Embedding-Augmented Deep CNN for PudMed Journal Recommendation

Calvin Chi
UC Berkeley
Berkeley, CA
calvin.chi@berkeley.edu

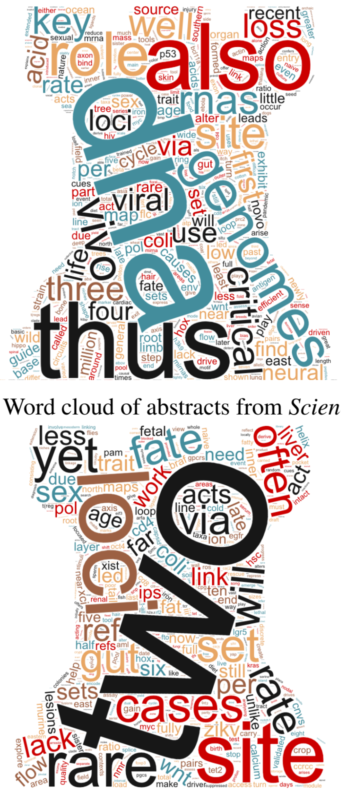
Abstract

Convolution neural networks (CNN) are effective methods for document classification (Kim, 2014). This paper explores whether CNNs are similarly effective for journal recommendation from PubMed abstracts, a task that if successful can help authors decide which journals to submit to. Of the CNN architectures explored, evidence points to the embedding-augmented CNN as the most effective neural architecture, which in this paper receives topic and impact factor embeddings following the last convolution layer as additional input. This result supports the intuition that research topic or result significance are relevant in helping determine the appropriate journal to submit to.

1 Introduction

Submitting an article to a scientific journal is a time-consuming process, with many format guidelines that need to be adhered to. Thus, it is ideal to know which journals are a good fit for a paper during the publication process. After a paper has been submitted, the editorial office sometimes first assesses an article to determine whether the paper is a good fit before assigning it to reviewers. This initial assessment is usually quick and can take less than a week, which suggests that the abstract alone may be sufficient for establishing the "fit" for a journal.

This paper evaluates the effectiveness of journal detection from abstract text, in research articles from PubMed (Coordinators, 2017), a popular database of biomedical research articles. PubMed articles are limited to those on research topics listed by the American Society of Human Genetics (ASHG 2018). Intuitively, the "fit" of a journal depends on factors such as research topic, significance of the result, or even the style of writing. For instance, journals such as *PLoS Genetics* or *BMC Bioinformatics* prefer research articles on specific topics whereas journals such as *Science* or *Nature* prefer articles of high significance and impact. The word clouds in Figure 1 shows that despite their supposed similarities, the abstracts from journals *Science* and *Nature* have a different distribution of words.



(a) Word cloud of abstracts from *Science*.

(b) Word cloud of abstracts from *Nature*.

Figure 1: Word Cloud comparison of two similar journals.

Neural-based methods have proven to be effective for supervised learning from text data. This paper evaluates several convolutional neural network (CNN) architectures to answer two questions

1. Does the abstract text alone contain enough information for effective journal detection?
2. How relevant are research topic and finding significance for journal detection?

Question 1 will be addressed with a standard CNN to predict journals. Question 2 will be addressed with a multitask CNN that predicts journal, research topic, and impact factor simultaneously, and a multi-input CNN that receives topic and impact factor embeddings after all convolution operations to help journal detection.

2 Related Work

Limited work has been done that directly addresses the journal detection problem. The two closest works found are Jane and Elsevier's journal recommendation systems (Schuemie and Kors, 2008; Elizabeth Ash and Lyndsay Scholefield). Both Jane and Elsevier's search engines suggest journals based on similarities between input text and existing journal articles, taking on a nearest neighbor's approach. This paper differs from this work by building a direct mapping between text and journal labels.

Other related works exist that study academic literature with different aims from ones in this paper. These include predicting the impact of an academic paper based on its citation network and predicting long-term citations of a paper based on citations in the first few years after publication (McNamara et al., 2013; Abrishami and Aliakbary, 2018).

