MEDINFO 2021: One World, One Health – Global Partnership for Digital Innovation
P. Otero et al. (Eds.)

doi:10.3233/SHTI220155

Towards Automated Screening of Literature on Artificial Intelligence in Nursing

Hans Moen[a], Dari Alhuwail[b], Jari Björne[a], Lori Block[c], Sven Celin[d], Eunjoo Jeon[e], Karl Kreiner[d], James Mitchell[f], Gabriela Ožegović[d], Charlene Esteban Ronquillo[c], Lydia Sequeira[g], Jude Tayaben[h], Maxim Topaz[i], Sai Pavan Kumar Veeranki[j], Laura-Maria Peltonen[k]

[a] Department of Computing, University of Turku, Turku, Finland
[b] Information Science Department, Kuwait University, Sabah AlSalem University City, Kuwait
[c] School of Nursing, University of British Columbia Okanagan, Kelowna, Canada
[d] AIT Austrian Institute of Technology, Graz GmbH, Austria
[e] Technology Research, Samsung SDS, Seoul, Republic of Korea
[f] School of Computing and Mathematics, Keele University, United Kingdom
[g] Institute of Health Policy, Management and Evaluation, University of Toronto, (Ontario) Canada
[h] College of Nursing, Benguet State University, Philippines
[i] Columbia University, New York, USA
[j] Institute of Neural Engineering, Graz University of Technology, Graz, Austria
[k] Department of Nursing Science, University of Turku, Turku, Finland

Abstract

We evaluate the performance of multiple text classification methods used to automate the screening of article abstracts in terms of their relevance to a topic of interest. The aim is to develop a system that can be first trained on a set of manually screened article abstracts before using it to identify additional articles on the same topic. Here the focus is on articles related to the topic "artificial intelligence in nursing". Eight text classification methods are tested, as well as two simple ensemble systems. The results indicate that it is feasible to use text classification technology to support the manual screening process of article abstracts when conducting a literature review. The best results are achieved by an ensemble system, which achieves a F1-score of 0.41, with a sensitivity of 0.54 and a specificity of 0.96. Future work directions are discussed.

Keywords:

Natural Language Processing; Nursing; Review

Introduction

Literature reviews are typically conducted for the purpose of summarizing existing scientific literature on a specific topic, or scientific field. The review type and methods depend on the research question at hand [1]. One laborious phase of any literature review is the assessment of article relevance through screening and selection of articles based on title and abstract. Research in nursing has increased exponentially during the last decade and almost 47,000 articles were indexed in PubMed in 2020 alone under ("Nursing"[MeSH] OR nursing) [2]. The increasing need for evidence generation through evidence syntheses would benefit from technologies that reduce the manual labor required to conduct a literature review.

The use of machine learning methods in the form of natural language processing (NLP) and text classification has been shown to be promising when it comes to assisting the screening phase of literature reviews [3-8]. For example, studies show that the specificity scores achieved by such NLP methods in systematic reviews in the field of medicine varies from 0.59 to 0.99 [9-10]. Approaches that have been explored include ensemble learning models (e.g., LightGBM) [3]; comparison of various machine learning algorithms for classification (e.g., support vector machines, naïve Bayes, bagged classification and regression trees) and comparison of different training set strategies (e.g., full data versus downsampling and using inclusion/exclusion decisions from abstracts versus full-text screening) [4]; while several studies have evaluated the performance of off-the-shelf online machine learning and deep learning tools for semi-automated title and abstract screening [5-8].

In an ongoing study, we extracted 4,186 abstracts on the topic of "artificial intelligence (AI) in nursing" for manual screening. Two reviewers manually screened these abstracts and found that 139 (3.3%) should be included in the review. Given the rapid rate at which new literature is being published, we now would like to use the results from the previously conducted screening to train a text classification system that we can use to help us identify and suggest additional relevant articles that have not yet been manually screened. This includes articles that are published on the same topic in the future. The aim of this study is to evaluate the applicability and performance of various text classification methods at the task of automated abstract screening on the research topic. The results could be useful for similar future efforts. Ethical review was not needed as the study was based on data published in scholarly journals.

