SciSearch: Query-by-Example for Scientific Article Retrieval

Ashutosh Kumar Singh
19CS30008
ashu11.03.2002@gmail.com

Ashwamegh Rathore
19CS30009
ashwameghrathorejsr@gmail.com

Nakul Aggarwal
19CS10044
nnakul.aggarwal@gmail.com

Suhas Jain
19CS30048
suhasjain142@gmail.com

1 Problem Statement

In research, it is naturally difficult to express information requirements as simple keyword queries because they can have a very broad meaning and we might not be able to retrieve a specific section from the whole bundle of documents. A far more effective way to understand the needs of the user is by asking them to give a research paper deemed as relevant to their aspirations. Hence, we aim to envision and develop a model that is able to retrieve scientific papers analogous to a query scientific paper, along specifically chosen rhetorical structure elements (facets/aspects), like background/objective, method and results. So we frame our task as one of retrieving scientific papers given a *query paper* and additional information indicating the *query facet*. We want to give any researcher the freedom to ask – *"I came across the paper XYZ during my research and am extremely inclined towards the results and mathematical background given by the author. Can you please get me some more papers that I might find relevant to my research?"*.

2 Motivation Behind the Proposed Solution

Most of the works till now use cosine similarity or L2 distance based measures, which fail to capture the semantic relatedness between the query and a document. We aim to capture the semantic similarity while returning and ranking results.

Mysore et al. (2021) show the inability of models to determine similarity between domain specific technical concepts. They show an example for the inability to rate "stacking", "ensemble strategy", and "bagging" as similar. However, in the context of papers in the domain of computer science, it is very important to treat such similar concepts as related. Hence, we aim to incorporate this too in our solution.

Nearly all methods perform poorly in the case of determining mechanistic similarity in *method* facets. This often relies on determining similarity across a sequence of actions. We aim to address this issue as well.

3 Related Work

Discovering Relevant Scientific Literature. Given a small set of papers Q that they refer to as the query set, El-Arini and Guestrin (2011) seek to return a set A of additional papers that are related to the concept defined by the query. Intuitively, a paper that cites all of the articles in Q is likely to represent related research. Likewise, a paper that is cited by every article in Q might contain relevant background information. They quantify the inference relation among the papers in the form of a weighted directed acyclic graph, and propose various statistical techniques to sample, sort and diversify the retrieved set of relevant documents.

While choosing candidates for a specific query paper, they only consider cited papers and other papers with the same authors, which may lead to missing out on several highly related and relevant papers, which belong to other authors, or may not have been cited. However, we construct the candidate pools from about 800, 000 computer science papers in the S2ORC Corpus (Lo et al., 2020) using a diverse set of retrieval methods, which helps us to capture a much wider range of candidate papers. Also, they do not capture the aspect, with respect to which we want to retrieve similar papers.

Aspect-based Academic Search using Domain-specific KB. Upadhyay et al. (2020) define and solve a novel academic search task, called aspect-based retrieval, which allows the user to specify the aspect along with the query to retrieve a ranked list of relevant documents. Their primary idea is to estimate a language model for the aspect as well as the query using a *domain-specific knowledge base* and use a mixture of the two to determine the relevance of the article.

They assume that users express their information need as strings of words called queries, e.g., *Evaluating the Performance of Dynamic Database Applications*. However, we consider the query to be the entire abstract of the paper. We also differ in the consideration of types of facets.

Also, the most novel idea in our work is that we create an ensemble retrieval engine after using a neural network to learn the semantic relatedness between two documents depending upon a facet.

4 Dataset Description

For all the purposes in the project like training, testing, validation and interface development we have used the CSFCube dataset (Mysore et al., 2021). The dataset relies on the Semantic Scholar Open Research Corpus (S2ORC) (Lo et al., 2020) from which 800,000 computer science papers were sourced out of total corpus of 81.1M English language research papers. It currently only has the abstract and the title as steps of calculating embeddings and semantic matching will become computationally very expensive for the whole body. The dataset has the following features:

- **Facet**: for a research paper corresponds to the dominant steps involved in carrying out scientific research. These facets are broadly defined as:
  - *background / objective*: Most often sets up the motivation for the work, states how it relates to prior work and states the problem or research question being asked.
  - *method*: Describes the method being proposed or used in the paper. The method could be described at a very high level or it might be specified at a very fine-grained level depending on the type of paper.
  - *result*: This may be a detailed statement of the findings of analysis, a statement of results or a concluding set of statements based on the type of paper.

