CS230

An ArXiv Paper Recommender

Milind Shyani
Department of Physics
Stanford University
shyani@stanford.edu

Abstract

We live in one of the most prolific eras of scientific research. More than 15,000 research papers are submitted to the Arxiv preprint server every month. It has become impossible for any researcher to keep track or search for papers relevant to his/her field of study. In this project, we address this embarrassment of riches by exploring six different deep learning architectures: TF-IDF, Doc2Vec, LSTM, RoBERTa, GPT-2 and Sentence-BERT to create an arxiv paper recommender. Our central aim is to create meaningful document embeddings for each paper based on its abstract. The embeddings are used to curate recommendations based on the cosine similarity distance between a given paper and the rest of the corpus. Our fine-tuned transformer architectures provide impressive recommendations that are at par with the current state of the art. We argue that our citation-agnostic content based models lead to more democratic and meaningful recommendations.

1 Introduction

The Arxiv serves one of the most essential needs of modern science i.e. quick and open access to research. It has more than 1.8 million research papers as of today and counting. This phenomenal rise of submissions has led to enormous progress in science, but has also made it harder for researchers to find papers closer to their interests. The difficulty of finding the right research papers not only affects our scientific endeavours as a whole but also leads to several inefficiencies in the day-to-day life of the individual researcher. There is an extremely high opportunity cost of not having found the relevant literature, or of working on a project for a few months only to stumble upon a similar paper that solved it several years ago. Worst of all many interesting papers, especially from less famous authors, get easily forgotten.

Scientific papers contain subject specific knowledge and a sophisticated logical structure. As a result, it is hard to meaningfully search for them using text based inquiries only. Although traditional search engines such as Google provide search results based on advanced text similarity protocols, page-ranking algorithms, collaborative filtering and etc., a content based deep learning model that is fine tuned on a corpus of scientific papers would be invaluable.

Our central goal is to obtain meaningful embeddings for each paper based on its abstract. The embedding space is then used to create a paper recommender and a contextual search engine by using the cosine similarity distance between a given vector and the rest of the corpus. We employ six different methods to obtain the document embeddings for each paper and present them in increasing order of complexity. In section 4.1, we use TF-IDF and Doc2Vec. In section 4.2, we employ an LSTM recurrent neural network with a proxy task of paper title prediction. In sections 4.3 and 4.4, we fine-tune GPT-2, RoBERTa and Sentence-BERT on the high energy physics abstracts corpus to obtain more meaningful document embeddings.

CS230: Deep Learning, Winter 2021, Stanford University. LateX template borrowed from NIPS '17.

Due to the absence of a database of paper preferences of several users, we formulate three novel metrics based on co-citations, relevance and novelty to judge performance.[1] We elaborate upon these metrics in section 5 and argue that our models are at par with the current state of the art. The problem of finding a quantifiable metric to rate "good" recommendations (purely based on content) is almost as hard as the problem itself but we believe that the combination of our three metrics provides a necessary, if not sufficient, measure of performance.

2 Related work

Recommendation systems can be broadly categorized into collaborative vs content based filtering [2, 3, 4]. Collaborative filtering provide recommendations based on the rating profiles and preferences of all users collectively, while content based filtering provides recommendations based on the content of the item and the preferences of a specific user. We focus on content based filtering methods for two main reasons. First, we do not have access to user ranking profiles for scientific papers or citation statistics that are essential for any collaborative filtering model. Second, we believe that citation-agnostic content based recommendations provide more democratic recommendations instead of recommending famous papers only.

A lot of interesting work has been done on academic paper recommendation in recent years [5, 6]. Collaborative filtering methods employ several different techniques including, but not limited to, knowledge graphs [7], graph neural networks [8], reinforcement learning [9] and etc. On the other hand most state of the art content based approaches use bag-of-words, TF-IDF vectorizer [10, 11] or Word2Vec [12, 13] to generate document embeddings. These embeddings are then used to obtain recommendations with the help of clustering algorithms or cosine similarity distance. Karpathy's arxiv-sanity [11] is a good example of this method.

