Automated Disease Normalization with Low Rank Approximations

Robert Leaman **Zhiyong Lu**
National Center for Biotechnology Information
National Library of Medicine
{robert.leaman, zhiyong.lu}@nih.gov

Abstract

While machine learning methods for named entity recognition (mention-level detection) have become common, machine learning methods have rarely been applied to normalization (concept-level identification). Recent research introduced a machine learning method for normalization based on pairwise learning to rank. This method, DNorm, uses a linear model to score the similarity between mentions and concept names, and has several desirable properties, including learning term variation directly from training data. In this manuscript we employ a dimensionality reduction technique based on low-rank matrix approximation, similar to latent semantic indexing. We compare the performance of the low rank method to previous work, using disease name normalization in the NCBI Disease Corpus as the test case, and demonstrate increased performance as the matrix rank increases. We further demonstrate a significant reduction in the number of parameters to be learned and discuss the implications of this result in the context of algorithm scalability.

1 Introduction

The data necessary to answer a wide variety of biomedical research questions is locked away in narrative text. Automating the location (named entity recognition) and identification (normalization) of key biomedical entities (Doğan et al., 2009; Névéol et al., 2011) such as diseases, proteins and chemicals in narrative text may reduce curation costs, enable significantly increased scale and ultimately accelerate biomedical discovery (Wei et al., 2012a).

Named entity recognition (NER) techniques have typically focused on machine learning methods such as conditional random fields (CRFs), which have provided high performance when coupled with a rich feature approach. The utility of NER for biomedical end users is limited, however, since many applications require each mention to be normalized, that is, identified within a specified controlled vocabulary.

The normalization task has been highlighted in the BioCreative challenges (Hirschman et al., 2005; Lu et al., 2011; Morgan et al., 2008), where a variety of methods have been explored for normalizing gene names, including string matching, pattern matching, and heuristic rules. Similar methods have been applied to disease names (Doğan & Lu, 2012b; Kang et al., 2012; Névéol et al., 2009) and species names (Gerner et al., 2010; Wei et al., 2012b), and the MetaMap program is used to locate and identify concepts from the UMLS MetaThesaurus (Aronson, 2001; Bodenreider, 2004).

Machine learning methods for NER have provided high performance, enhanced system adaptability to new entity types, and abstracted many details of specific rule patterns. While machine learning methods for normalization have been explored (Tsuruoka et al., 2007; Wermter et al., 2009), these are far less common. This is partially due to the lack of appropriate training data, and also partially due to the need for a generalizable supporting framework.

Normalization is frequently decomposed into the sub-tasks of candidate generation and disambiguation (Lu et al., 2011; Morgan et al., 2008). During candidate generation, the set of concept names is constrained to a set of possible matches using the text of the mention. The primary difficulty addressed in candidate generation is term variation: the need to identify terms which are semantically similar but textually distinct (e.g. “nephropathy” and “kidney disease”). The disambiguation step then differentiates between the different candidates to remove false positives, typically using the context of the mention and the article metadata.

Proceedings of the 2014 Workshop on Biomedical Natural Language Processing (BioNLP 2014), pages 24–28,
Baltimore, Maryland USA, June 26-27 2014. ©2014 Association for Computational Linguistics

Recently, Leaman et al. (2013a) developed an algorithm (DNorm) that directly addresses the term variation problem with machine learning, and used diseases – an important biomedical entity – as the first case study. The algorithm learns a similarity function between mentions and concept names directly from training data using a method based on pairwise learning to rank. The method was shown to provide high performance on the NCBI Disease Corpus (Doğan et al., 2014; Doğan & Lu, 2012a), and was also applied to clinical notes in the ShARe / CLEF eHealth task (Suominen et al., 2013), where it achieved the highest normalization performance out of 17 international teams (Leaman et al., 2013b). The normalization step does not consider context, and therefore must be combined with a disambiguation method for tasks where disambiguation is important. However, this method provides high performance when paired with a conditional random field system for NER, making the combination a step towards fully adaptable mention recognition and normalization systems.

