Medical Abbreviation Disambiguation

Yuyao Zhang, McGill
Francis Liu, McGill
Ricardo Alvarez, McGill

Abstract

Hospitals produce millions of clinical notes consisting of annotations containing extensive usage of abbreviations. Due to the ambiguity of these medical abbreviations with respect to their sense, complete and accurate text analysis requires careful identification of the sense that was intended for them. The main objective of our project is to compare different NLP techniques in terms of their capability of finding the correct expansions of ambiguous acronyms in a given corpus. We train 4 different models using a large, annotated corpus curated for abbreviation disambiguation and a database that covers frequently used abbreviations and their expansions. We then disambiguate the sense of abbreviations in the test corpus using the trained models to compare their accuracy.

1 Introduction

Millions of text-based records are produced at hospitals every year. A clinical record usually contains more than 50 different types of medical annotations that may include radiology reports, surgical notes, and discharge summaries. The usage of abbreviations in clinical annotations is extensive as it makes writing the notes much easier. However, the use of abbreviations can also make text ambiguous and understanding the clinical record difficult without knowledge of the sense of all abbreviations in the document. For example, according to the *UMLS 2001AB*[1], the term “TCT” can stand for “thyrocalcitonin”, “tracheal cytotoxin”, “tube coagulase test” and “Transmission computed tomography”. This kind of ambiguous acronym used routinely throughout clinical texts can cause problems for human comprehension and interpretation and the natural language processing system.

Although natural language processing systems have been developed in the clinical domain to unlock information from free text for many years, modern clinical NLP systems still have much room for improvement in acronym disambiguation (Haug et al., 2017)(Moon et al., 2015). Resolving ambiguities in clinical records is a major concern for improving information retrieval (Walsh et al., 2008). Fortunately, this problem is handled well by fully supervised machine learning methods, which achieves a precision of 87.5% on the test dataset (Xu et al., 2012). The main objective of this project is to conduct a supervised training of CNN, logistic regression, random forest, and support vector classifier with model variations on a large-scale corpus of 3,000,000 texts. Our hypothesis is that the neural network model will be the best model for this task and will provide an accuracy of 80% or more because it has non-linear expressive power. We filter the training and validation data to keep only the text part and corresponding labels that indicate the accurate sense of abbreviations in the text. We only keep about 20 most common abbreviation expansions, as higher occurrences make it easier to gather context data and disambiguate. We then train the models with variations, which is explained in detail in the method section. The parameter choice and evaluation of the models are done using the same validation and test set.

In the rest of the report, we introduce some related works, a description of the dataset and methods,

[1] UMLS Knowledge Sources [computer program]. Version 2001 AB, 2004 AA. Bethesda, MD: National Library of Medicine; 2004.

as well as a discussion on the models' performances.

2 Related Work

2.1 MeDaL

MeDaL (Medical Dataset for Abbreviation Disambiguation for Natural Language Understanding) is a large public dataset that was assembled from PubMed abstracts. Wen et al. assembled it for their work on medical abbreviation disambiguation (2020). Their work focuses on using disambiguation as a pre-training task to transfer learning onto other downstream clinical tasks such as mortality prediction and diagnosis prediction. Therefore, they focus on evaluating their models on the success criteria of those downstream tasks instead of only the accuracy of their abbreviation disambiguation. This goal difference is where our works differ as our evaluation metrics center around accuracy and F1 scores relative to the predicted and target abbreviation expansions. Moreover, their goal influences their model choices (LSTM, transformers) which differ from ours (neural network, logistic regression, random forest, SVM).

2.2 A Deep Database of Medical Abbreviations and Acronyms for Natural Language Processing

This paper by Liu et al. explores the creation of a "Meta-Inventory " of medical abbreviations and their expansions (2021). Their work centers on compiling medical abbreviations and expansions from different sources. The paper's significant contribution comes from normalizing abbreviations with the same meaning but with linguistic variation into common abbreviations and concepts. Their work informs our work by giving perspective on the linguistic variation between abbreviations and by influencing our design decisions. For example, disambiguating "AT" as "anticoagulating" instead of "anticoagulation" should be a correct prediction if we apply lemmatization. However, this might have unwanted consequences as for example, disambiguating "DC" and as "dyschondroplasia" and lemmatizing would be a false positive if the correct prediction is "dyschondrosteosis" as these words have a distinct meaning.