Methods

Data

The data set used in this study contains 4,186 abstracts, out of which 139 (3.3%) were included (i.e. relevant to the topic of interest). These abstracts were published after 2010 and were obtained from four databases: including PubMed (MEDLINE), CINAHL (Ebsco), Web of Science and IEEE. As search terms,

we used a range of AI and machine learning methods and concept names combined with nursing specific terms. Database specific terms (e.g. MeSH) were also used when appropriate in the search. The abstracts were read and labelled independently by two domain experts to determine their eligibility for inclusion/exclusion based on the predefined criteria. Disagreements were discussed until consensus was reached. Studies included were: experimental or observational studies; qualitative, quantitative and mixed methods approaches; and studies that developed or validated AI technologies applicable to nursing. Studies without explicit description of the relationship between the AI technology and potential impact on nursing practice or education were excluded.

For this experiment, the title and abstract of each paper were concatenated into a single textual representation. The basic preprocessing involved tokenization and lowercasing of the text. The full dataset was split into training (60%), development (15%) and test (25%) sets using stratified sampling. The training set was used to train the methods/models, while the development set was used for hyper parameter optimization. Finally, the test set was used to generate the reported results. To evaluate the methods/models performance we report area under the curve (AUC), sensitivity (SE), specificity (SP), precision (P), recall (R, same as SE), and F1-score - the weighted average of precision and recall. For optimization on the development set we used the F1 metric.

The exploration of different text classification methods was formulated as a small shared task, where the participants took on the responsibility of training and optimizing one method each.

`BERT` and `BioBERT` - Transformer-based language models

BERT is a popular transformer-based language model developed by Google [13]. In this work, the biobert-v1.1 [14] and the bert-base-uncased models were used via the Huggingface Transformers library [15]. Combinations of values between 1×10^{-1} and 1×10^{-15} for both the learning rate and the epsilon parameter were evaluated. Model training was extremely unstable and the resulting F1-scores differed by even 20 percentage points on different runs with the exact same parameters, possibly due to the small dataset size. Thus, three duplicates were trained for each parameter combination, and the best model was chosen out of all the trained models based on development set performance.

`BiLSTM` - Bidirectional long short-term memory network

Long short-term memory (LSTM) networks is a recurrent neural network architecture able to process sequential data (words in this case) in which each decision is influenced by the previous observations [17]. Here we use bidirectional LSTMs, meaning the network reads the input sentence from both directions. As input layer we used an embedding layer initialized with pre-trained word embeddings [18]. For the implementation we used the Keras API [24], which runs on top of Tensorflow [25]. Class weighting was used to tackle the data imbalance issue. Based on a simple grid search, the best performing model on the development set had two bidirectional LSTM layers (dim=500, dropout=0.20), followed by three dense layers (dim of 1000, 500 and 250, with dropout=0.20 between each) and finally the binary decision layer, all with Sigmoid activations.

`CNN` - Convolutional neural network

Convolutional neural networks (CNN) for textual data [11] treat each word in a sentence as a k-dimensional vector, after which convolution operations with various filters are applied to the concatenation. Sets of feature maps from the convolution operations are passed with maximum pooling to a fully connected Softmax layer whose output is the probability distribution over the labels to predict. Text-CNN model parameters include filter sizes of 1,2,3 and 5; 36 filters; dropout probability of 0.1; batch size of 64; learning rate of 0.001; a maximum word count of 200; and a maximum of 50 epochs. The model was initiated with pre-trained word embeddings [12]. Class weighting and early stopping was used as means to avoid overfitting. Random oversampling was also used.

`FastText` - FastText classifier

FastText is a library for learning word embeddings and text classification created by Facebook Inc [16]. It relies on a relatively simple densely connected neural network with the option to use n-gram features in addition to the individual words. To address the data imbalance issue, we here used an iterative method of oversampling. This implemented a random set of positive samples for each negative sample in the data set. Based on a simple parameter grid search, we set the learning rate (lr) at 0.2; dimension (dim) to 100; size of the vector window (ws) to 1; number of epochs to 15 and wordNgrams to 3.