In our opinion, Microsoft Academic (MA) [14, 15] serves as the current state of the art for scientific paper recommendation. It uses a combination of collaborative and content based filtering and benefits from their in-house Bing search engine, citation graphs and other data mining techniques. However, its over reliance on citations only leads to relevant recommendations that are also highly popular. As a result it misses out on several interesting but less famous papers.

To the best of our knowledge, our work is the first to use the Transformer [16] based architectures [17, 18, 19] for scientific paper recommendation. We find that the recommendations obtained from fine-tuning GPT-2 and SBERT are almost at par with traditional content methods based on TF-IDF and Doc2Vec in terms of co-citations as shown in figure 3. We also find that our recommendations are at par with the elaborate collaborative+content filtering methods used in MA in terms of relevance and novelty as shown in table A.1. Finally, we discover that TF-IDF provides a tough to beat baseline for scientific paper recommendation.

3 Dataset

We use the data set provided by Arxiv on Kaggle consisting of 1.7M json entries [20]. Each json item consists of the paper metadata such as its abstract, subject category, author name, title, date of submission and etc. We focus on the domain of High energy physics theory (hep-th) including cross-lists. Our Pandas dataframe contains 140,500 papers with 4 columns, one each for arxiv id, abstract, title and authors. The average length of an abstract is 114.6 words with a standard deviation of 55 words. The total vocabulary count after preprocessing is 36210. We also create two additional datasets for measuring model performance.

The first dataset contains a list of 16 influential papers in hep-th and a corresponding list of recommendations. Each one of our six models recommends 30 papers for a given paper (hereafter referred to as the parent). Thus this dataset is a list of $16 \times 6 \times 30 = 2880$ (parent, recommendation) records with a co-citation count for each record. The co-citation count

[1] After the completion of our work we found that Microsoft Academic [1] independently also uses a similar co-citation based metric.

has been obtained using the Inspire API [21].[2] This dataset will be used to measure model performance between our six models.

The second dataset is created from a list of 8 influential papers and the corresponding recommendations from four of our best performing models on the co-citation metric (SBERT, Doc2Vec, TF-IDF and GPT-2) and MA. Each record contains (parent, recommendation), and a relevance and novelty score as given by two Stanford physics postdocs and a University of Chicago physics postdoc.[3]

4 Methods

In this section we describe the six different methods of obtaining document vectors. These vectors will be loaded to Word2Vec with their arxiv id serving as the token. The recommendations can then be easily obtained by gensim's wv.most_similar. As elaborated in the results section, the models in section 4.2 through 4.4 also use a cut-off TF-IDF similarity score while providing recommendations.

4.1 TF-IDF & Doc2Vec

We utilise the implementation by gensim [22] of TF-IDF and Doc2Vec [23] to create embeddings for each paper. The abstracts are first preprocessed by lemmatization, stemming and by removing capitalization, numerical factors, punctuation and stop words. As can be seen from the results in figure 3, both TF-IDF and Doc2Vec have varying levels of success and there isn't a huge difference in performance. By construction, TF-IDF is better at recommending papers based on specific key words while Doc2Vec is better at recommending papers that are relevant even if they do not contain the necessary key words.

TF-IDF vectorization does not have any hyperparameter to tune while Doc2Vec requires several hyperparameter and model choices. The two main methods for training Doc2Vec are the distributed bag of words (similar to Word2Vec CBOW [24]) and the distributed memory version (similar to Word2Vec Skip-gram). We trained a total of 20 models equally split between CBOW and Skip-gram. The ten choices correspond to the choice in the dimensionality of the vectors (200 vs. 300) and the window size (2, 3, 4, 5 or 6). We found that the bag of words Doc2Vec model with a window choice of 5 and vector dimensionality 300 when run for 10 epochs gave us the best results for the co-citation metric. Running for longer epochs such as 20 or 30 led to worse results. We haven't exhaustively explored window sizes beyond 6 and it is possible that our Doc2Vec implementation is not the most optimal implementation.