One of the challenging aspects of this work is that the output space for journal detection is in the thousands. A related work addresses this challenge with hierarchical deep learning, which classifies a document successively from broad to fine classes (Kowsari et al., 2017). However, this hierarchical approach cannot be applied to journal prediction yet because the notion of journal categories is currently ill-defined and not readily available. For example, it is not possible to group journals by research topic since some journals publish research in a variety of topics.

3 Data

Abstracts are obtained from PubMed, with the inclusion criteria based on whether its research topic is listed by ASHG (ASHG 2018). The research topics included are Mendelian diseases, statistical genetics, population genetics, bioinformatics, omics technologies, genome structure, epigenetics, and developmental genetics[1]. Abstracts that show up in more than one PubMed topic search result are not included so that each abstract is assigned strictly one research topic. After data cleaning, the final dataset comprises 415,381 PubMed abstracts, each labeled with its research topic, impact factor of the journal it was published in, and name of journal.

Data collection starts by programmatically downloading abstracts by topic using Biopython (Cock et al., 2009). All articles must be within 10 years of 2018 and review articles are excluded from the search result. Impact factor annotations are obtained from other online resources since PubMed does not provide this information. Metadata of all journals in PubMed are first obtained from PubMed[2]. As of October 21, 2018, this list contains 31,689 journal records. Next, impact factors are obtained from CiteFactor (Citefactor), a resource containing impact factors for 8,771 journals. The impact factors for journals not in CiteFactor are obtained by scraping Google's quick answer box results using the `requests` and `BeautifulSoup` Python libraries. Together, this resulted in a final journal list of 7,150 journals with impact factors, of which $\sim 25\%$ of the impact factors are obtained from Google. Abstracts published by a journal without an impact factor annotation are excluded from the final dataset. Other reasons for excluding journals include missing online or print ISSN, journal names with the word "review", missing journal name abbreviation, or duplicate journal name abbreviation[3]. Although nearly 80% of journals in the original master list are excluded because of missing impact factor, all mainstream biomedical journals are in the master list. Many journals for which the impact factor are not easily found are lesser known journals, review journals, or international journals.

Mendelian phenotypes, epigenetics, and omics

[1] The topic of complex disease is excluded because most, if not all of the results returned using the "complex traits" or "polygenic disorder" keywords are review articles. Thus, keywords at the level of complex traits is too general for finding abstracts on specific complex diseases such as multiple sclerosis.

[2] `ftp://ftp.ncbi.nih.gov/pubmed/J_Medline.txt`.

[3] In which case one journal is chosen at random.

technologies are the least common research topics in the final dataset. This is not surprising because epigenetics is a relatively new field in genomics research and most diseases are not Mendelian diseases. Since omics technologies is concerned with the development and improvement of large-scale assaying techniques for molecular biology, it is not reasonable to expect these types of articles to outnumber articles whose research depends on these technologies. Figure 2 shows the distribution of research topics in the dataset.

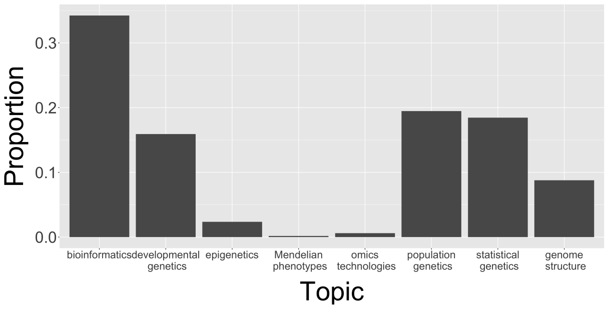


Figure 2: Research topics distribution.

The majority of abstracts belong to journals with impact factors less than 25, with only a few outliers, such as abstracts published by *A Cancer Journal for Clinicians*, which has an impact factor of 162.5. The distribution of impact factors is shown in Figure 3.