- **Query Abstract Selection**: There are a total of 50 query abstract-facet pairs. Out of theses 50, 16 abstracts have two different facets each (total of 32 query abstract-facet pairs). The remaining 18 abstracts have a single query facet each. In total, our dataset contains 16 background queries, 17 method queries, and 17 result queries. See the tables for more details:

Table 1: Statistics for the test collection

Statistic		All
Query abstract-facet pairs	-	50
Unique query abstracts	-	34
Mean candidate pool size	-	124.9
Query-candidate pairs	-	6244
Candidates rated +1 per query	min	12
	max	87
	avg	36.9
Candidates rated +2/+3 per query	min	1
	max	35
	avg	9.8

5 Technique and Experiments

5.1 Relevance Grades

The information retrieval model we are formulating should derive the papers most relevant to an example paper. One solution to deal with the subjective notion of relevance is to retrieve papers similar on the basis of a particular facet. Another solution is to define several grades of relevance, rather than hard 0/1 labels, that are able to capture the magnitude of differences or similarities more finely.

We define 4 grades of relevance – 3 (near identical), 2 (similar), 1 (related), 0 (unrelated). Note that we have replaced a single hard relevant label with 3 labels reflecting varying degrees of relevance.

5.2 Architecture

Ranking is nothing more than sorting the candidates based on the relevance scores. The larger question is how we learn these relevance scores. For a facet-based query-by-example model, we need an architecture that can learn and predict the relevance score (0-3) between two papers for a particular facet.

The design of such an architecture consists of three component-models that are cascaded one after the other. Each of these three models are responsible for different tasks.

- **Feature Extractor (FE)**: It converts the text-based papers in human-readable form to vectors of real numbers that encode the meaning of a particular facet of the paper such that the features that are closer in the vector space

are expected to be similar in semantics and information.

We use a handful of the state-of-the-art neural networks, pre-trained on huge data, for extracting features (or embeddings) from the different facets of the papers.

- Unsupervised Simple Contrastive Learning of Sentence Embeddings (*UnSimCSE*) (Gao et al., 2021a)
- Supervised Simple Contrastive Learning of Sentence Embeddings (*SuSimCSE*) (Gao et al., 2021a)
- Cased BERT Model for Scientific Text (*SciBERT-cased*) (Beltagy et al., 2019)
- Uncased BERT Model for Scientific Text (*SciBERT-uncased*) (Beltagy et al., 2019)
- BERT Model with Natural Language Interface (*BERT-NLI*) (Gao et al., 2021b)
- BERT Model for Paraphrase (*BERT-PP*) (Arase and Tsujii, 2021)
- Scientific Paper Embeddings using Citation-informed Transformers (*SPECTER*) (Cohan et al., 2020)

- **Fine-Tuning Network (FTNN)**: Though all these baseline are very popularly used in all the downstream tasks, it is better to slightly tweak the knowledge gained by these pre-trained models before applying to a new problem that might not necessarily be exactly the same as the actual problem they were designed for. *FTNN* is responsible for this transfer of learning from one domain to the other. Considering that the pre-trained feature extractors are intelligent and complex enough to yield good representations, even for a different domain, we use a simple fine-tuning neural network that comprises two fully-connected linear layers with linear bias and *TanH* activation.

- **Semantic Relevance Predictor (SRNN)**: This component learns the relevance score (0-3) between any two representations. Instead of solely using a non-flexible metric of dot-product to gauge the relevance between two vectors, it is better to use that as a component of an intelligent model. *SRNN* maps the dot-product to a 4-dimensional vector of logits by passing it through a linear layer. These logits are then softmax-ed in order to obtain a probability distribution over the scores 0-3. SRNN is held responsible for learning the intervals or margins on the real-axis (co-domain of the dot-product function) that stand representative to a specific relevance grade. Read (Kiros et al., 2015) for more in-depth information.