2.3 BERT-based Acronym Disambiguation

Pan et al. (2021) explored BERT-based models, defining the acronym disambiguation task as a binary classification task. In other words, for each potential expansion of an acronym, the model predicted a probability of that expansion being the correct one and selected the one with the highest probability. However, a list of potential matches was not always guaranteed. In our report, we tested the model while hiding this information as well. Furthermore, it mentioned that neural network-based techniques typically required a lot of training data. In our research, this was further justified.

2.4 Abbreviation and Acronym Disambiguation in Clinical Discourse

In *Abbreviation and Acronym Disambiguation in Clinical Discourse*, the authors collect contextual data at the Mayo Clinic and only keep the random samples in which one or more of the 8 acronyms occurred (Pakhomov et al., 2005). In our case, we use a dataset collected from medical articles and consider all abbreviations. The authors experiment with fully supervised and semi-supervised approaches, but fully supervised learning is only used to establish the upper bound for subsequent evaluation of the semi-supervised learning, which is the main contribution of their experiment. Unlike the authors, we focus solely on fully supervised learning. During the phase of semi-supervised learning, the authors collect context for each sense of acronym and generate context vectors of lexical items and their frequency. They then compute the cosine between the context vectors and select the vector with the largest cosine to be the sense of the acronym. This is similar to our approach as we also apply vectorization on tokens of the articles, but the classification sense of acronym is done independently by several models.

3 Method

3.1 Dataset and preprocessing

The dataset used for our experiment is the MeDaL dataset assembled by Wen et al. It contains 15 GB of medical articles that use abbreviations, the index location of the abbreviation in the article, and the corresponding abbreviation expansion. The dataset came already split into training, validation, and testing data. The data preprocessing tokenized the articles simply by space characters. This method was prioritized over more sophisticated ones as the dataset's abbreviation location index was already tokenized in this way. In order to better condition on the medical context around the abbreviation, stop words were removed.

3.2 Neural Network

Some neural network (NN) structures were built to fit the data. Admittedly, with only the top 20 frequent expansions for each acronym, the scale of the dataset did not reach the million level. It was possible that a NN's potential was not fully reached. Nonetheless, because the expressive power of NN is strong, we still tried this method. The input for the NN model was the same as described in the feature engineering part to ensure a fair starting point as the rest of the models. The data was then passed into the fully connected layers. In the evaluation phase, the NN predicted the probability for every potentially corresponding label, choosing the one with the highest. Different structures varying from 0.1, 0.01, or 0.001 learning rate, 1 to 5 layer(s) of 16, 32, 64, 128, or 256 neurons were generated and evaluated to find the best settings.

3.3 Logistic Regression

A multinomial logistic regression model was created to predict the abbreviation expansion (classes) from the articles' abbreviations. Vectorization was first applied to the tokens of the articles and used to fit the model along with the corresponding expansions. When predicting, the model assigned probabilities to the potential expansions**.** The final prediction was made by choosing the expansion that had the highest probability and a matching abbreviation as the predicted expansion. To choose the best parameters for the model, GridSearchCV was performed with validation data and several candidates for both the inverse of regularization strength parameter (multiples of 10 from 0.001 to 100) and solver type (all multinomial solvers, namely, 'sag', 'saga', 'lbfgs' and 'newton-cg').

3.4 Random Forest Classifier

A random forest classifier model was created to predict abbreviation expansions (classes) from medical articles' tokens. The same vectorization process that was applied with the logistic regression model was used. Thus, the model was trained with the vectorization of the article tokens and their corresponding expansion. GridSearchCV was performed to maximize the model's disambiguation accuracy. Different candidates were combined for the n_estimators (100, 1000, or 10000), max_features (auto or log2), and max_depth (7 or 8) parameters. These parameters represent the number of trees, the function to determine the number of features used for the tree split, and the maximum depth of the tree, respectively. The scope of the candidates was informed by initial parameter selection validation runs done with a smaller training dataset.