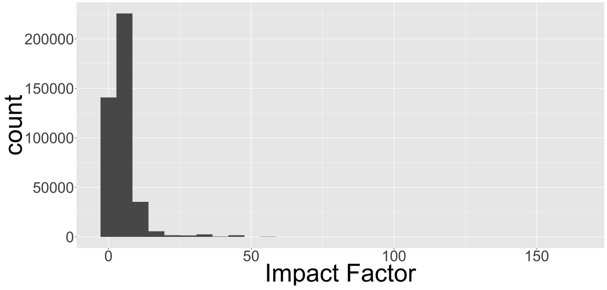


Figure 3: Impact factor distribution.

After the data cleaning steps described, a total of 4,222 journals exist in the dataset. Figure 4 plots the number of abstracts by most popular journals. The top five most popular journals are *PLoS ONE*, *bioinformatics*, *scientific reports*, *BMC bioinformatics*, and *PNAS*, all of which are well-known and commonly read in the scientific community. Given that the topic bioinformatics makes up the majority of research topics, it is not surprising to see the journals *bioinformatics* and *BMC bioinformatics* among the top five popular journals. The rest of the top five journals publish a broader range of research topics.

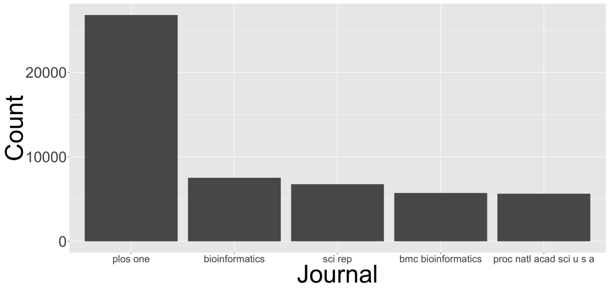


Figure 4: Number of abstracts published by the top five frequent journals in dataset.

Finally, rare journals are removed from the dataset to establish a more feasible goal for relevant journals. In this project, all journals represented by less than 0.01% (~ 40 abstracts) of abstracts in the dataset are excluded. This removed roughly 7.1% of abstracts. This should not compromise the purpose of this project since rare journals may have a narrower scope. For instance, a journal may be rare because it only publishes on a niche research area or tends to publish papers from a certain country. The final dataset comprises 1,548 unique journals.

4 Method

The three CNN architectures to be explored for journal prediction are

- Baseline CNN architecture: single input, single output CNN for journal prediction.
- Multitask CNN: learns prediction tasks of research topic, impact factor, and journal simultaneously.
- Embedding-augmented CNN: multi-input CNN that receives topic and impact factor embeddings trained from the baseline CNN.

In this work, words are represented by embeddings in $\mathbb{R}^{200}$ pre-trained on around 700,000 PubMed articles. The training was done with word2vec with a window size of five (Pyysalo et al., 2013). All embeddings are normalized to have $l2$ norm of one. Embeddings represent words in a meaningful space based on their context distribution in the corpus. Due to memory considerations, only the top 750,000 frequent words in the training dataset are assigned embedding representations. The maximum allowed abstract length is set to 500 words with post-text padding for shorter abstracts. The optimizer of choice is Adam with a learning rate of 0.001, and all training is performed for 2 epochs through the training dataset. The training dataset comprises 80% of the entire dataset, and the development and test datasets each comprise 10% of the dataset. Computation and storage was provided by AWS instance type c5.2xlarge.

4.1 Baseline CNN

The CNN architecture of the baseline model is based on the CNN model reported by Yoon Kim in 2014 (Kim, 2014). Briefly, CNNs train by learning filters to recognize phrases of a sentence that are relevant to the document classification task at hand. Thus, a filter of size n is designed to recognize n-grams. The architecture of the baseline CNN in this work is

1. Fixed embedding layer of pre-trained weights

2. First convolution layer: 1D convolution with 128 filters of size 5 with ReLU activation, batch normalization, max pooling of window size 5

3. Second convolution layer: 1D convolution with 128 filters of size 5 with ReLU activation, batch normalization, max pooling of window size 35

4. Dense layer with 128 output units with ReLU activation

5. Dense output layer with softmax

The batch normalization layer is placed after non-linearity as recommended by an analysis of batch normalization placement (Dmytro Mishkin). Briefly, the main purpose of batch normalization is to limit the covariance shift by normalizing the activations of each layer. Batch normalization also has the effect of making the neural network more robust to changes in hyperparameters.

