r} = 1$ if application a matched with reviewer r, else 0

Balancing Number of Applications per Reviewer

Output of algorithm is two lists:

- Top R reviewers per application, no more than A applications per reviewer
- Top 5 reviewers per application, ignoring A

$\rightarrow$ Domain experts at the SNSF can more **efficiently and effectively assign reviewers**.

Conclusion

"You saved us weeks of work!"

"A **manual check** of the attributions was **still important** and we were able to optimize, given both lists."

We will continue our internal development, since currently available commercial tools cannot be sufficiently tailored to meet our needs.

Note: this technique is not applied to most grant applications at the SNSF - only when a pool of reviewers is known in advance.

Thank you to all SNSF collaborators who provided feedback for this work!