5.3 Loss Function

5.3.1 Negative Log Likelihood Loss

Since the overall model aims at classifying a query-candidate pair into one of three relevance classes, it is intuitive to use an NLL (negative log-likelihood) loss function. So in the first prototype of the architecture, we use the NLL-loss function. We got significant improvements on the baselines using this loss function , but while using this function we are still treating the problem like a normal multi-class classification problem, which it is not. For a query-candidate pair, the dataset defines a unique score and hence the other three scores will be counted as wrong predictions. So for a (*Q, D*) pair with score 2, the score of 3 will be treated as wrong as the score of 0. So, NLL-loss enables learning only the hard 0/1/2/3 labels, with no tolerance whatsoever. This might not affect the accuracy, but will have a significant impact on the *NDCG* values.

5.3.2 Kullback-Leibler Divergence Loss

The goal is to somehow tell the model that when the ground truth similarity score of a query-document pair is let's say 2, predicting the score as 3 or 1 is much better than predicting the score as 0. To do this we define a new loss function, here we do not use the ground truth like we used in NLL-loss, but we consider a Gaussian probability distribution centered around the ground truth score (see Fig. 1). From the soft-max layer of our model we get probabilities of the 4 possible similarity scores. We extrapolate these probabilities to find a suitable fit Gaussian probability distribution, this acts as a proxy for our predicted value (see Fig. 2). Now, compared to NLL-loss where we calculated loss as a function of predicted and actual labels, we now define loss as a function of difference between these 2 probability distributions. For finding the loss we use KL divergence score (Kullback and Leibler, 1951), it quantifies how much one probability distribution differs from another probability distribution. The intuition for the KL divergence score is that when the probability for an event from P is large, but the probability for the same event

in Q is small, there is a large divergence. When the probability from P is small and the probability from Q is large, there is also a large divergence, but not as large as the first case.

5.4 Training

The gradients with respect to the objective loss function back-propagate through *SRNN* upto *FTNN*. A total of 2x7x3 models were trained; with an *ADAM* optimizer, (initial) learning rate of 0.01, batch size of 16 and an early-stopping tolerance of 4 epochs.

In the original dataset, the ranking of the candidate papers with the query papers is given, along with their respective relevance scores. The ranked results for the pool of queries are not directly useful for training our model. They are translated into a collection of triplets – query paper representation (for a facet X), candidate paper representation (for the facet X), relevance score (0-3).

The distribution of the triplets with respect to the relevance score was found to be highly imbalanced. Triplets with high relevance scores had much less samples than those with lower scores. *SMOTE* technique was used to generate synthetic samples in order to balance the imbalanced dataset.

6 Results and Analysis

6.1 Improvement Over Baselines

Table 2 shows the baseline metrics for various models mentioned in Mysore et al. (2021). Table 3 shows the same metrics for several models with NLLLoss as the loss function. Finally, Table 4 shows the metrics with KLDivLoss as the loss function.

Overall, there is a significant improvement in precision, recall and NDCG with both the loss functions - NLLLoss and KLDivLoss, compared to the baselines mentioned in Mysore et al. (2021). The most prominent improvement is for the *method* facet, which indicates that the neural network effectively captures domain specific similarities. The highest increase is in recall, which indicates that we now retrieve a greater fraction of the total relevant documents. The increase in NDCG shows that our scheme also produces better rankings as compared to the baselines.

With NLLLoss as the loss function, the best performing models are: SciBERT-cased for the *background* and *method* facets, and SPECTER for the *result* facet.

With KLDivLoss as the loss function, we get the best results, and the best performing models are: SciBERT-cased for the *background* facet, and SPECTER for the *method* and *result* facets.

6.2 Specific Examples

We also take up some specific examples from Mysore et al. (2021) and show how our model overcomes all these errors in the earlier techniques. These are mentioned in Appendix Sec. D.