4.2 Multitask CNN

One way to address how important research topic and finding significance are for journal prediction is with multitask learning. Multitask learning is the task of simultaneously learning multiple prediction tasks, which in this project are predictions of impact factor, research topic, and journal. This approach is especially effective when the different prediction tasks are sufficiently similar such that the weights obtained from training each prediction tasks independently end up being quite similar (Caruana, 1997).

Since impact factor is a continuous measure, this leads to multitask learning involving two classification tasks and one regression task. However, this is problematic because the CNN has to train for shared weights that output values between 0 and 1 and values in $[0, \infty)$. Thus, impact factor is first discretized so that all tasks are in the classification setting. In a preliminary analysis, both impact factor and journal detection improved in development accuracy after this conversion. Impact factor is discretized into the bins $[0, 2.5]$, $(2.5, 5]$, $(5, 10]$, $(10, 15]$, $(15, \infty)$. These bins are arbitrarily defined based on the observation from Figure 3 that most impact factors fall between 0 to 10 with a mode close to 5. Although arbitrary, this does not detract from the main purpose of learning a mapping between abstract text and impact factor so that embeddings can be generated.

4.3 Embedding-Augmented CNN

The embedding-augmented CNN is a multi-input version of the baseline CNN for predicting journals. The additional inputs are embeddings trained from research topic and impact factor. The embeddings are concatenated with the vector output after the last max pooling layer. The architecture of the embedding-augmented CNN up to the concatenation is shown in Figure 5.

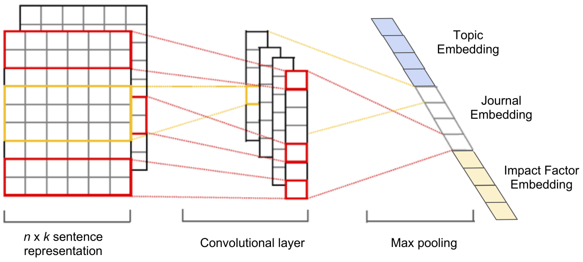


Figure 5: Multi-input CNN architecture.

After the concatenation layer are the following densely-connected layers.

1. Dense layer with 1,000 output units with ReLU activation, batch normalization

2. Dropout layer with probability 0.1

3. Dense layer with 1,000 output units with ReLU activation

4. Dense output layer with softmax

The topic and impact factor embeddings are trained using with the baseline CNN architecture. Both embeddings are taken to be the activation output in $\mathbb{R}^{128}$ before the final softmax layer of the standard CNN.

By design, the embedding-augmented CNN architecture can decide how much the topic or impact factor embeddings should influence journal prediction. If the topic and impact factor embeddings are completely irrelevant, then the model falls back to become similar to the baseline CNN. This flexibility in training on the inputs is in a similar spirit with the state-of-the-art ResNet, which effectively allows for the training of shallower neural networks if the prediction task does not necessarily need training on all specified layers (He et al., 2015).

4.4 Evaluation

Performance will be evaluated on a completely independent test dataset that comprises 10% of the entire dataset. Both the research topic and impact factor embeddings are strictly trained from the training dataset. Although accuracy can be used to reflect performance, it is not the best metric for journal recommendation. Since a good paper can usually be accepted by multiple journals, in reality a recommendation for a paper should consist of all journals that would accept the paper. However, since the counter-factual of submitting to a journal other than the one the paper was published by does not exist in the data, training is limited to building a mapping between one abstract and one journal. Ideally, true performance should be measured by precision and recall on counter-factual journal submission attempts.