4.2 Word embeddings using LSTM and a proxy task

Attention based bidirectional LSTM networks are often used in complex natural language processing tasks such as machine translation, language modelling and etc. We use the LSTM network shown in figure 1, and assign to it a proxy task of predicting paper titles based on their abstracts with a categorical cross entropy loss. The actual goal of this task is to learn better word embeddings. This is achieved by using a trainable Keras embedding layer between the inputs and the first bidirectional LSTM layer. The document vectors can then be obtained by applying an average pooling layer to the word vectors. Alternatively the document vectors could also be obtained by averaging the hidden states $A^{<t>}$ or $S^{<t>}$.[4]

We limit the length of the input and the output to be 155 words and 12 words respectively, since this choice covers more than 95% of our dataset. We also use Word2Vec (with continuous bag of words, window size 5, vector dimension 52 and epochs 1) to pretrain our word embeddings before feeding them into the LSTM network. We experimented with four LSTM networks based on the number (1, 2, 4 or 10) of bidirectional LSTM layers and

[2] Inspire is an open access digital library of high energy physics papers and citation counts. We would like to thank Aarohi Oza for teaching me the Inspire API to obtain citations counts.

[3] We are grateful to postdocs R. Mahajan, R. Soni and E. Mazenc for providing this data.

[4] We would like to thank the TA Sherry Ruan for suggesting this idea to us.

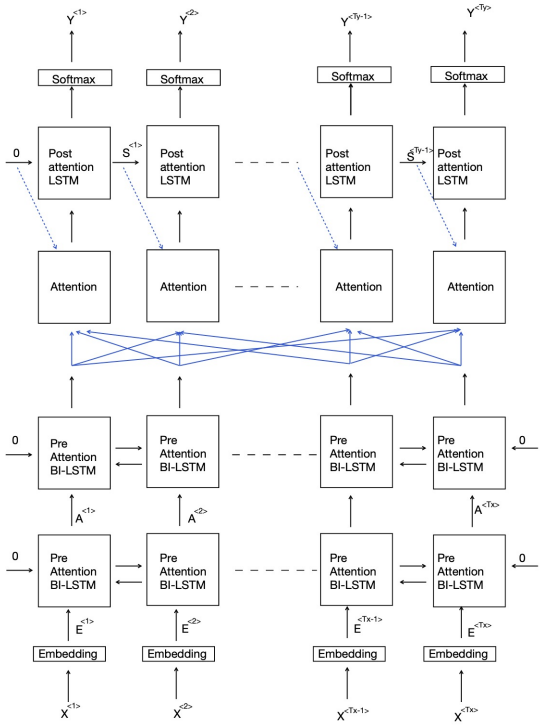

Figure 1: LSTM network with 2 bidirectional LSTM layers.

found that model performance worsens with more LSTM layers. We also noticed that longer training or larger vector dimensionality led to worse results. See appendix A.2 for more details about the FCC attention layer and plots of loss functions.

4.3 Word embeddings from fine-tuning GPT-2 and RoBERTa

We fine-tune the two famous transformer architectures, GPT-2 (medium) and RoBERTa (base) on the hep-th abstracts corpus using the huggingface transformers library [25] in PyTorch. The GPT-2 model is fined tuned using a causal language modelling task i.e. given the set of tokens from 0 to i, the model tries to predict $i + 1^{th}$ token. RoBERTa is trained using a masked language modelling task where the model tries to predict masked words in a sentence based on the unmasked words. We mask 15% of the words while training and find that the RoBERTa results are much worse than GPT2 or SBERT. A higher mask probability might have given better results but we did not have the time to explore that unfortunately.

The document embeddings are obtained using three different ways. First by applying an average pooling layer to the fine-tuned word vectors obtained from the zeroth layer of the transformer. Second by applying an average pooling layer to the average of the last four hidden states. Third by applying an average pooling layer to the concatenation of the last four hidden states. To our surprise the first method provides better recommendations than the more sophisticated second and third methods.