7 Future Thoughts

Our current focus has been on devising a strategy to effectively query research papers and improve upon the baselines mentioned in Mysore et al. (2021), and eliminate the factors which cause the baseline methods to underperform. There are still many improvements and ideas that we have thought of which can lead to performance and functionality improvements. They are as follows:

- Before matching a query with a candidate paper and calculating the similarity we need to make a candidate pool out of numerous papers that exist in the S2ORC corpus (Lo et al., 2020), as calculating similarity for all pairs is computationally infeasible. We want to devise a strategy that makes a relatively small candidate pool in very short time with high recall.

- In the dataset we noticed that there are a lot of special unicode characters and latex snippets which act as noise when we feed the text into the pretrained models we are using. We aim to either remove these or replace them with something meaningful in a pre-processing step in the future.

- Presently, we operate at the level of abstracts rather than the body text of papers under the understanding that salient information about a paper is contained in the abstract. However, in the future one may think of incorporating the full body text as well, without being computationally slow.

- Faceted similarities as labelled here often also show context dependence on other facets. This is notable in the case of result queries. Determining the result similarity may a time relies heavily on modeling method similarity. We believe approaches which improve upon

method similarity, will likely benefit overall performance on other facets as well.

Acknowledgements

We would like to express our sincerest gratitude to our guide, Dr. Somak Aditya, for his guidance and support, without which this project would not have been successful. We also want to extend our heartfelt thanks to Ankan Mullick for his ideas and suggestions, all of which had a huge impact in the way the work proceeded.

References

Yuki Arase and Junichi Tsujii. 2021. Transfer fine-tuning of bert with phrasal paraphrases. *Computer Speech Language*, 66:101164.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S. Weld. 2020. Specter: Document-level representation learning using citation-informed transformers.

Khalid El-Arini and Carlos Guestrin. 2011. Beyond keyword search: Discovering relevant scientific literature. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, page 439–447, New York, NY, USA. Association for Computing Machinery.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021a. Simcse: Simple contrastive learning of sentence embeddings.

Yang Gao, Nicolo Colombo, and Wei Wang. 2021b. Adapting by pruning: A case study on bert.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-thought vectors.

S. Kullback and R. A. Leibler. 1951. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79 – 86.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. 2020. S2ORC: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983, Online. Association for Computational Linguistics.

Sheshera Mysore, Tim O'Gorman, Andrew McCallum, and Hamed Zamani. 2021. Csfcube – a test collection of computer science research articles for faceted query by example.

Prajna Upadhyay, Srikanta Bedathur, Tanmoy Chakraborty, and Maya Ramanath. 2020. Aspect-based academic search using domain-specific kb.