One way to approximate the ideal evaluation is to consider the model correct for paper i if the journal it is published by is contained within the top k journal predictions ranked by probability. The assumption behind this approximation is that if true journal label is contained within the top k predictions, then most of the $k - 1$ journals are also journals the paper could be accepted by. This accuracy, which will be referred to as coverage accuracy at k in this paper, is computed as

$$\mathcal{A}(y, H_K) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(y^{(i)} \in H_k^{(i)}) \qquad (1)$$

where $y^{(i)}$ is the true label of abstract i and $H_k^{(i)}$ is the set of k journals with the top k prediction probabilities for abstract i. Note that this metric is in the same spirit as, but still slightly different from, the precision at k metric, which is the proportion of k recommendations that are true.

Since in practice the best k is unknown, another way to evaluate the model is with the area under the curve (AUC) of coverage accuracy at k against percent coverage of classes due to k. This metric is similar to other AUC metrics and ranges from 0 to 1, with 1 corresponding to the score of a perfect model. Alternatively, one can report performance as the smallest k that reaches a desired coverage accuracy.

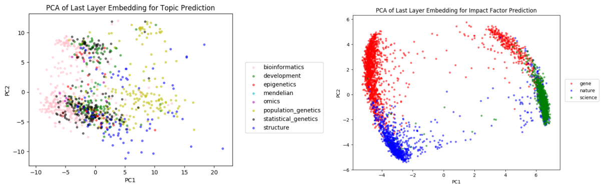
5 Results

Across the metrics of accuracy, AUC, and k to achieve 90% coverage accuracy, evidence points to the embedding-augmented CNN as the best neural architecture for journal prediction. Performance is then followed, in order, by the baseline CNN and multitask CNN. As reference, the majority class classifier would achieve 6.6% accuracy on the test dataset. Table 1 summarizes the numerical performance results for all models.

Model	Accuracy	AUC	k for 90% coverage accuracy
Baseline CNN	21.6%	0.975	80
Multitask CNN	19.1%	0.971	100
Embedding-augmented CNN	23.7%	0.978	60
Majority class	6.6%	NA	NA

Table 1: Performance summary of baseline, multitask, and embedding-augmented CNN.

By design, the embedding-augmented CNN should have performance at least as good as the baseline CNN, and this was indeed the outcome of the experiments. In a preliminary analysis, a 3-layer fully-connected neural network on just the embeddings was only able to reach a 12% development accuracy. Thus, the embeddings themselves were insufficient to reach performance comparable to the baseline CNN. Based on the performances of topic and impact factor prediction using standard CNNs, the research topic embedding seems to be more successful in generating features that are discriminative of topic than the impact factor embedding. The developmental test accuracy for research topic was 80.2% and 53.9% for impact factor. Figure 6 plots PCA of topic and impact factor embeddings. In Figure 6*a*, maximum separation can be observed between points from bioinformatics and points from population genetics and gene structure. This is expected since bioinformatics is generally considered quite different from population genetics and gene structure. Considerable overlap can be observed between points from bioinformatics with points from statistical genetics, developmental genetics, epigenetics, and omics. This again is unsurprising except for the overlap between developmental genetics and bioinformatics, since development genetics focuses on the genetic basis of embryonic and postnatal development and growth. In Figure 6*b*, the separation between points from *Gene*, *Nature*, and *Science* is observed. The journal *Gene* has impact factor 2.32, *Nature* has impact factor 42.35, and *Science* has impact factor 37.21.

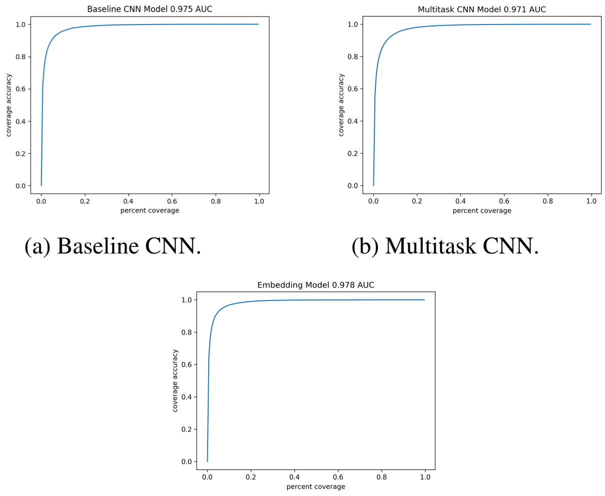


(a) Embeddings PCA of 1,000 randomly selected abstracts.

