ECE1786 Project Final Report

EZPaperSearch
Paper Recommendation Application

Yuhan Wei 1000893161
Evan Lyu 1002058024

word count = 1984
penalty = 0

Permission

	Yuhan Wei	Evan Yunze Lyu
Permission to post video	Yes	Yes
Permission to post final Report	Yes	Yes
Permission to post source code	No	No

1   Introduction

Paper searching can be very time consuming and tiring, especially when looking for papers on a specific topic. The project aims to build an effective and efficient paper recommendation application, EZPaperSearch, that helps users find papers they might be interested in without having to spend hours reading through dozens of articles. The paper recommendation application takes in the title and abstract of the context paper from the user and outputs a list of related articles to the user as recommendations (Figure 1).
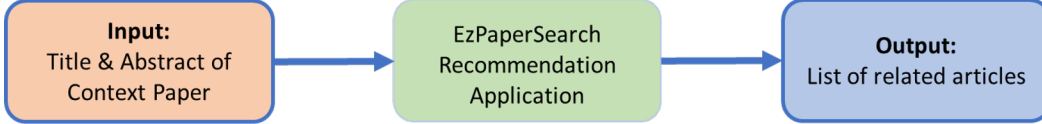
Figure 1: Input and Output of EZPaperSearch Application

2   Background and Related Work

To achieve the task, we will be measuring the similarity of papers. The most intuitive method to evaluate the similarity of two documents is comparing them word by word. Rada et al. [1]comprehensively evaluated various corpus-based or knowledge-based word semantic similarity measures, such as Pointwise Mutual Information [2], Latent Semantic Analysis [3], Leacock & Chodorow similarity [4] and Lesk similarity [5]. The best F1 score(81.3%) was achieved by combining several methods into one with simple average while the vector-based similarity baseline using TF-IDF weights achieved 75.3% F1 score. However, this method relies heavily on external resources: corpus and knowledge. In addition, it ignores the connection and semantics between words.

Recently, with the explosion of deep learning techniques, more and more researchers designed siamese deep learning networks in the semantic text match area. Jyun Yu et al. proposed a Siamese multi-depth attention-based hierarchical recurrent neural network (SMASH RNN [6]. SMASH RNN is designed to tackle the challenge of extracting the semantics of individual words, phrases and sentences from complex document structures. It incorporates two MASH RNNs and fuses the output using a siamese structure and is shown to be effective in measuring the semantic similarity between two long-form documents. Thus, the SMASH RNN model has the potential for application in related article recommendations where precise long-form semantics analysis is essential. Extensive research also shows that SMASH RNN is superior (82.19% accuracy) in related article recommendations compared to other methods such as regular RNN, CNN, and DeepQA.

Since 2018, the emergence of large pre-trained transformers has significantly change the natural language processing area. BERT [7], RoBERTa [8] and GPT2 [9]have been successfully applied in the task of semantic textual similarity, and achieved the state of art performance [10]. In 2019, Nils Reimers and Iryna Gurevych designed a Siamese BERT network: Sentence-BERT to derive meaningful text embeddings from the input [10]. This structure reduced a high computational overhead required by

a single pre-trained transformer network. Sentence-BERT achieved around 87.79% accuracy compared to single BERT network whose accuracy was 84.94% with SentEval toolkit [11].

3   Data Processing

The dataset used for this project is the open-source S2ORC dataset [12], which covers 136m+ scientific papers from 20 fields of study. This dataset is sufficiently large and contains all the necessary information required for training our model: the title, abstract, and references used in the papers as well as the field of study.

Considering the wide range and the huge amount of papers in our dataset, we decided to concentrate on papers in one field of study: Computer Science. We downloaded 1,795,148 paper entries and performed data cleaning and labeling to train our model.

3.1   Data Cleaning

- Load data from Json file and extract columns containing useful information: 'id', 'title', 'paperAbstract', 'Out-Citations', 'fieldsOfStudy'.

- Remove data points with missing information.

- Remove non-English papers.