3.5 SVM

A multiclass support vector machine (SVM) was trained for each abbreviation using hyperparameter values, 'C' and gamma, where C is a regularization parameter that decides the trade-off between the correct classification of acronym expansion against maximization of margin and gamma is an inverse of the radius of the influence of each abbreviation expansion that defines how far the influence of reaches. The tokens of articles in the training set were transformed into feature vectors, which were used along with abbreviation expansions as input of the support vector classifier. During the training phase, this model performed 5-fold cross-validation to find the best hyperparameters. To efficiently perform non-linear classification, we used the Kernel method that enabled SVM to operate in a high-dimensional, implicit feature space by computing the inner products between the distances of all pairs of acronym expansion in the feature space. In this project, we used Radial Basis Function as the kernel function.

4 Results

4.1 Neural Network

The following hyperparameters gave the best results in terms of accuracy and F1 score for the neural network model: Learning rate of 0.01, one fully connected layer of 64 neurons with ReLU activation function, and one output layer with a softmax activation function.

4.2 Logistic Regression

The results of the cross-validation informed the authors that the best parameters for the logistic regression model were an inverse regularization strength (or 'C') of 10 and a solver of type 'saga'. The smaller the 'C' parameter is, the more regularization there will be, and the risk of overfitting will be reduced. In the case of the solver, ‘saga’ or Stochastic Average Gradient descent is the best multinomial solver to maximize accuracy.

4.3 Random Forest Classifier

The results of the cross-validation informed the authors that the best parameters for the random forest model were a maximum feature function of log base 2, 1000 trees, and a tree depth of 8. This combination of parameters maximized model accuracy.

4.4 SVM

The results of cross-validation showed that SVM gave the best accuracy using the following hyperparameters: a regularization parameter (C) of 10 and an inverse of the radius of data points’ influence selected by the model as support vectors (gamma) of 0.01. This combination of hyperparameters gave the best trade-off between misclassification and soft margins, it also produced the highest validation and test accuracy.

Model	Neural Network	Logistic Regression	Random Forest	SVM
Testing accuracy	65.04%	81.93%	58.66%	80.62%
Testing F1	64.69%	81.68%	55.10%	80.29%

Table 1: Testing accuracy and F1 scores per model.

5 Discussion

5.1 Neural Network

It was first speculated that 3 dense layers of 256, 128, and 64 neurons should perform the best. Given the results, it was reasonable to believe that such settings caused overfitting, resulting in only 0.586 accuracy and 0.584 F1 score. Deep neural nets were not necessary and even counterproductive for our acronym disambiguation task. Moreover, when the number of neurons was more than 64, adding more neurons in that layer basically did not affect the performance, which in turn demonstrated that one dense layer of 128 neurons was enough. The results also verified that neural methods typically do not reach their full potential with limited data (with the top 20 frequent expansion for each acronym, the scale of the dataset was way below million level), as mentioned by Pan et al. and as seen in the course as well (2021).

5.2 Logistic Regression

A concept seen in the course about logistic regression was how difficult it was to include features as one must give their probability distribution. The design decision to tokenize and vectorize the text was made to facilitate this process and lessen the feature engineering. Moreover, the model was implemented with a heuristic that assigned probabilities to all potential expansions but had to choose the one with the highest probability and the same abbreviation as the abbreviation being predicted. This heuristic allows the authors to extend the model from a binary to a multi-class problem. Therefore, both design decisions built on material seen in the course and pushed the authors’ reflections forward. Perhaps a more rigorous model could expand on this model.

5.3 Random Forest Classifier

The random forest classifier performed significantly worse than the rest of the models. This might be because the model was not based on supporting literature or examples. Therefore, the parameters were determined on a trial and error basis. It was, however, an interesting model to explore in comparison to the other ones.

5.4 SVM

Unlike a linear model that classifies data in 2D space, a high dimensional support vector machine computes the relationships between the observations in higher dimension and finds the support vector classifier. This makes the model sensitive to hyperparameter change, which can easily lead to overfitting or inefficient training time. SVM is highly sensitive to gamma. When gamma was too small, the model was too constrained and was not able to capture the complexity of the abbreviation-expansion pair, and the resulting model would behave like a linear model. If gamma was too large, the model was overfitting. We decided to use low gamma value and high C value, as large C helped the model to capture complex abbreviation-expansion relationships to prevent underfitting. Unfortunately, this hyperparameter combination led to longer training time, as a virtual machine with 8GB memory and 12 CPU cores spent 400 seconds to train the model using only 3,000,000 data instances. Since training time wasn’t the focus of this project, we eventually chose performance over runtime.