`LinSVM` - Linear support vector classifier

Support Vector Machine (SVM) [19] has been shown to be highly effective in classifying high-dimensional feature spaces such as vectorization of common words. This method was chosen because of its speed and simplicity in implementation. SVM works by iteratively finding the hyperplane that separates two different classes with maximum marginal hyperplane and minimizing the error. SVM performance can be improved with stochastic gradient descent (SGD) learning. Here we used SGDClassifier with Randomized search for choosing the optimal parameters. Parameters: {'penalty': 'l2', 'loss': 'modified_huber', 'learning_rate': 'optimal', 'eta0': 10, 'class_weight': {1: 0.7, 0: 0.3}, 'alpha': 1}. For the implementation we used the scikit-learn library [20]. A term frequency (TF) representation was used to generate the required BoW representation. SMOTE oversampling [23] was used in an attempt to deal with the unbalanced dataset.

`RF` - Random Forest

Random Forest (RF) is an ensemble of decision trees [21]. It relies on a BoW representation of the text. Gradient Boosting Trees (GBT), which builds trees in a sequence based on the performance of the previous trees, were also tested. However, this did not give any improvements relative to RF. RF with 1000 trees were used in this experiment. Oversampling was used as a strategy to deal with the data imbalance issue.

`LR` - Logistic Regression with L1 and L2 constraints

Logistic regression models the probability of each abstract belonging to a particular category (i.e. included or excluded) based on their vectorized representations [22]. Here we used Elastic Net Regression, which linearly combines the $\ell1$ and $\ell2$ penalties of the Lasso and Ridge methods: Ridge regression minimizes regression coefficients for variables with minor

contribution to the outcome. Lasso regression uses an ℓ1 penalty, and similar to Ridge regression, shrinks the coefficient estimates towards zero, forcing some coefficient estimates to be exactly equal to zero when performing variable selection. To convert the abstracts into a vectorized representation we used GloVe word embeddings [12]. Oversampling was used to deal with the imbalanced data issue. Optimization was done by exploring lambda values that gave us the minimum mean cross-validated error (λ=min) and the most regularized model such that error is within one standard error of the minimum (λ=1se). Additionally, we optimized the models based on α values. The parameters that gave the highest performance on the development set were λ= 1se and α = 0.1.

`Random` - Naive baseline

A primitive baseline that randomly selects labels (relevant, non-relevant) was also included as reference.

$\texttt{Combo}_{\texttt{min1}}$ and $\texttt{Combo}_{\texttt{min2}}$ - Ensemble systems

Finally, we also explore some simple ensemble systems which combine the predictions of the different methods (`Random` not included). One ensemble system, $\texttt{Combo}_{\texttt{min1}}$, works by doing a straightforward combination of all method's individual predictions, which labels an abstract as relevant if a minimum of one method has classified this. The other ensemble system, $\texttt{Combo}_{\texttt{min2}}$, requires that a minimum of two methods has classified an abstract as relevant in order to label it as relevant.

Results

The results can be seen in Table 1.

Table 1– Results showing the performance of each method and system on the test set.

Method	AUC	SE/R	SP	P	F1
BERT	0.59	0.20	0.98	0.29	0.24
BioBERT	0.56	0.14	**0.99**	0.26	0.19
BiLSTM	**0.71**	**0.49**	0.94	0.23	0.31
CNN	0.63	0.29	0.97	0.24	0.26
FastText	0.64	0.29	**0.99**	**0.42**	**0.34**
LinSVM	0.52	0.06	0.98	0.10	0.07
RF	0.51	0.06	0.97	0.06	0.06
LR	0.58	0.17	0.98	0.27	0.21
Random	0.54	0.60	0.47	0.04	0.07
$\texttt{Combo}_{\texttt{min1}}$	**0.80**	**0.74**	0.86	0.15	0.25
$\texttt{Combo}_{\texttt{min2}}$	0.75	0.54	**0.96**	**0.33**	**0.41**

The best results among the individual methods in terms of F1-score was achieved with the `FastText` method, with a score of 0.34. As we optimized the methods against the F1-score, this indicates that `FastText` achieved the best balance between precision and recall. The best AUC score was 0.71 with the `BiLSTM` method. `BiLSTM` also achieved the best sensitivity/recall score of 0.49, which indicates the fraction of relevant abstracts found overall. The best precision score was achieved with `FastText`, which tells the fraction of relevant abstracts among those retrieved. All methods (except `Random`) achieved high specificity scores, which reflects their ability to identify the non-relevant abstracts.