The cleaned dataset contains 224,394 paper entries.

3.2   Data Labeling

The models take inputs as pairs of papers (described in more detail in section 4.0). The following steps showed how we constructed our data samples and how we labeled them.

- Generate all possible unique paper pairs among all papers

- For each paper pair, calculate the similarity of citations using Jaccard similarity.The reference similarity is used as a metric for measuring the relatedness of papers in each paper pair [6]

$$Jaccard(U, V) = \frac{|U \cap V|}{|U \cup V|}$$

- Label positive samples whose similarity is greater than 0.5.

- Randomly select an equal number of pairs whose similarity is 0 as negative samples. Figure 2 shows two examples of positive paper pair and negative paper pair. The label of the data looks reasonable to us.

- Split the dataset into training, validation, and testing sets

| | | | | | 1 entry Filter |
index	title1	abstract1	title2	abstract2	out_similarity
204	Excitation, observation, and ELF-MD: optimization criteria for high quality test sets	In previous work, we have shown that optimizing the number of site observations leads to more defect detection. However, for increasingly difficult defects, optimizing patterns for balanced random excitation also enhances test effectiveness. We can also reduce the effect of undetected defects by choosing tests that minimize the likelihood of field failures.	Balanced excitation and its effect on the fortuitous detection of dynamic defects	Dynamic defects are less likely to be fortuitously detected than static defects because they have more stringent detection requirements. We show that (in addition to more site observations) balanced excitation is essential for detection of these defects, and we present ametric for estimating this degree of balance. We also show that excitation balance correlates with the parameter ¿ in the MPG-D defective part level model.	0.89

| | | | | | 1 entry Filter |
index	title1	abstract1	title2	abstract2	out_similarity
18	Alternate hammering test for application-specific DRAMs and an industrial case study	This paper presents a novel memory test algorithm, named alternate hammering test, to detect the pairwise word-line hammering faults for application-specific DRAMs. Unlike previous hammering tests, which require excessively long test time, the alternate hammering test is designed scalable to industrial DRAM arrays by considering the array layout for potential fault sites and the highest DRAM-access frequency in real system applications. The effectiveness and efficiency of the proposed alternate hammering test are validated through the test application to an eDRAM macro embedded in a storage-application SoC.	Efficient 1064 nm conversion to the eye-safe region using an external cavity diamond Raman laser	We present a 1064 nm pumped diamond Raman laser operating with 71% quantum conversion efficiency to the 2nd Stokes wavelength of 1485 nm. Up to 1.6 W average power is produced with excellent beam quality.	0.0

Figure 2: Examples of paper pair

Table 1: Data splitting

	Training	Validation	Testing
Split Ratio	0.7	0.15	0.15
Sample Count	1160	248	248

4   Baseline model

The baseline model is a TFIDF-based model that measures document similarity by evaluating how relevant a word is to a document in a collection of documents. The model takes into consideration two factors: term frequency (TF) and inverse document frequency (IDF). The former measures how many times a word appears in a document, and a higher value means the higher importance of the word to this document. IDF evaluates how frequently a word appears in a set of documents. A word occurs frequently among all documents carries less information from a specific document, such as 'then', 'a'. The final metric TFIDF is the multiply of TF and IDF [13]. In this sense, the model is well suited for the task of measuring the similarity between scientific papers. With the baseline model, the input abstract of the context paper and candidate paper are converted into vector representations using the sklearn TfidfVectorizer. Then the cosine similarity between the vector representations is calculated, and a threshold of 0.5 is applied to the cosine similarity to determine whether the pair of papers are related, indicated by a binary label output from the model.