This manuscript adapts DNorm to use a dimensionality reduction technique based on low rank matrix approximation. This may provide several benefits. First, it may increase the scalability of the method, since the number of parameters used by the original technique is proportional to the square of the number of unique tokens. Second, reducing the number of parameters may, in turn, improve the stability of the method and improve its generalization due to the induction of a latent "concept space," similar to latent semantic indexing (Bai et al., 2010). Finally, while the rich feature approach typically used with conditional random fields allows it to partially compensate for out-of-vocabulary effects, DNorm ignores unknown tokens. This reduces the ability of the model to generalize, due to the zipfian distribution of text (Manning & Schütze, 1999), and is especially problematic in text which contains many misspellings, such as consumer text. Using a richer feature space with DNorm would not be feasible, however, unless the parameter scalability problem is resolved.

In this article we expand the DNorm method in a pilot study on feasibility of using low rank approximation methods for disease name normalization. To make this work comparable to the previous work on DNorm, we again employed the NCBI Disease Corpus (Doğan et al., 2014). This corpus contains nearly 800 abstracts, split into training, development, and test sets, as described in Table 1. Each disease mention is annotated for span and concept, using the MEDIC vocabulary (Davis et al., 2012), which combines MeSH® (Coletti & Bleich, 2001) and OMIM® (Amberger et al., 2011). The average number of concepts for each name in the vocabulary is 5.72. Disease names exhibit relatively low ambiguity, with an average number of concepts per name of 1.01.

Subset	Abstracts	Mentions	Concepts
Training	593	5145	670
Development	100	787	176
Test	100	960	203

Table 1. Descriptive statistics for the NCBI Disease Corpus.

2 Methods

DNorm uses the BANNER NER system (Leaman & Gonzalez, 2008) to locate disease mentions, and then employs a ranking method to normalize each mention found to the disease concepts in the lexicon (Leaman et al., 2013a). Briefly, we define $\mathcal{T}$ to be the set of tokens from both the disease mentions in the training data and the concept names in the lexicon. We stem each token in both disease mentions and concept names (Porter, 1980), and then convert each to TF-IDF vectors of dimensionality $|\mathcal{T}|$, where the document frequency for each token is taken to be the number of names in the lexicon containing it (Manning et al., 2008). All vectors are normalized to unit length. We define a similarity score between mention vector m and name vector n, $score(m,n)$, and each mention is normalized by iterating through all concept names and returning the disease concept corresponding to the one with the highest score.

In previous work, $score(m,n) = m^\mathsf{T}Wn$, where W is a weight matrix and each entry w_{ij} represents the correlation between token t_i appearing in a mention and token t_j appearing in a concept name from the lexicon. In this work, however, we set W to be a low-rank approximation of the form $W = U^\mathsf{T}V + I$, where U and V are both $r \times |\mathcal{T}|$ matrices, r being the rank (number of linearly independent rows), and $r \ll |\mathcal{T}|$ (Bai et al., 2010).

For efficiency, the low-rank scoring function can be rewritten and evaluated as $score(m,n) = (Um)^\mathsf{T}(Vn) + m^\mathsf{T}n$, allowing the respective Um and Vn vectors to be calculated once and then reused. This view provides an intuitive explanation of the purpose of the U and V matrices: to convert the sparse, high-dimensional mention and concept name vectors (m and n) into dense, low dimensional vectors (as Um and Vn). Under this interpretation, we found that performance improved if each Um and Vn vector was renormalized to unit length.

This model retains many useful properties of the original model, such as the ability to represent both positive and negative correlations between tokens, to represent both synonymy and polysemy, and to allow the token distributions between the mentions and the names to be different. The new model also adds one important additional property: the number of parameters is linear in the number of unique tokens, potentially enabling greater scalability.