5.5 Acronym Disambiguation without a Dictionary

Sometimes, multiple acronyms refer to the same expansion. For example, ConvNet and CNN are both acronyms of Convolutional Neural Network. It is possible that a full name does not have a unique standard abbreviation. This problem may become more serious as more words and acronyms are invented in the future.

In our previous models, a dictionary mapping the acronyms and their potential expansions were always provided to the model. In other words, the model knew the range of the answer; in the major settings, it needed to choose one from the 20 expansions, since it was guaranteed that the correct answer was among the 20. In an attempt to test how comprehensive a model could be, we also conducted the experiment while hiding the dictionary. Specifically, the model needed to finish a multiclass classification task with over 20000 classes in total.

Unfortunately, the results were not as satisfactory. In the NN settings, the model only achieved 13.5% accuracy. It was concluded that, to our best knowledge, the information provided by the potential expansions was indispensable. Passing the information to the model and letting it know the answer range described as Pan et al. 2021 is necessary.

5.6 Limitations in General

The limitations of our models, in general, were a lack of computational power as the validation across many parameters significantly delayed the results, analysis, and ensuing critical thinking about the parameters. Also, to reduce the computational expense of our models, our experiment focuses on training our models on the articles of the top 20 most common abbreviations in the training set and disambiguating them in the testing set. This choice might limit the models' abilities to find patterns among several abbreviations' contexts.

6 Conclusion

To conclude, our hypothesis was not verified as the best model was the logic regression model that provided an accuracy score of 81.93%. Our work can provide insights into the implementation difficulties and pros and cons of different models in abbreviation disambiguation. It can be extended with ensemble learning and applications in different fields such as the mental health field.

7 Statement of Contributions

Yuyao Zhang (260832483) contributed to the implementation and evaluation of the neural network model and the experiment on the acronym disambiguation without a dictionary.

Francis Liu (260836334) contributed to the implementation and evaluation of the SVM model. Wrote abstract, introduction, 3.4 Abbreviation and Acronym Disambiguation in Clinical Discourse in related work section and all the parts about SVM in the report.

Ricardo Alvarez (260926136) contributed by finding and writing about the MeDaL and Meta-Investory bodies of work, implementing the logistic regression and random forest model as well as writing the corresponding sections and conclusion.

References

Haug, P.J., Christensen, L., Gundersen, M., Clemons, B., Koehler, S. and Bauer, K., 1997. A natural language parsing system for encoding admitting diagnoses. In *Proceedings of the AMIA Annual Fall Symposium* (p. 814). American Medical Informatics Association.

Liu, L.G., Grossman, R.H., Mitchell, E.G., Weng, C., Natarajan, K., Hripcsak, G. and Vawdrey, D.K., 2021. A deep database of medical abbreviations and acronyms for natural language processing. *Scientific Data*, *8*(1), pp.1-9.

Moon, S., McInnes, B. and Melton, G.B., 2015. Challenges and practical approaches with word sense disambiguation of acronyms and abbreviations in the clinical domain. *Healthcare informatics research*, *21*(1), pp.35-42.

Pakhomov, S., Pedersen, T. and Chute, C.G., 2005. Abbreviation and acronym disambiguation in clinical discourse. In *AMIA annual symposium proceedings* (Vol. 2005, p. 589). American Medical Informatics Association.

Pan, C., Song, B., Wang, S. and Luo, Z., 2021. BERT-based Acronym Disambiguation with Multiple Training Strategies. *arXiv preprint arXiv:2103.00488*.

Walsh, K.E. and Gurwitz, J.H., 2008. Medical abbreviations: writing little and communicating less. *Archives of disease in childhood*, *93*(10), pp.816-817.

Wen, Z., Lu, X.H. and Reddy, S., 2020. MeDAL: Medical abbreviation disambiguation dataset for natural language understanding pretraining. *arXiv preprint arXiv:2012.13978*.

Xu, H., Stetson, P.D. and Friedman, C., 2012. Combining corpus-derived sense profiles with estimated frequency information to disambiguate clinical abbreviations. In *AMIA annual symposium proceedings* (Vol. 2012, p. 1004). American Medical Informatics Association.