5   Deep Learning Model

5.1   Training Phase

Inspired by Sentence-BERT, we decided to employ the Siamese Deep Learning Networks (Figure 4) [10]as our main model. In this architecture, two identical transformers with mean pooling are used to extract the text embeddings from two papers in
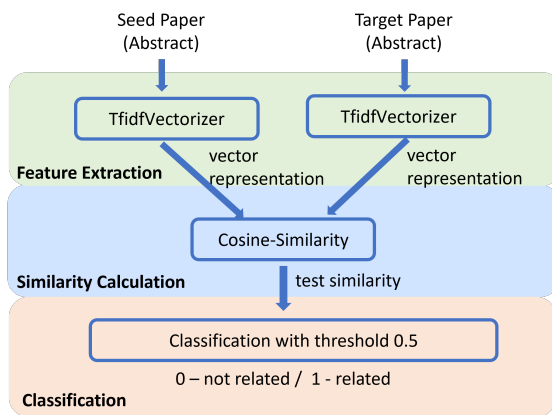
Figure 3: Baseline model architecture

the pair. Then, these two embeddings are concatenated together with their difference and fed into a binary classifier. Our networks are compatible with both BERT-based and GPT-based transformers. The output embedding has a dimension of 768, and the binary classifier is a fully-connected layer (768 x 2) for the label output.

After the training procedure, we got two trained models: the feature extraction model and the classification model. The former would be used to compute text embeddings of all papers in advance. The latter would be used to perform a pair-wise comparison on all pre-computed embeddings in the inference phase.
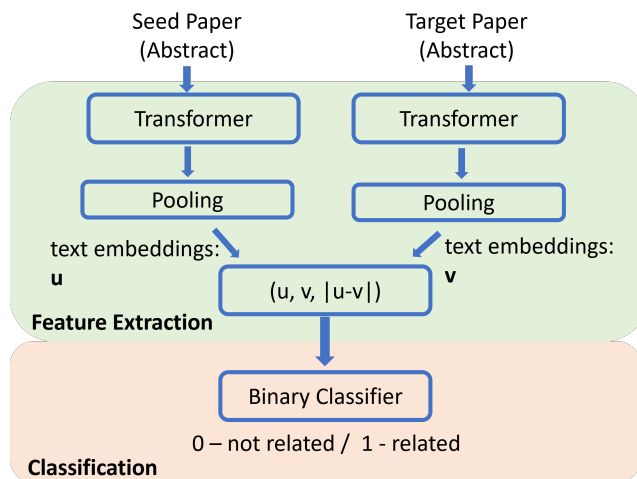
Figure 4: Deep leaning model architecture

5.2   Inference Phase

During the inference phase, the workflow would be different from the training phase, as shown in Figure 5. To save the inference time, we pre-computed text embeddings of all papers with our trained feature extraction model and saved them in a database. After the application receives the input paper from the user, it will first check whether the paper is in the database. If it is, the corresponding embeddings will be fetched directly. Otherwise, the trained feature extraction model will be invoked to compute

4

its embedding and insert it into the database. Afterward, the trained classifier will be called for input embeddings and each of the other embeddings in the database to determine the relatedness with a confidence value. Finally, the application will output the top k recommendations to the user.
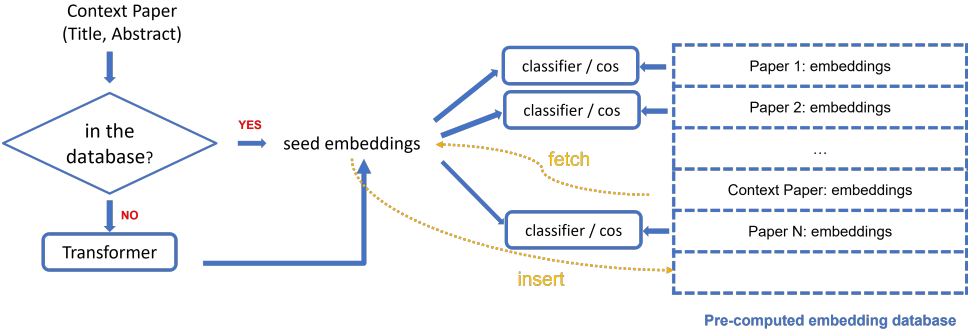
Figure 5: EZPaperSearch application architecture

6 Quantitive Results

The following table shows the quantitative results with the classifier head after we finished training. We tested multiple pre-trained transformers: GPT2-Medium, GPT2-Small, and RoBERTa-base, and used F1 score as our evaluation metric. Among all transformers, GPT2-Small achieved the best performance with reasonable processing time. Our baseline model: TFIDF model had the worst performance but the shortest processing time.