4.4 Sentence embeddings from fine-tuning SBERT

Sentence-BERT [19] is a Siamese (distil) BERT architecture that is extremely useful to find meaningful sentence embeddings. It is trained on a pair of sentences labelled according to their similarity with a mean-squared loss. We fine tune it on the hep-th abstracts corpus. The sentence dataset is created using three different sets of sentence pairs. The first set contains pairs that are formed by a sentence and its immediate neighbor and is given a similarity score of 1. The second consists of a sentence and its next to immediate neighbor

and is given a similarity score of 0.8. The third consists of a sentence and any randomly selected sentence from the corpus and is given a similarity score of 0. The total dataset consists of 1.1 million pairs with a split between the three sets given by 26%-26%-48%. The document vector is then obtained by averaging the sentence vectors that appear in the abstract.

5 Results and Discussion

Having obtained the abstract vectors we can obtain recommendations for any number of papers using the cosine similarity distance. We ignore all recommendations that have a TF-IDF cosine distance of less than 0.1, and provide 30 recommendations for every parent paper. We propose three novel metrics to rate the quality of our recommendations. For any given paper i they are defined as,

$$\text{Normalized relevance} \equiv nRP_i = \frac{1}{N}\sum_{j=1}^{N} r_{i,j}, \qquad \text{Normalized novelty} \equiv nNP_i = \frac{1}{N}\sum_{j=1}^{N} n_{i,j},$$

$$\text{Normalized number of co-citations} \equiv nCP_i = \frac{1}{N}\sum_{j=1}^{N} c_{i,j}, \qquad \text{where } N = 30. \tag{1}$$

Here $c_{i,j}$ is the number of times the paper i and paper j are cited together. The number $r_{i,j}$ measures the relevance of a recommended paper j to the parent paper i and is ranked on a scale of 1 (completely irrelevant) to 5 (very relevant) by an advanced physics researcher. For a given i and j if $r_{i,j}$ is greater or equal to 3 *and* if the recommended paper j is unknown to the researcher, $n_{i,j}$ is marked 1. For all other cases $n_{i,j}$ is set to zero. nCP_i serves as an objective metric while nRP_i and nNP_i serve as a subjective but more meaningful metric for paper recommendations.

We collect co-citation scores for 16 influential physics papers from the last two decades for each of our 6 models and MA. The individual scores for each paper can be found in figure 3 in the appendix, here we only mention the average co-citation score $\frac{1}{16}\sum_{i=1}^{16} cCP_i$,

	TF-IDF	Doc2Vec	LSTM	RoBERTa	GPT-2	SBERT	MA
Average co-citations	18.19	16.17	7.21	5.28	12.44	13.87	127.80

MA is by far the winner in terms of co-citation. This is expected since MA by design [1] maximizes the co-citation score. We believe that the average co-citation metric is far from perfect and we need more human input. We collect the relevance and novelty score for 8 influential papers from 3 advanced physics researchers. We compare the best performing models on the co-citation score with each other. The individual scores can be found in table A.1 and we limit here to the average relevance score $\frac{1}{8}\sum_{i=1}^{8} nRP_i$ and the average novelty score $\frac{1}{8}\sum_{i=1}^{8} nNP_i$,

	TF-IDF	Doc2Vec	GPT-2	SBERT	MA
Average relevance	3.61	2.91	3.12	2.85	3.52
Average novelty	0.36	0.38	0.35	0.29	0.03

When relevance and novelty are taken in to account MA performs significantly worse than our content based models. TF-IDF emerges as the most promising model when taken all three metrics in to account. It benefits from the fact that scientific keywords are very specific in their meaning and use. Nevertheless we believe that our work provides a proof of concept that transformers are capable of providing state of the art results. Indeed, as shown in appendix A.1, our fine-tuned transformers beat Doc2Vec and TF-IDF 34% of the time. We believe that SBERT and GPT-2 when trained from scratch on entire papers will serve as the new state of the art for scientific paper recommendation and we are excited to report on it in the future.