Appendix

A Figures related to KLDiv-Loss

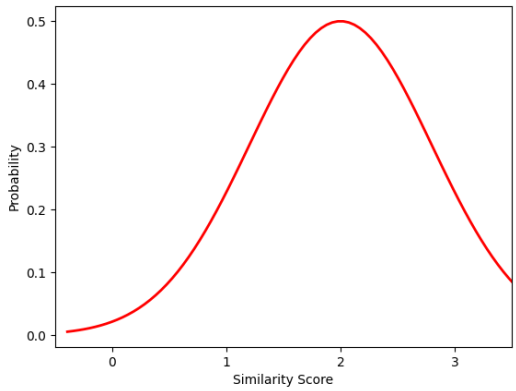

Figure 1: Probability Distribution About Predicted Label

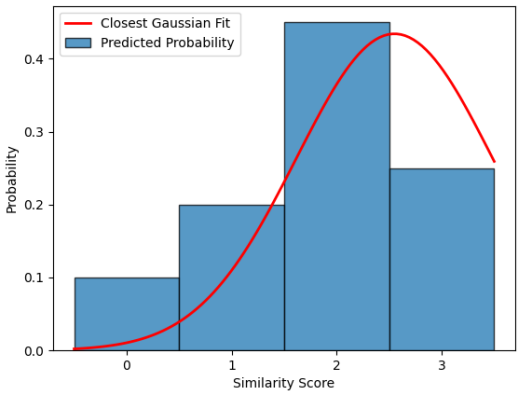

Figure 2: Estimated Gaussian Distribution on Predicted Probabilities

B Interactive GUI

To demonstrate the functionality of our model and the vision, we have we have made a a graphical interface for SciSearch. It is currently hosted at: share.streamlit.io/suhas4122/scisearch/main/SciSearch.py. Our goal while making SciSearch was to make an application which is user friendly yet demonstrates all the aspects of our research. We have hosted SciSearch to make sure it is accessible to everyone across devices. As we demonstrated earlier (refer to section 5) we experimented with different pre-trained models and loss functions. For different scenarios, different combinations of model and loss-function give the best result. Facet, pre-trained model and loss-function can be selected as the user wants after which the applications gives a list of available queries in the CSFCube dataset. A query along with desired number of results can be selected to get ranked list of research papers along with their abstracts.
Currently, we can only query on the limited number of facet wise query documents present in CSFCube datset. It is not a general purpose query tool but we aim to make such a tool in the near future. Our focus has been on to validate our methodologies and give users a taste of what is about to come.

C Tabulated Results

Please refer to next page.

Table 2: Baseline metrics from Mysore et al. (2021)

	Background					*Method*				
	RP	P@20	R@20	NDCG$_{\%100}$	NDCG$_{\%20}$	RP	P@20	R@20	NDCG$_{\%100}$	NDCG$_{\%20}$
SentBERT-NLI	19.02	25.00	40.13	75.80	54.23	09.11	11.46	02.89	58.52	31.10
SentBERT-PP	21.24	28.75	46.67	79.14	60.80	10.00	10.83	36.30	59.50	33.40
UnSimCSE-BERT	18.15	23.44	36.05	74.34	51.59	08.86	09.65	27.92	59.21	31.23
SuSimCSE-BERT	19.22	22.81	46.75	76.70	55.22	08.58	09.76	29.01	58.54	30.88
SPECTER	**24.81**	**35.31**	**57.45**	**82.24**	**66.70**	**11.72**	**13.58**	**40.81**	**62.77**	**37.41**
	Result					*Aggregated*				
	RP	P@20	R@20	NDCG$_{\%100}$	NDCG$_{\%20}$	RP	P@20	R@20	NDCG$_{\%100}$	NDCG$_{\%20}$
SentBERT-NLI	14.23	22.05	46.99	72.13	51.30	14.04	19.42	38.67	68.68	45.39
SentBERT-PP	13.60	19.83	41.73	71.90	52.35	14.83	19.62	41.41	69.98	48.57
UnSimCSE-BERT	12.00	19.58	38.95	68.44	45.55	12.92	17.41	34.43	67.17	42.59
SuSimCSE-BERT	12.37	18.58	39.76	68.78	44.93	13.33	16.95	34.83	67.83	43.45
SPECTER	**18.62**	**23.78**	**52.72**	**75.47**	**56.67**	**18.29**	**23.97**	**50.14**	**73.30**	**53.28**

Table 3: Metrics with NLLLoss as the loss function

	Background					*Method*				
	RP	P@20	R@20	NDCG$_{\%100}$	NDCG$_{\%20}$	RP	P@20	R@20	NDCG$_{\%100}$	NDCG$_{\%20}$
SentBERT-NLI	31.07	26.56	43.05	82.38	65.79	40.90	28.24	81.90	74.20	56.00
SentBERT-PP	50.49	37.81	65.27	81.06	64.44	47.90	**32.35**	**96.78**	75.90	61.37
SciBERT-cased	60.43	50.31	81.34	**91.34**	**81.72**	**61.80**	23.53	66.84	**82.51**	**66.23**
SciBERT-uncased	**62.54**	49.06	79.41	87.13	75.57	27.10	22.06	72.76	66.22	43.77
UnSimCSE-BERT	47.94	40.62	66.60	85.37	71.27	42.54	29.41	91.32	79.88	65.07
SuSimCSE-BERT	29.71	30.00	48.68	72.70	50.54	34.52	27.65	87.78	69.79	48.35
SPECTER	59.18	**51.87**	**84.41**	84.36	73.41	26.05	22.06	64.32	64.89	39.48
	Result					*Aggregated*				
	RP	P@20	R@20	NDCG$_{\%100}$	NDCG$_{\%20}$	RP	P@20	R@20	NDCG$_{\%100}$	NDCG$_{\%20}$
SentBERT-NLI	34.67	35.88	83.42	70.10	54.98	35.64	30.30	69.98	75.43	58.78
SentBERT-PP	44.71	37.65	85.95	76.72	59.70	47.65	35.90	**83.01**	77.83	61.78
SciBERT-cased	50.18	33.53	79.84	78.51	60.83	**57.41**	35.50	75.90	**83.98**	**69.35**
SciBERT-uncased	58.32	40.00	84.58	82.67	69.35	49.06	36.80	78.91	78.51	62.64
UnSimCSE-BERT	37.82	36.18	83.05	72.62	57.59	42.67	35.30	80.60	79.17	64.51
SuSimCSE-BERT	60.56	42.06	93.44	86.18	76.31	41.84	33.30	77.19	76.29	58.56
SPECTER	**68.93**	**44.41**	**93.89**	**89.44**	**80.97**	51.23	**39.20**	80.80	79.47	64.45

