Unsupervised Resolution of Acronyms and Abbreviations in Nursing Notes Using Document-Level Context Models

Katrin Kirchhoff
Department of Electrical Engineering
University of Washington
kk2@u.washington.edu

Anne M. Turner
Department of Health Services
Department of Biomedical Informatics
and Medical Education
University of Washington
amturner@uw.edu

Abstract

Automatic simplification of clinical notes continues to be an important challenge for NLP systems. A frequent obstacle to developing more robust NLP systems for the clinical domain is the lack of annotated training data. This study investigates unsupervised techniques for one key aspect of medical text simplification, viz. the expansion and disambiguation of acronyms and abbreviations. Our approach combines statistical machine translation with document-context neural language models for the disambiguation of multi-sense terms. In addition we investigate the use of mismatched training data and self-training. These techniques are evaluated on nursing progress notes and obtain a disambiguation accuracy of 71.6% without any manual annotation effort.

1 Introduction

As part of a general trend towards patient-centered care many healthcare systems in the U.S. are starting to provide patients with expanded access to clinical notes, often through patient portals connected to their electronic medical record (EMR) systems. Recent studies, such as the OpenNotes project (Delbanco et al., 2012), have found that that patients with access to their health records are more involved in their care and have a better understanding of their treatment plan (Esch et al., 2016; Wolff et al., 2016). However, medical notes often contain complex technical language and medical jargon, requiring patients to seek additional help for linguistic clarification (Walker et al., 2015). Natural language processing (NLP) has the potential to bridge the gap between increased access to medical information and the lack of domain-specific medical training on the patient side. However, in spite of previous work in this area, medical text simplification systems are still not sufficiently mature to be routinely deployed in practice. One problem is the large variety of medical sub-disciplines and document types that need to be covered; another is the lack of annotated training data, often due to constraints on data sharing for reasons of patient privacy.

In this study we investigate *unsupervised* statistical NLP techniques to address one key aspect of medical text simplification, viz. the expansion of medical acronyms and abbreviations (AAs). In addition to text simplification, AA resolution can also help a variety of downstream information extraction tasks. While AA resolution has been studied extensively in the biomedical domain, studies on clinical text are comparatively rare. Moreover, most previous studies use traditional supervised machine learning techniques, consisting of feature extraction and supervised classifiers such as naive Bayes or Support Vector Machines (SVMs) that utilize a carefully developed AA sense inventory and a large amount of hand-annotated ground-truth data. In spite of recently developed methods for rapid data acquisition (crowdsourcing), obtaining reliable manual annotations for highly specialized domains is still difficult and acts as a bottleneck in the development of high-quality medical text simplification systems.

Our proposed approach combines automatic mining of AAs and their possible expansions from medical websites, a first-pass simplification step using

Proceedings of the Seventh International Workshop on Health Text Mining and Information Analysis (LOUHI), pages 52–60,
Austin, TX, November 5, 2016. ©2016 Association for Computational Linguistics

statistical machine translation, and a second-pass rescoring step using recently-developed document-level neural language models. To address the data sparsity issue we investigate model training with mismatched training data as well as self-training.

We evaluate our approach on a subset of a publicly available corpus of nursing progress notes from the MIMIC-II database. Results show an F1 score for AA identification of 0.96, an overall expansion accuracy of 74.3%, and a disambiguation accuracy of 71.6%, all without any supervised annotations used during training.

2 Prior Work

AA identification and resolution has a long history in the biomedical domain. Inventories of AAs and their full forms have been compiled by rule-based (Ao and Takagi, 2005) or machine learning techniques (Movshovitz-Attias and Cohen, 2012; Henriksson et al., 2014; Okazaki et al., 2010), often aided by the fact that biomedical texts tend to define AAs at their first mention. Disambiguation of biomedical AAs has been achieved using traditional machine learning approaches, such as vector space methods (Stevenson et al., 2009), naive Bayes classifiers (Bracewell et al., 2005; Stevenson et al., 2009), and SVMs (Joshi et al., 2006; Stevenson et al., 2009). Clustering has also been used for the purpose of disambiguation (Okazaki and Ananiadou, 2006).

Studies on AAs in clinical text are rarer than those for biomedical texts. In (Pakhomov et al., 2005), disambiguation of clinical AAs was achieved using decision trees and maximum entropy models trained on bag-of-word features from hand-annotated and web-collected text. Moon et al. (2012; 2015) similarly investigated several supervised machine learning techniques and text features for disambiguation of AAs in clinical text, including naive Bayes classifiers, SVMs and decision trees trained on bag-of-word features or Unified Medical Language System (UMLS) concepts. They also noted general problems with AA disambiguation in clinical text, such as shortage of training data due to patient privacy constraints, lack of resources developed for clinical text, and non-standard and highly variable language use in clinical notes. Wu et al. (2015) extended SVM classification with vectors based on neural word embeddings. Several systems that participated in the ShaRe/CLEF eHealth Challenge Task on AA normalization (Mowery et al., 2016) utilized conditional random fields (e.g.,(Wu et al., 2013)). Customized expansion dictionaries for clinical text were added in (Xia et al., 2013).

Finally, AA identification and expansion for general English has been addressed by (Ammar et al., 2011; Tevana et al., 2013; Ahmed et al., 2015), among others. The studies most closely related to ours are (Ahmed et al., 2015), which uses language modeling techniques (though not at the document level), and (Ammar et al., 2011), which makes use of statistical machine translation.

resp care note : pt on nrb mask + 6l nc required nt sx due inability to clear secretions. sx copious th yellow sput. pt sats didn't recover after sx + a&a tx.