Table 2: Quantitive results

Model	Dataset	F1 Score	Processing Time
TFIDF Model (baseline)	Validation	0.78	-
	Test	0.83	**0.1s**
GPT2-Medium	Validation	0.92	-
	Testing	0.91	7s
GPT2-Small	Validation	0.89	-
	Testing	**0.93**	2s
Roberta-base	Validation	0.90	-
	Testing	0.87	2s

7 Qualitative Results

The following tables shows some examples of top 3 recommendations from our application. Overall, the recommendations are reasonable. For example, the input papers in the Figure 6 are talking about robotic control and superconductor respectively, the top 1 result from EZPaperSeach looks the most related from our perspective.

Table 3: Reasonable recommendations from EZPaperSearch (Please see the screenshots with full text in Appendix A)

Input Paper	*Local Gaussian process regression for real-time model-based robot control* High performance and compliant robot control requires accurate dynamics models which cannot be obtained analytically for sufficiently complex robot systems...
Top 1 Recommendation	*Real-Time Local GP Model Learning*
Top 2 Recommendation	*7DOFs Robot Numerical Approach Method with Jacobian*
Top 3 Recommendation	*Bifurcations and symmetries of optimal solutions for distributed robotic systems*
Input Paper	*Operating Experience of Superconductor Dynamic Synchronous Condenser* High-temperature superconductor (HTS) dynamic synchronous condensers have a small footprint...
Top 1 Recommendation	*Superconductor synchronous condenser for reactive power support in an electric grid*
Top 2 Recommendation	*Sliding mode control of three-phase four-leg shunt active power filter*
Top 3 Recommendation	*Superconducting fault current limiter to mitigate the effect of DC line fault in VSC-HVDC system*

However, we do find some cases where EZPaperSearch does not work well. In the following example, two input papers are the most related to each other based on their reference similarity. But the results from EZPaperSearch look irrelevant to the input. The embeddings from our trained transformers fail to capture all semantic meanings in this case, which leads to the misbehavior of the classifier.

Table 4: Inappropriate recommendations from EZPaperSearch (Please see the screenshots with full text in Appendix B)

Input Paper	*Excitation, observation, and ELF-MD: optimization criteria for high quality test sets* In previous work, we have shown that optimizing the number of site observations leads to more defect detection...
Top 1 Recommendation	*Parameter domain pruning for improving convergence of synthesis algorithms*
Top 2 Recommendation	*A feature for character recognition based on directional distance distributions*
Top 3 Recommendation	*Identity-Aware Convolutional Neural Network for Facial Expression Recognition*
Input Title	*Balanced excitation and its effect on the fortuitous detection of dynamic defects*
Input Abstract	Dynamic defects are less likely to be fortuitously detected than static defects because they have more stringent detection requirements...
Top 1 Recommendation	*7DOFs Robot Numerical Approach Method with Jacobian*
Top 2 Recommendation	*The improved depth estimation algorithm based on FTV*
Top 3 Recommendation	*The discussion of the principle and application about remanufacturing of covered mold welding*

8   Discussion and Learning

Overall, our model has reasonable performance on the recommendations, but the inference time is much longer than the baseline model. The reason is that he binary classifier gets invoked for each paper in the database, which significantly increases the computational overhead. The baseline model uses cosine similarity with threshold 0.5 to find the most similar items. We tried this method in our experiments, but the results looked not acceptable, especially when we provided an arbitrary sentence as the input abstract. The following two figures shows the input the cosine similarity method still recommended some papers, while the classifier returned nothing. We might need to change the way of training our model to replace the classifier with the simple cosine similarity method.