Table 4: Metrics with KLDivLoss as the loss function

	Background					*Method*				
	RP	P@20	R@20	NDCG$_{\%100}$	NDCG$_{\%20}$	RP	P@20	R@20	NDCG$_{\%100}$	NDCG$_{\%20}$
SentBERT-NLI	**66.54**	50.63	81.61	91.06	82.95	61.17	32.65	97.62	82.97	69.03
SentBERT-PP	60.45	50.00	82.78	89.67	79.26	59.87	32.65	97.51	82.16	68.35
SciBERT-cased	65.34	**51.56**	**84.14**	**92.55**	**84.79**	82.32	33.82	99.16	88.84	79.24
SciBERT-uncased	58.12	46.25	76.13	89.22	77.69	66.21	31.47	92.65	85.57	71.79
UnSimCSE-BERT	61.61	48.12	80.15	89.34	80.24	56.37	31.18	94.47	79.29	63.20
SuSimCSE-BERT	62.06	48.44	77.43	88.05	76.81	55.25	31.76	95.20	79.05	65.56
SPECTER	64.82	50.94	82.50	89.13	79.87	**84.07**	**33.82**	**99.58**	**90.49**	**81.43**
	Result					*Aggregated*				
	RP	P@20	R@20	NDCG$_{\%100}$	NDCG$_{\%20}$	RP	P@20	R@20	NDCG$_{\%100}$	NDCG$_{\%20}$
SentBERT-NLI	77.49	45.00	97.11	90.96	81.93	68.44	42.60	92.32	88.28	77.87
SentBERT-PP	57.59	42.35	93.83	86.43	76.26	59.28	41.50	91.55	86.02	74.53
SciBERT-cased	71.69	45.00	95.55	86.56	76.90	73.27	43.30	93.13	89.25	80.22
SciBERT-uncased	71.84	44.41	96.65	88.34	77.98	65.54	40.60	88.72	87.68	75.78
UnSimCSE-BERT	71.49	44.12	94.31	88.19	77.98	63.19	41.00	89.83	85.53	73.68
SuSimCSE-BERT	68.82	42.94	93.54	87.60	76.52	62.05	40.90	88.95	84.84	72.89
SPECTER	**78.22**	**45.88**	**98.20**	**92.75**	**84.58**	**75.92**	**43.40**	**93.65**	**90.82**	**82.00**

D Specific Examples

Salient Aspects: One source of error was the inability of models to identify the most salient aspects for similarity, often expressed only in part of a larger set of facet sentences.

> BACKGROUND Q: "Many classification problems require decisions among a large number of competing classes."[1791179]
>
> DOCUMENT: "Several real problems involve the classification of data into categories or classes."[12156882]

Earlier, this document was a false positive, however, now, it falls into the true negative class.

Multiple Aspects: Within a given facet, papers often express multiple finer grained aspects, the earlier techniques however often only retrieved based on a single aspect.