When looking at the ensemble systems, $\texttt{Combo}_{\texttt{min1}}$ was able to identify 74% of the relevant abstracts. With this system, one would avoid having to manually screen 84% of the abstracts in the test set, at the cost of missing 26% of the relevant ones. The $\texttt{Combo}_{\texttt{min2}}$ system achieves the highest overall F1 score of 0.41. However, it is only able to correctly identify 54% of the relevant abstracts.

Discussion

Overall, the results indicate that it is feasible to use text classification technology to automate the screening of article abstracts with a nursing relevance when a training set exists. Among the evaluated methods and ensemble systems, $\texttt{Combo}_{\texttt{min1}}$ seems to be the most promising approach. One challenge that we encountered was the relatively small number of relevant abstracts together with the imbalance between relevant and non-relevant abstracts in our dataset. Qin et al. [3] achieved a sensitivity of 0.96 and a specificity of 0.78 when using an ensemble system with four BERT-based models for title and abstract classification. However, in their test dataset the percentage of relevant abstracts was 19.0%, compared to 3.3% in ours. Another challenge is that the topic "AI in nursing" is not well defined. There is no standard way of reporting studies on this topic, thus it can be difficult to assess if technology that can be classified as AI is used from only reading the abstracts.

It is somewhat surprising to observe that the BERT-based models, `BioBERT` in particular, did not perform better on the test set, given its state-of-the-art performance on multiple text classification tasks. The development set `BioBERT` got a F1-score of 0.40, while only 0.19 on the test set (21 pp decrease). A similar drop in performance between development and test sets was observed with `CNN`, which went from a F1-score of 0.45 to 0.26 (19 pp decrease). However, the other methods were relatively stable in their performance across the two sets. Observing that `FastText` performed relatively well is encouraging due to its speed and relatively low complexity and computational costs.

As future work we plan to use an ensemble system akin to $\texttt{Combo}_{\texttt{min}N}$ (probably without some of the least suitable methods) to sift through massive amounts of additional abstracts indexed by PubMed and similar databases, then manually verify the positive predictions. This could provide us with additional relevant articles that were not included in the initially extracted dataset. In addition, this would provide us with a "silver standard" dataset that can be used for retraining to further increase in performance. So far we have optimized our methods against their F1 score, which is the weighted average of precision and recall. However, in the continuation we are considering giving more weight to recall to penalize more when relevant abstracts are not identified. Since the topic "AI in nursing" is not well defined, and since the manual screening was mainly done by reading only the abstracts, we are also considering manual assessment of the full text of the false positives in our current dataset predicted by the system. Further (hyper) parameter optimization of the different methods is also planned. This system can help to identify additional literature on AI in nursing as well as other topics for which manual screening is ongoing or has been previously conducted. This includes identifying new relevant literature published in the future.

Limitations

The limitations of this work include lack of further classification model (hyper) parameter training and imbalanced datasets, which we plan to address in our further work.

Conclusions

We applied eight text classification methods to automatically screen article abstracts and identify articles related to AI in nursing. We found the classification task feasible, but also encountered several challenges, which we will continue to explore in the future. Overall, our final system can be used to enhance and streamline a tedious task of literature screening in literature reviews and to help in identifying relevant literature published in the future.

Acknowledgements

This paper was initiated and written by the members of the Students and Emerging Professionals Special Interest Group of the International Medical Informatics Association. The research was supported by the Academy of Finland (315376).

References

[1] S.S. Samnani, M. Vaska, S. Ahmed, T.C Turin. Review Typology: The Basic Types of Reviews for Synthesizing Evidence for the Purpose of Knowledge Translation. *J Coll Physicians Surg Pak*. (2017), 27(10), 635-641.