(b) Embeddings PCA of abstracts from *Gene*, *Nature*, and *Science*.

Figure 6: PCA of embeddings.

Figure 7 shows the coverage accuracy versus percent coverage for all three neural architectures. It can be observed that 90% coverage accuracy can be achieved with $k \leq 100$ for all models, and this demonstrates that all three models were able to identify journals highly irrelevant for a given abstract.



(a) Baseline CNN.

(b) Multitask CNN.

(c) Embedding-Augmented CNN.

Figure 7: Plot of coverage accuracy versus percent coverage. Percent coverage is calculated as k divided by 1,548, the number of unique journals in the dataset.

In the multitask CNN, both journal and research topic prediction suffered while impact factor prediction improved, compared to the respective standard CNN models. On the development dataset, journal prediction accuracy dropped from 21.7% to 18.9%, topic accuracy dropped from 80.2% to 77.3%, and impact factor accuracy improved from 53.9% to 55.4%.

6 Conclusion and Future Directions

This paper reports the effectiveness of CNNs in predicting the journal from abstract text and supports the hypothesis that the abstract text alone contains sufficient information to effectively detect the journal it was published by. Between multitask CNN and embedding-augmented CNN, the embedding-augmented CNN demonstrates itself as more effective in utilizing additional information on research topic and impact factor associated with the abstract. The improvement in performance from baseline provides evidence that research topic and result significance are relevant to determining which journal a paper was published by, although error bars on the performance are needed to establish statistical significance in the future. Overall, all three models are quite effective at identifying irrelevant journals for an abstract. In at least 90% of predictions, the true journal is contained within the top 6.5% of journal recommendations ranked by probability.

There is evidence that there is a stronger association between abstract text and research topic than between abstract text and result significance. The baseline CNN was able to achieve a validation accuracy of 80.2% on research topic prediction across 8 categories but only 53.9% on impact factor prediction across 5 categories. This outcome is not unexpected. The usage of a few keywords is itself often sufficient for establishing the research topic for the reader. However, determining the significance of a result should require more than interpreting the meaning of phrases or style of writing. Instead, significance is usually also evaluated by considering what has already been discovered in the field, in order to establish the novelty of the work or how the results can help progress the field further in the future. Thus, if higher-order reasoning based on knowledge of a research field is what primarily determines significance, then the abstract alone is not enough to infer significance.

Research topic and impact factor embeddings generated in this work are by themselves not sufficient for establishing superior performance over the baseline CNN model. This could be due to several reasons. One reason is that the eight predefined research topics and impact factor labels may not themselves be the appropriate labels for representing topic and significance for discriminating between journals. For instance, the topic label of Mendelian disease may be too broad to help differentiate between journals that publish exclusively on sickle-cell anemia or Tay-Sachs disease, both of which are subtopics of Mendelian disease. Impact factor may not be the best measure for individual paper quality since it is a quality measure of papers published by a journal overall. A second reason research topic and impact factor embeddings alone may not be enough to effectively discriminate between journals is because there are other unknown factors which are important for predicting journals, such as style of writing.

The multitask CNN was the worst performing model of the three. Although the three prediction tasks for journal, topic, and impact factor are similar enough that the drops in performance are not appreciable, they are not sufficiently similar to yield an improvement in performance for journal prediction. Multitask learning is effective if the lower-order features for the prediction tasks are quite similar, such as how edges or corners are key features for most object detection tasks in computer vision. The multitask CNN in this paper may have suffered performance because the lower-order features for impact factor may be very different from the lower-order features used for remaining prediction tasks. Incidentally, the development accuracy for impact factor is the only task with accuracy improvement, so it may be interesting to see if multitask CNN performance improves when the tasks are limited to just journal and topic predictions.