2.1 Model Training

Given any pair of disease names where one (n^+) is for c^+, the correct disease concept for tion m, and the other, n^-, is for c^-, an incorrect concept , we would like to update the weight matrix W so that $m^\intercal W n^+ > m^\intercal W n^-$. Following Leaman et al. (2013a), we iterate through each $\langle m, c^+, c^- \rangle$ tuple, selecting $n^+$and n^-as the name for c^+ and c^-, respectively, with the highest similarity score to m, using stochastic gradient descent to make updates to W. With a dense weight matrix W, the update rule is: if $m^\intercal W n^+ - m^\intercal W n^- < 1$, then W is updated as $W \leftarrow W + \eta(m(n^+)^\intercal - m(n^-)^\intercal)$, where η is the learning rate, a parameter controlling the size of the change to W. Under the low-rank approximation, the update rules are: if $m^\intercal W n^+ - m^\intercal W n^- < 1$, then U is updated as $U \leftarrow U + \eta V(n^+ - n^-)m^\intercal$, and V is updated as $V \leftarrow V + \eta U m(n^+ - n^-)^\intercal$, noting that the updates are applied simultaneously (Bai et al., 2010). Overfitting is avoided using a holdout set, using the average of the ranks of the correct concept as the performance measurement, as in previous work.

We initialize U using values chosen randomly from a normal distribution with mean 0 and standard deviation 1. We found it useful to initialize V as $U^\intercal$, since this causes the representation for disease mentions and disease names to initially be the same.

We employed an adaptive learning rate using the schedule $\eta_k = \eta_0 \frac{\tau}{\tau + k}$, where k is the iteration, η_0 is the initial learning rate, and τ is the discount (Finkel et al., 2008). We used an initial learning rate of $\eta_0 = 10^{-7}$. This is much lower than reported by Leaman et al. (2013a), since we found that higher values caused the training to found that higher values caused the training to diverge. We used a discount parameter of $\tau = 5$, so that the learning rate is equal to one half the initial rate after five iterations.

3 Results

Our results were evaluated at the abstract level, allowing comparison to the previous work on DNorm (Leaman et al., 2013a). This evaluation considers the set of disease concepts found in the abstract, and ignores the exact location(s) where each concept was found. A true positive consists of the system returning a disease concept annotated within the NCBI Disease Corpus, and the number of false negatives and false positives are defined similarly. We calculated the precision, recall and F-measure as follows:

$$p = \frac{tp}{tp + fp} \quad r = \frac{tp}{tp + fn} \quad f = \frac{2pr}{p + r}$$

We list the micro-averaged results in Table 2.

Rank	Precision	Recall	F-measure
50	0.648	0.671	0.659
100	0.673	0.685	0.679
250	0.697	0.697	0.697
500	0.702	0.700	0.701
(Full)	0.828	0.819	0.809

Table 2. Performance measurements for each model on the NCBI Disease Test set. Full corresponds with the full-rank matrix used in previous work.

4 Discussion

There are two primary trends to note. First, the performance of the low rank models is about 10%-15% lower than the full rank model. Second, there is a clear trend towards higher precision and recall as the rank of the matrix increases. This trend is reinforced in Figure 1, which shows the learning curve for all models. These describe the performance on the holdout set after each iteration through the training data, and are measured using the average rank of the correct concept in the holdout set, which is dominated by a small number of difficult cases.

Using the low rank approximation, the number of parameters is equal to $2 \times r \times |\mathcal{T}|$. Since r is fixed and independent of $|\mathcal{T}|$, the number of parameters is now linear in the number of tokens, effectively solving the parameter scalability problem. Table 3 lists the number of parameters for each of the models used in this study.

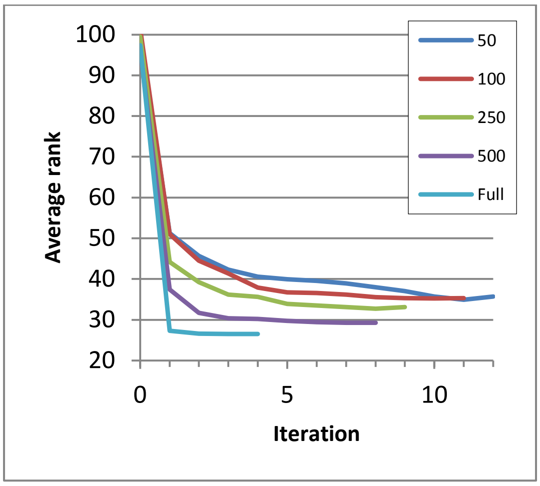

Figure 1. Learning curves showing holdout performance at each iteration through the training data.