[2] National Library of Medicine. www.PubMed.gov. Accessed 19.04.2021

[3] X. Qin, J. Liu, Y. Wang, Y. Liu, K. Deng, Y. Ma, et al. Natural language processing was effective in assisting rapid title and abstract screening when updating systematic reviews. *J. Clin. Epidemiol* 133 (2021), 121- 129.

[4] E. Popoff, M. Besada, J.P. Jansen, S. Cope, S. Kanters. Aligning text mining and machine learning algorithms with best practices for study selection in systematic literature reviews. *Syst Rev* 9 (2020), 293.

[5] A. Gates, M. Gates, D. Da Rosa, S.A. Elliot, J. Pillay, S. Rahman, et al. Decoding semi-automated title-abstract screening: findings from a convenience sample of reviews. *Syst Rev*. 9 (2020), 272.

[6] S.M. Reddy, S. Patel, M. Weyrich, J. Fenton , M. Viswanathan. Comparison of a traditional systematic review approach with review-of reviews and semi-automation as strategies to update the evidence. *Syst Rev*. 9 (2020), 243. doi:10.1186/s13643-020-01450-2.

[7] A.Y. Tsou, J.R. Treadwell, E. Erinoff , K. Schoelles. Machine learning for screening prioritization in systematic reviews: comparative performance of Abstrackr and EPPI-Reviewer. *Syst Rev* 9(2020), 73.

[8] T. Yamada, D. Yoneoka, Y. Hiraike, K. Hino, H. Toyoshiba, A. Shishido, et al. Deep neural network for reducing the screening workload in systematic reviews for clinical guidelines: algorithm validation study. *J Med Internet Res* (2020), 22(12):e22422. doi:10.2196/22422

[9] J. Zimmerman, R.E. Soler, J. Lavinder, et al. Iterative guided machine learning-assisted systematic literature reviews: a diabetes case study. *Syst Rev*. (2021), 10(1), 1-8.

[10] S. Kharawala, A. Mahajan, P. Gandhi. Artificial intelligence in systematic literature reviews: a case for cautious optimism. *J Clin Epidemiol* 19 (2021), S0895-4356(21)00084-6.

[11] Y. Kim. Convolutional neural networks for sentence classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2014), 1746-51.

[12] J. Penninton, R. Socher, C.D. Manning. Glove: Global vectors for word representation. *Proceedings of the 2014 conference on Empirical Methods in Natural Language Processing (EMNLP)* (2014), 1532-43.

[13] J. Devlin, M-W. Chang, K. Lee, K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805v2 (2018).

[14] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C.H. So, J. Kang. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* (2020), 36(4), 1234-1240.

[15] T. Wolf, J. Chaumond, L. Debut, V. Sanh, et al. Transformers: State-of-the-art natural language processing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (2020), 38-45.

[16] A. Joulin, E. Grave, P. Bojanowski, T. Mikolov. Bag of tricks for efficient text classification. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2* (2017), 427-431. Association for Computational Linguistics.

[17] S. Hochreiter, J. Schmidhuber. Long short-term memory. *Neural Computation* (1997), 9(8), 1735-1780.

[18] S. Pyysalo, F. Ginter, H. Moen, T. Salakoski, T. Ananiadou. Distributional semantics resources for biomedical text processing. *Proceedings of LBM* (2013), 39-44.

[19] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. *European conference on machine learning* (1998), 137-142. Springer, Berlin, Heidelberg.

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* (2011), 12, 2825-2830.

[21] A. Liaw, M. Wiener. Classification and regression by randomForest. *R news* (2002), 2(3), 18-22.

[22] G. James, D. Witten, T. Hastie, R. Tibshirani. An introduction to statistical learning, vol 112, p. 18. New York: Springer, 2013.

[23] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *CoRR* (2011), abs/1106.1813.

[24] F. Chollet and others. Keras. GitHub (2015). Available at: https://github.com/fchollet/keras

[25] M. Abadi, A. Agarwal, P. Barham et al. TensorFlow: Large-scale machine learning on heterogeneous systems (2015). Available at: tensorflow.org

Address for correspondence
Hans Moen, hnsmoen(at)gmail.com.