A Appendix

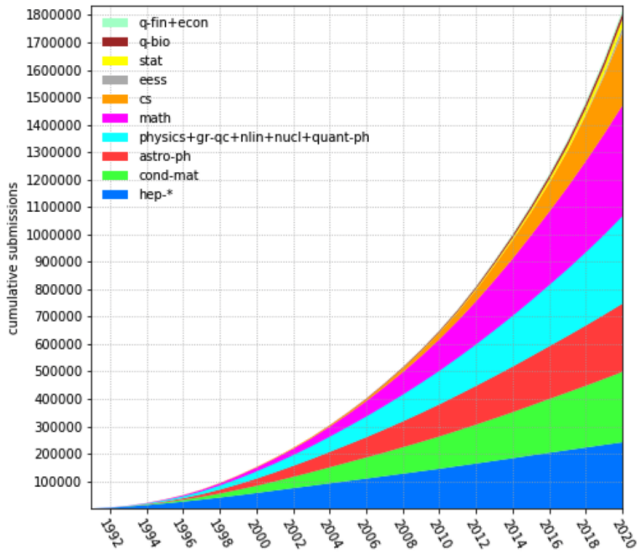

Figure 2: Houston, we have a problem...

A.1 Detailed results

In this subsection we provide the results dataset created with the help of Inspire API and the feedback of 3 advanced physics researchers. Figure 3 contains 16 rows, one for each paper, and 7 columns, six for our models and one for Microsoft Academic. We find that our transformer architectures place first or second more than 34% of the time. Although Doc2Vec and TF-IDF win 66% of the time, we believe that transformers hold a lot of promise and are worthy of further investigation.

Paper name	TF-IDF	Doc2Vec	MA	SBERT	GPT-2	LSTM	RoBERTa
Aharony hep-th/0310285	622	189		393	136	22	73
Dong 1411.7041	641	765	2818	531	680	251	383
Dong 1601.05416	783	408		494	538	260	306
DSD 1805.00098	217	337		314	196	191	146
Faulkner 1605.08072	492	198		244	134	148	37
Hartman 1509.0014	267	314	1102	410	171	234	61
HKS 1405.5137	228	233		144	171	158	120
Islands 1908.10996	433	566		235	340	126	188
Kaplan 1212.3616	563	1107		836	931	521	218
Maldacena 1606.01857	676	164	3748	62	486	67	180
Minwalla 0712.2456	775	746		413	289	317	62
MMV 1611.03470	877	250		190	443	217	187
Saad 1611.04650	686	716		442	484	356	47
Simon 1611.04650	504	477		789	111	234	46
SYK 1604.07818	913	1280		849	854	322	439
Witten 0712.0155	54	12		310	9	35	40
Grand Total	**8731**	**7762**	**7668**	**6656**	**5973**	**3459**	**2533**
Average co-citations	18.19	16.17	127.80	13.87	12.44	7.21	5.28

Figure 3: A list of 16 parent papers and the number of co-citations by each model. Green corresponds to the highest number of co-citations while yellow corresponds to the second highest. We do not include MA in this ranking scheme. The average co-citations are obtained by summing the entries of every column, dividing by the number of non-zero rows i.e. 16 for our models and 3 for MA, and finally dividing by the number of recommendations provided by that model i.e. 30 for our models and 20 for MA. Thus for any given row i the entry in the column calculate $\sum_{j=1}^{N} c_{i,j}$ where $N = 30$ for our models and $N = 20$ for MA.

Table A.1 contains the relevance and novelty for 8 influential papers as judged by 3 advanced physics researchers. Figure 4 contains the histogram of the relevance score of four of our best performing models.

	TF-IDF	Doc2Vec	GPT-2	SBERT	MA
Verlinde et al.	(136,13)		(104,11)		(64,2)
Kaplan et al.	(88,11)		(89,5)		(78,1)
Dong et al.	(107,8)			(94,5)	(65,0)
DSD et al.	(102,11)			(98,14)	(67,0)
Maldacena et al.		(88,10)	(72,10)		(87,0)
Faulkner et al.		(76,9)		(71,6)	(62,0)
Hartman et al.		(70,7)		(79,10)	(74,2)
Harlow et al.		(115,19)	(109,16)		(74,2)
Normalized average	(3.61,0.36)	(2.91,0.38)	(3.12,0.35)	(2.85,0.29)	(3.52,0.03)

Table 1: The (relevance,novelty) score for 8 papers. The normalized average is obtained first by summing all the elements of a given column and diving by the number of non-empty rows (4 for our models and 8 for MA), and further dividing by the number of recommendations per parent paper of that model i.e. 30 for all our models and 20 for Microsoft academic.