Rank	Parameters
50	1.8×10^6
100	3.7×10^6
250	9.1×10^6
500	1.8×10^7
(Full)	3.3×10^8

Table 3. Number of model parameters for each variant, showing the low rank methods using 1 to 2 orders of magnitude fewer parameters.

There are two trade-offs for this improvement in scalability. First, there is a substantial performance reduction, though this might be mitigated somewhat in the future by using a richer feature set – a possibility enabled by the use of the low rank approximation. Second, training and inference times are significantly increased; training the largest low-rank model ($r = 500$) required approximately 9 days, though the full-rank model trains in under an hour.

The view that the U and V matrices convert the TF-IDF vectors to a lower dimensional space suggests that the function of U and V is to provide word embeddings or word representations – a vector space where each word vector encodes its relationships with other words. This further suggests that one way to provide higher performance may be to take advantage of unsupervised pre-training (Erhan et al., 2010). Instead of initializing U and V randomly, they could be initialized using a set of word embeddings trained on a large amount of biomedical text, such as with neural network language models (Collobert & Weston, 2008; Mikolov et al., 2013).

5 Conclusion

We performed a pilot study to determine whether a low rank approximation may increase the scalability of normalization using pairwise learning to rank. We showed that the reduction in the number of parameters is substantial: it is now linear to the number of tokens, rather than proportional to the square of the number of tokens. We further observed that the precision and recall increase as the rank of the matrices is increased.

We believe that further performance increases may be possible through the use of a richer feature set, unsupervised pre-training, or other dimensionality reduction techniques including feature selection or L_1 regularization (Tibshirani, 1996). We also intend to apply the method to additional entity types, using recently released corpora such as CRAFT (Bada et al., 2012).

Acknowledgments

The authors would like to thank the anonymous reviewers for their helpful suggestions. This research was supported by the NIH Intramural Research Program, National Library of Medicine.

References

Amberger, J., Bocchini, C., & Hamosh, A. (2011). A new face and new challenges for Online Mendelian Inheritance in Man (OMIM(R)). *Hum Mutat, 32*(5), 564-567.

Aronson, A. R. (2001). *Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program*. In Proceedings of the AMIA Symposium, 17-21.

Bada, M., Eckert, M., Evans, D., Garcia, K., Shipley, K., Sitnikov, D., et al. (2012). Concept annotation in the CRAFT corpus. *BMC Bioinformatics, 13*, 161.

Bai, B., Weston, J., Grangier, D., Collobert, R., Sadamasa, K., Qi, Y. J., et al. (2010). Learning to rank with (a lot of) word features. *Inform. Retrieval, 13*(3), 291-314.

Bodenreider, O. (2004). The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Res, 32*, D267-270.

Coletti, M. H., & Bleich, H. L. (2001). Medical subject headings used to search the biomedical literature. *J Am Med Inform Assoc, 8*(4), 317-323.

Collobert, R., & Weston, J. (2008). *A unified architecture for natural language processing: deep neural networks with multitask learning*. In Proceedings of the ICML, 160-167.

Davis, A. P., Wiegers, T. C., Rosenstein, M. C., & Mattingly, C. J. (2012). MEDIC: a practical disease vocabulary used at the Comparative

Toxicogenomics Database. *Database, 2012*, bar065.

Doğan, R. I., Leaman, R., & Lu, Z. (2014). NCBI disease corpus: A resource for disease name recognition and concept normalization. *J Biomed Inform, 47*, 1-10.

Doğan, R. I., & Lu, Z. (2012a). *An improved corpus of disease mentions in PubMed citations*. In Proceedings of the ACL 2012 Workshop on BioNLP, 91-99.

Doğan, R. I., & Lu, Z. (2012b). *An Inference Method for Disease Name Normalization*. In Proceedings of the AAAI 2012 Fall Symposium on Information Retrieval and Knowledge Discovery in Biomedical Text, 8-13.

Doğan, R. I., Murray, G. C., Névéol, A., & Lu, Z. (2009). Understanding PubMed user search behavior through log analysis. *Database (Oxford), 2009*, bap018.

Erhan, D., Bengio, Y., Courville, A., Manzagol, P.-A., Vincent, P., & Bengio, S. (2010). Why does unsupervised pre-training help deep learning? *J. Machine Learning Res., 11*, 625-660.