The fact that the embedding-augmented CNN only offered a 2.1% additional improvement in test accuracy over the baseline CNN suggests that there may exist other higher-order features that are important for journal prediction, such as the style of writing. While this work evaluates the effectiveness of incorporating topic and impact factor for journal prediction, it also illustrates one way to evaluate how important a particular aspect of a document is for a given classification task. Future work for journal prediction could use the embedding-augmented CNN to evaluate the relevance of other embeddings for journal prediction, such as embeddings for academic institution or reputation of authors.

This paper also reports the effectiveness of CNNs for journal recommendation from abstract text. For an author with a pre-determined list of potential journals to submit to, it is more practical to train a CNN model with a journal output space constrained by the pre-determined list. Alternatively, one could train a general journal recommendation model like the one in this work and eliminate potential journals that are not part of the top k recommendations.

This work can be extended in several ways. One way is to find additional search keywords used to retrieve abstracts of a research topic so that more data is collected. Along with more data, it may be interesting to see if the introduction text can yield additional discriminative power. Another way is to perform hyperparameter tuning and allow the training to continue for more epochs. Regarding the neural architecture, convolution filters of different lengths should be experimented with so that relevant phrases of varying lengths can be learned. For evaluation, it may be interesting to see if recall and precision vary with how common the journal is in the dataset. Finally, it may be interesting to characterize journals based on how much recall and precision improves when different embeddings fed into the embedding-augmented CNN.

References

A Abrishami and S Aliakbary. 2018. Nncp: A citation count prediction methodology based on deep neural network learning techniques. *arXiv:1809.04365v2 [cs.DL]*.

ASHG 2018. Abstract Submission: Topic Categories. https://www.ashg.org/2018meeting/pages/abstract_topics.shtml. Accessed: 2018-10-27.

Rich Caruana. 1997. Multitask learning. *Kluwer Academic Publishers*.

Citefactor. Impact Factor List 2015. https://www.elsevier.com/connect/whats-the-best-journal-for-my-paper-new-tool-can-help. Accessed: 2018-09-24.

PA Cock, T Antao, JT Chang, BA Chapman, CJ Cox, A Dalke, I Friedberg, T Hamelryck, and F Kauff. 2009. Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25:1422–1423.

NCBI Resource Coordinators. 2017. Database resources of the national center for biotechnology information. *Nucleic Acids Research*, 45:12–17.

howpublished = https://github.com/ducha-aiki/caffenet-benchmark/blob/master/batchnorm.md note = Accessed: 2018-12-5 Dmytro Mishkin, title = BatchNorm.

Elizabeth Ash and Lyndsay Scholefield. What's the best journal for my paper?' New tool can help. https://www.elsevier.com/connect/whats-the-best-journal-for-my-paper-new-tool-can-help. Accessed: 2018-09-24.

K He, X Zhang, S Ren, and J Sun. 2015. Deep residual learning for image recognition. *arXiv:1512.03385 [cs.CV]*.

Y Kim. 2014. Convolutional neural networks for sentence classification. *arXiv:1408.5882v2 [cs.CL]*.

K Kowsari, DE Brown, M Heidarysafa, KJ Meimandi, MS Gerber, and LE Barnes. 2017. Hdltex: Hierarchical deep learning for text classification. *arXiv:1709.08267v2 [cs.LG]*.

D McNamara, P Wong, and K.S. Ng. 2013. Predicting high impact academic papers using citation network features. *Lecture Notes in Computer Science*, 7867.

Sampo Pyysalo, Filip Ginter, Hans Moen, Tapio Salakoski, and Sophia Ananiadou. 2013. Distributional semantics resources for biomedical text processing. In *Proceedings of LBM 2013*, pages 39–44.

Martijn J. Schuemie and Jan A. Kors. 2008. Jane: suggesting journals, finding experts. *Bioinformatics*, 24:727–728.