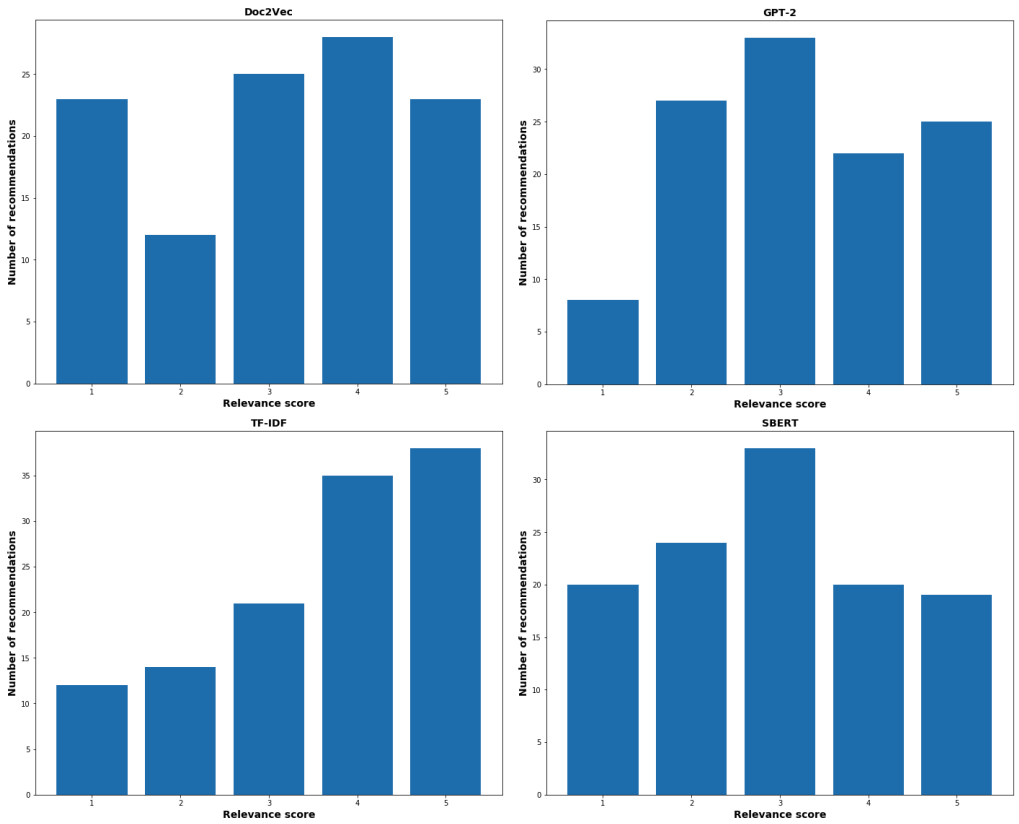

Figure 4: Histogram of relevance as judged by advanced physics researchers.

A.2 Loss functions

All our models are convergent and here we show the loss functions for our LSTM model and GPT-2. The LSTM attention layer is made up of 3 fully connected layers with 155 (max length of the input), 10 and 1 neurons respectively.