Finkel, J. R., Kleenman, A., & Manning, C. D. (2008). *Efficient, Feature-based, Conditional Random Field Parsing*. In Proceedings of the 46th Annual Meeting of the ACL, 959-967.

Gerner, M., Nenadic, G., & Bergman, C. M. (2010). LINNAEUS: a species name identification system for biomedical literature. *BMC Bioinformatics, 11*, 85.

Hirschman, L., Colosimo, M., Morgan, A., & Yeh, A. (2005). Overview of BioCreAtIvE task 1B: normalized gene lists. *BMC Bioinformatics, 6 Suppl 1*, S11.

Kang, N., Singh, B., Afzal, Z., van Mulligen, E. M., & Kors, J. A. (2012). Using rule-based natural language processing to improve disease normalization in biomedical text. *J. Am. Med. Inform. Assoc., 20*, 876-881.

Leaman, R., Doğan, R. I., & Lu, Z. (2013a). DNorm: Disease name normalization with pairwise learning-to-rank. *Bioinformatics, 29*(22), 2909-2917.

Leaman, R., & Gonzalez, G. (2008). BANNER: an executable survey of advances in biomedical named entity recognition. *Pac. Symp. Biocomput.*, 652-663.

Leaman, R., Khare, R., & Lu, Z. (2013b). *NCBI at 2013 ShARe/CLEF eHealth Shared Task: Disorder Normalization in Clinical Notes with DNorm.* In Working Notes of the Conference and Labs of the Evaluation Forum Valencia, Spain.

Lu, Z., Kao, H. Y., Wei, C. H., Huang, M., Liu, J., Kuo, C. J., et al. (2011). The gene normalization task in BioCreative III. *BMC Bioinformatics, 12 Suppl 8*, S2.

Manning, C., & Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*: Massachusetts Institute of Technology.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*: Cambridge University Press.

Mikolov, T., Yih, W.-t., & Zweig, G. (2013). *Linguistic Regularities in Continuous Space Word Representations*. In Proceedings of the 2013 Conference of the NAACL-HLT, 746-751.

Morgan, A. A., Lu, Z., Wang, X., Cohen, A. M., Fluck, J., Ruch, P., et al. (2008). Overview of BioCreative II gene normalization. *Genome Biol., 9 Suppl 2*, S3.

Névéol, A., Doğan, R. I., & Lu, Z. (2011). Semi-automatic semantic annotation of PubMed queries: a study on quality, efficiency, satisfaction. *J Biomed Inform, 44*(2), 310-318.

Névéol, A., Kim, W., Wilbur, W. J., & Lu, Z. (2009). *Exploring two biomedical text genres for disease recognition*. In Proceedings of the ACL 2009 BioNLP Workshop, 144-152.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program, 14*, 130-137.

Suominen, H., Salanterä, S., Velupillai, S., Chapman, W., Savova, G., Elhadad, N., et al. (2013). Overview of the ShARe/CLEF eHealth Evaluation Lab 2013. In P. Forner, H. Müller, R. Paredes, P. Rosso & B. Stein (Eds.), *Information Access Evaluation. Multilinguality, Multimodality, and Visualization* (Vol. 8138, pp. 212-231): Springer Berlin Heidelberg.

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society Series B-Methodological, 58*(1), 267-288.

Tsuruoka, Y., McNaught, J., Tsujii, J., & Ananiadou, S. (2007). Learning string similarity measures for gene/protein name dictionary look-up using logistic regression. *Bioinformatics, 23*(20), 2768-2774.

Wei, C. H., Harris, B. R., Li, D., Berardini, T. Z., Huala, E., Kao, H. Y., et al. (2012a). Accelerating literature curation with text-mining tools: a case study of using PubTator to curate genes in PubMed abstracts. *Database (Oxford), 2012*, bas041.

Wei, C. H., Kao, H. Y., & Lu, Z. (2012b). SR4GN: a species recognition software tool for gene normalization. *PLoS One, 7*(6), e38460.

Wermter, J., Tomanek, K., & Hahn, U. (2009). High-performance gene name normalization with GeNo. *Bioinformatics, 25*(6), 815-821.