> METHOD Q: "We present a Few-Shot Relation Classification Dataset (FewRel), . . . The relation of each sentence is first recognized by distant supervision methods, and then filtered by crowdworkers. We adapt the most recent state-of-the-art few-shot learning methods for relation classification and conduct a thorough evaluation of these methods."[53080736]

In this example, baseline models often retrieved based on one or the other aspect, however our technique mitigates this difficulty.

Domain Specific Similarities: Another source of error was the inability of models to determine similarity between technical concepts. For example, consider the terms "stacking", "ensemble strategy", and "bagging".

> RESULT Q: "Using a public corpus, we show that stacking can improve the efficiency of automatically induced anti-spam filters, . . . "[3264891]
>
> DOCUMENT: "The experiments on standard WEBSPAM-UK2006 benchmark showed that the ensemble strategy can improve the web spam detection performance effectively."[40151482]
>
> DOCUMENT: "We evaluate the classifier performances and find that BAGGING performs the best. . . . our method may be an excellent means to classify spam emails"[15748075]

Both these documents were earlier false negatives, however now, using our technique, they are classified as true positives.

Mechanistic Similarities: Nearly all methods perform poorly in the case of determining mechanistic similarity in *method* facets. This often relies on determining similarity across a sequence of actions. Baseline models failed to align steps [1] and [2] across abstracts below.

> METHOD Q: "Using an annotated set of "factual" and "feeling" debate forum posts, [1]we extract patterns that are highly correlated with factual and emotional arguments, and [2]then apply a bootstrapping methodology to find new patterns in a larger pool of unannotated forum posts."[10010426]
>
> DOCUMENT: "[1]High-precision classifiers label unannotated data to automatically create a large training set, which is then given to an extraction pattern learning algorithm. [2]The learned patterns are then used to identify more subjective sentences."[6541910]

Earlier this document was a false negative, however now it is a true positive.

Context Dependence of Facets: Faceted similarities as labelled here often also show context dependence on other facets. This is notable in the case of *result* queries. Given that one major guideline for result similarity in our dataset are if "the same finding or conclusion" is found, being able to determine context similarity is important.

> RESULT Q: ". . . Subsequently, lexical cue proportions, predicted certainty, as well as their time course characteristics are used to compute veracity for each rumor tweet . . . . Evaluated on the data portion for which hand-labeled examples were available, it achieves .74 F1-score on identifying rumor resolving tweets and .76 F1-score on predicting if a rumor is resolved as true or false.[5052952]
>
> DOCUMENT: "In this study, we propose a novel approach to capture the temporal characteristics of these features based on the time series of rumor's lifecycle, for which time series modeling technique is applied to incorporate various social context information. Our experiments using the events in two microblog datasets confirm that the method outperforms state-of-the-art rumor detection approaches by large margins."[17025981]

Earlier this document was a false negative, however now it is a true positive.

E Breakup of Labour

Individual	Roll No.	Responsibilities
Ashutosh Kumar Singh	19CS30008	• Preliminary research and literature review • Parsing and preprocessing dataset - proposing a suitable conversion format and writing code for the same • Coding the methods to extract embeddings for all queries and documents for each facet and each pretrained model • Coding functions for dataset utlities required for training • Writing code for the plotting utilities • Writing the report - the sections on problem statement, related work, dataset, results and analysis and appendix
Ashwamegh Rathore	19CS30009	• Preliminary research and literature review • Deciding a suitable format for storing document and query embeddings • Coding the functions for evaluation metrics - individually for each facet and aggregated • Training the model and testing it to generate results, metrics and plots • Writing the backend code, integrating with frontend, and deploying the website • Organizing code for submission, listing all requirements and writing README.md

Nakul Aggarwal	19CS10044	• Preliminary research and literature review • Proposing the idea of a neural network to capture semantic relatedness with NLLLoss as the loss function • Coding the neural network with NLLLoss as the loss function • Coding training utilities • Writing the techniques and experiments section in the report • Creating the demo video
Suhas Jain	19CS30048	• Preliminary research and literature review • Proposing the idea of using KLDivLoss as the loss function • Coding the neural network with KLDivLoss as the loss function • Designing the graphical user interface and coding the frontend • Adding tables, images and plots to the report • Designing and creating the presentation