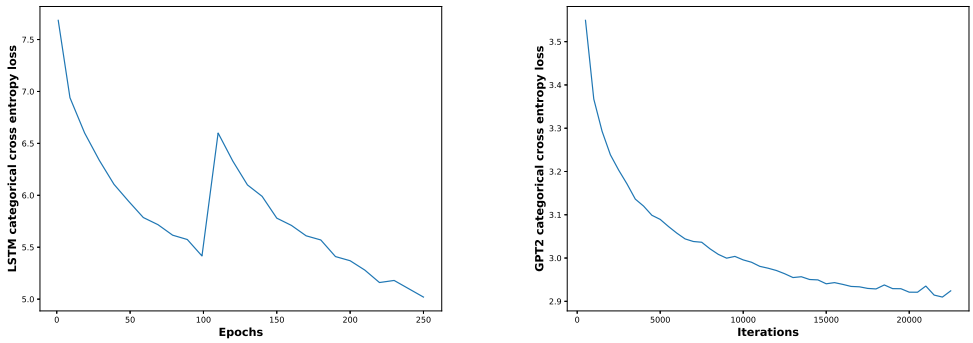

Figure 5: Left panel : LSTM cross entropy loss. Note that the LSTM loss jumps from time to time and it is crucial to stop at the right number of epochs. Right panel : GPT-2 cross entropy loss.

A.3 Dataset and sample results

Figure 6 shows a few rows of our Pandas dataframe.

	id	abstract	title	authors
444	0704.0445	This paper extends and builds upon the resul...	Geometrically Engineering the Standard Model: ...	Jacob L. Bourjaily
448	0704.0449	We apply mirror symmetry to the problem of c...	Worldsheet Instantons and Torsion Curves, Part...	Volker Braun, Maximilian Kreuzer, Burt A. Ovru...
489	0704.0490	The possible existence of axion-like particl...	Long Distance Signaling Using Axion-like Parti...	Daniel D. Stancil
504	0704.0505	We construct a classical solution of an Eins...	Exact Solutions of Einstein-Yang-Mills Theory ...	Hironobu Kihara, Muneto Nitta

Figure 6: A few entries from the `hep-th` pandas dataframe.

The recommendations can be obtained by inputting Arxiv ids of one or more papers as shown in figure 8. We can also use the embedding space to create a contextual search engine as shown in figure 7.

```
search("ads cft entanglment islands information paradox",15)
```

```
1) Quantum Extremal Islands Made Easy, Part I: Entanglement on the Brane by Hong Zhe Chen, Robert C. Myers, Dominik N
euenfeld, Ignacio A. Reyes,
  Joshua Sandor (0.5870394). Arxiv: 2006.04851

2) Islands outside the horizon by Ahmed Almheiri, Raghu Mahajan, Juan Maldacena (0.58141845). Arxiv: 1910.11077

3) Islands in Asymptotically Flat 2D Gravity by Thomas Hartman, Edgar Shaghoulian, Andrew Strominger (0.56930244). Ar
xiv: 2004.13857

4) Notes on islands in asymptotically flat 2d dilaton black holes by Takanori Anegawa, Norihiro Iizuka (0.54549414).
Arxiv: 2004.01601

5) Islands in cosmology by Thomas Hartman, Yikun Jiang, Edgar Shaghoulian (0.5453882). Arxiv: 2008.01022
```

Figure 7: Contextual search engine via better document embeddings.

```
fast_trans_recommender(sentence_model_gpt2,"1908.10996")
```

```
 1) With TFIDF score 0.563462495803833 and GPT-2 rank 0
 Paper title : ['Geometric secret sharing in a model of Hawking radiation'].
 Abstract : ['  We consider a black hole in three dimensional AdS space entangled with an\nauxiliary radiation syste
m. We model the microstates of the black hole in terms\nof a field theory living on an end of the world brane behind
the horizon, and\nallow this field theory to itself have a holographic dual geometry. This\ngeometry is also a black
hole since entanglement of the microstates with the\nradiation leaves them in a mixed state. This "inception black ho
le" can be\npurified by entanglement through a wormhole with an auxiliary system which is\nnaturally identified with
the external radiation, giving a realization of the\nER=EPR scenario. In this context, we propose an extension of the
Ryu-Takayanagi\n(RT) formula, in which extremal surfaces computing entanglement entropy are\nallowed to pass through
the brane into its dual geometry. This new rule\nreproduces the Page curve for evaporating black holes, consistently
with the\nrecently proposed "island formula". We then separate the radiation system into\npieces. Our extended RT rul
e shows that the entanglement wedge of the union of\nradiation subsystems covers the black hole interior at late time
s, but the\nunion of entanglement wedges of the subsystems may not. This result points to a\nsecret sharing scheme in
Hawking radiation wherein reconstruction of certain\nregions in the interior is impossible with any subsystem of the
radiation, but\npossible with all of it.\n'].
 Authors : ["Vijay Balasubramanian, Arjun Kar, Onkar Parrikar, G\\'abor S\\'arosi,\n  Tomonori Ugajin"].
 Arxiv ID : ['2003.05448']


 2) With TFIDF score 0.3994777202606201 and GPT-2 rank 1
 Paper title : ['Entanglement entropy of black holes'].
 Abstract : ["  The entanglement entropy is a fundamental quantity which characterizes the\ncorrelations between sub-
systems in a larger quantum-mechanical system. For two\nsub-systems separated by a surface the entanglement entropy i
s proportional to\nthe area of the surface and depends on the UV cutoff which regulates the\nshort-distance correlati
```