Table 5: Inputs that cosine-similarity still returns some recommendations (Please see the screenshots with full text in Appendix C)

Input Title	*ece1786*
Input Abstract	ece1786
Input Title	*Writing Equipment and Women in Europe 1500-1900* (History Paper)
Input Abstract	This article examines the history and development of the material culture of...

Looking back on the whole project, we found that the workload spent on the data processing was much heavier than what we expected. We evaluated the relatedness between all unique pairs among 224,394 papers in the cleaned data set, which required significant computation time and power. We spent lots of time optimizing our code and leaning to use multiprocessing and parallel computing with Dask dataframe. It was a great learning experience for us. Moreover, after reviewing the quantitative results, we realized that the more complex model does not necessarily yield a better performance but is always more expensive. Therefore, choosing a suitable model would be a core decision for a real application design.

9 Contribution

The following table shows the contributions from each team members.

Table 6: Individual contributions

Task	Contributions
Data Processing	- Evan wrote the initial data cleaning and labeling code - Yuhan optimized the data processing code
Algorithm	- Evan implemented the baseline model. - Evan evaluated the baseline model. - Yuhan implemented the main model. - Yuhan trained and evaluated the main model. - Evan wrote the gradio user-facing of the software. - Yuhan implemented the architecture of the software.
Documentation	- Evan and Yuhan created and reviewed Proposal, Proposal Presentation, Progress Report and Final Presentation together. - Evan wrote Introduction, Background and Related Work, Data Processing and Baseline Model in Final Report. - Yuhan wrote Architecture and Software, Quantitive Results, Qualitative Results and Discussion & Learning in Final Report.

References

[1] "Corpus-based and knowledge-based measures of text semantic similarity." AAAI Press, 2006, pp. 775–780.

[2] P. D. Turney, "Mining the web for synonyms: Pmi-ir versus lsa on toefl," 2002. [Online]. Available: https://arxiv.org/abs/cs/0212033

[3] T. K. Landauer, P. W. Foltz, and D. Laham, "An introduction to latent semantic analysis," Discourse Processes, vol. 25, no. 2-3, pp. 259–284, 1998. [Online]. Available: https://doi.org/10.1080/01638539809545028

[4] C. Leacock and M. Chodorow, Combining Local Context and WordNet Similarity for Word Sense Identification, 01 1998, vol. 49, pp. 265–.

[5] M. Lesk, "Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone," in Proceedings of the 5th Annual International Conference on Systems Documentation, ser. SIGDOC '86. New York, NY, USA: Association for Computing Machinery, 1986, p. 24–26. [Online]. Available: https://doi.org/10.1145/318723.318728

[6] J.-Y. Jiang, M. Zhang, C. Li, M. Bendersky, N. Golbandi, and M. Najork, "Semantic text matching for long-form documents," in The World Wide Web Conference, ser. WWW '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 795–806. [Online]. Available: https://doi.org/10.1145/3308558.3313707

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018. [Online]. Available: https://arxiv.org/abs/1810.04805

[8] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," 2019. [Online]. Available: https://arxiv.org/abs/1907.11692

[9] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.

[10] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," 2019. [Online]. Available: https://arxiv.org/abs/1908.10084

[11] A. Conneau and D. Kiela, "Senteval: An evaluation toolkit for universal sentence representations," arXiv preprint arXiv:1803.05449, 2018.

[12] K. Lo, L. L. Wang, M. Neumann, R. Kinney, and D. Weld, "S2ORC: The semantic scholar open research corpus," in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Online: Association for Computational Linguistics, Jul. 2020, pp. 4969–4983. [Online]. Available: https://www.aclweb.org/anthology/2020.acl-main.447

[13] J.-M. G. Sang-Woon Kim, "Research paper classification systems based on tf-idf and lda schemes," Human-centric Computing and Information Sciences, vol. 9, pp. 2192–1962, 2019.

Appendices

Appendix A

Figure 6: Reasonable recommendations from EZPaperSearch

Appendix B

Figure 7: Inappropriate recommendations from EZPaperSearch

Appendix C

Figure 8: Incorrect recommendations with cosine similarity method