Figure 8: Paper recommendation using GPT-2

References

[1] Anshul Kanakia, Zhihong Shen, Darrin Eide, and Kuansan Wang. A scalable hybrid research paper recommender system for microsoft academic. *The World Wide Web Conference*, 2019.

[2] P. Melville and V. Sindhwani. Recommender systems. In *Encyclopedia of Machine Learning*, 2010.

[3] Shuai Zhang, L. Yao, Aixin Sun, and Yi Tay. Deep learning based recommender system. *ACM Computing Surveys (CSUR)*, 52:1 – 38, 2019.

[4] Dhoha Almazro, Ghadeer Shahatah, Lamia Albdulkarim, Mona Kherees, Romy Martinez, and William Nzoukou. A survey paper on recommender systems. *ArXiv*, abs/1006.5278, 2010.

[5] Xiaomei Bai, Mengyang Wang, I. Lee, Z. Yang, Xiangjie Kong, and Feng Xia. Scientific paper recommendation: A survey. *IEEE Access*, 7:9324–9339, 2019.

[6] Ivens Portugal, P. Alencar, and D. Cowan. The use of machine learning algorithms in recommender systems: A systematic review. *Expert Syst. Appl.*, 97:205–227, 2018.

[7] Qingyu Guo, Fuzhen Zhuang, C. Qin, H. Zhu, X. Xie, H. Xiong, and Q. He. A survey on knowledge graph-based recommender systems. *ArXiv*, abs/2003.00911, 2020.

[8] S. Wu, Wentao Zhang, Fei Sun, and B. Cui. Graph neural networks in recommender systems: A survey. *ArXiv*, abs/2011.02260, 2020.

[9] M. Afsar, Trafford Crump, and B. Far. Reinforcement learning based recommender systems: A survey. *ArXiv*, abs/2101.06286, 2021.

[10] Robin van Meteren. Using content-based filtering for recommendation. 2000.

[11] http://www.arxiv-sanity.com.

[12] H. Hassan. Personalized research paper recommendation using deep learning. *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*, 2017.

[13] Hugo Caselles-Dupré, Florian Lesaint, and Jimena Royo-Letelier. Word2vec applied to recommendation: hyperparameters matter. *Proceedings of the 12th ACM Conference on Recommender Systems*, 2018.

[14] https://academic.microsoft.com/home.

[15] A. Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, B. Hsu, and Kuansan Wang. An overview of microsoft academic service (mas) and applications. *Proceedings of the 24th International Conference on World Wide Web*, 2015.

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[17] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[18] Y. Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692, 2019.

[19] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP/IJCNLP*, 2019.

[20] https://www.kaggle.com/Cornell-University/arxiv.

[21] https://github.com/inspirehep/rest-api-doc.

[22] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA.

[23] Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents. *ArXiv*, abs/1405.4053, 2014.

[24] Tomas Mikolov, Kai Chen, G. S. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In *ICLR*, 2013.

[25] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.