Figure 1: Sample nursing note.

3 Data and Task

Our test data consists of nursing progress notes from the MIMIC-II database (Saeed et al., 2011), written by nurses in a cardiac intensive care unit. This data set was chosen because (a) it is publicly available[1]; (b) the documents contain a very high percentage of AAs, thus presenting the problem in a condensed form ; (c) it presents interesting additional challenges: it is characteristic of a highly specialized medical sub-domain, and it contains frequent misspellings, non-standard use of AAs, and elliptical syntax, which we plan to address in future work. The present study is intended to serve as the first step in a more comprehensive simplification system for challenging clinical texts. A sample from a nursing note is shown in Figure 1. AAs are not marked as such – the original documents are either all lowercased, all uppercased, or mixed-case with inconsistent casing; acronyms are not marked by periods. Thus, AAs often overlap in form with regular words, in particular function words – e.g., *is* can be the function word "is" or an abbreviation for *incentive spirometry*.

[1] https://mimic.physionet.org/

# words	% AAs	% ambig.
dev set		
125.6 (± 104.3)	25.4 (± 20.4)	73.8 (± 13.6)
eval set		
123.7 (± 112.4)	24.8 (± 9.9)	75.1 (± 12.2)

Table 1: Average number (and stddev) of words, percentage of AAs, and percentage of ambiguous AAs per document, for development (dev) and evaluation (eval) sets.

We use a total of 30 documents (written by various nurses) as reference material. These were split into 15 development and 15 evaluation documents and were manually expanded by medically trained annotators (two medical specialists, one of whom was a hospitalist, and two RNs as additional consultants). The annotation was a consensus annotation; thus, inter-annotator agreement was not measured. The total number of unique AAs in this set is 229, with 611 different instances. Table 1 shows the averages and standard deviations of the number of words, percentage of AAs, and percentage of ambiguous AAs per document. We see fairly large variation in the length of documents and percentages of AAs. On average, however, roughly a quarter of all words are AAs, and 75% of these are ambiguous. We use two other clinical data sets as additional training data: a set of 696 hospital discharge summaries from the i2b2 challenge task (Uzuner et al., 2007) (henceforth "i2b2") and a corpus of 2,365 clinical notes (doctor's notes, hospital discharge summaries, autopsy reports, etc.) from the iDASH repository[2] (henceforth "Cases").

4 Unsupervised Resolution of Abbreviation and Acronyms

Our proposed approach resolves AAs in a largely unsupervised way, requiring true AA sense labels only for system tuning and evaluation but not for training. The first step towards this goal is the acquisition of possible mappings of AAs to their expanded forms. The second step involves preprocessing the nursing documents and generating multiple expanded versions by considering possible combinations of expansions at the sentence level. In a third step, hypotheses are rescored by a document-level language model in order to achieve better disambiguation and selection of expansions.

[2] http://dx.doi.org/10.15147/J2H59S

# mappings	9,852
# unique AAs	4,608
# ambiguous AAs	2,817

Table 2: Number of term mappings (total, unique, and ambiguous) extracted from medical terminology websites.

4.1 Collecting Term Mappings

The first step towards AA resolution is the collection of a glossary that maps AAs to their expanded forms. We found that existing clinical sense inventories did not provide good coverage for the more specialized domain of ICU nursing – e.g., the clinical sense inventory of (Moon et al., 2012) only covered 7% of the AAs in our development and test data; even the much larger ADAM database of MEDLINE abbreviations (Zhou et al., 2006) covered only 65%. Therefore, we are interested in exploring the feasibility of extracting term mappings automatically from generally accessible resources, without additional human curation. Lists of medical and nursing abbreviations were collected from more than a dozen websites, such as Wikipedia's List of Medical Abbreviations, NIH Medline Plus, ECommunity Health Network, etc., by extracting AAs from html and pdf documents using semi-automated scripts. Note that in order to ensure wide coverage, websites were not restricted to those with nursing terminology; neither was the search biased to maximize coverage of the AAs in our corpus. Rather, we aimed at including a wide range of medical AAs to ensure future reusability for other tasks and domains. A total of 10k mappings were collected; after cleaning and removing duplicates the total number was 9,852. These include medical acronyms and abbreviations, but also health insurance terms, proper names, drug names, etc. The resulting mappings were not hand-curated, annotated, or selected for relevance, in order to minimize the amount of human labor involved. The resulting number of unique AAs is 4,608. 2,817 AAs (61.1%) of these have more than one possible expansion. The maximum number of different expansions is 10; the average is 2.6. As an example, the abbreviation *pt* has the following long forms: *patient*, *physical therapy*, *physical therapist*, *patient teaching*, *pint*, *prothrombin time*,

protime. Note that we accepted all possible expansions gathered from the websites as valid; we also did not attempt to cluster potential minor variants (like *protime* and *prothrombin time*) into single entries. Although such cleaning steps might improve results, our goal was to evaluate the performance of our approach with potentially noisy data. The final list of term mappings was found to cover 89% of the AAs in our development and test data.

4.2 Term Expansion

The documents are preprocessed by tokenization of punctuation and mapping all numbers to a generic symbol. To create initial expanded versions of our nursing documents with different possible term expansions we utilize a phrase-based statistical machine translation (SMT) system. An SMT system generates target-language translations from source-language input by finding the maximum-likelihood sentence hypothesis obtained by concatenating individual phrase-level translations. The final score for each hypothesis is provided by a log-linear model that computes a weighted sum of feature functions defined on the input s, the output t, or both:

$$score(s,t) = \frac{1}{Z} exp(\sum_k \lambda_k f_k(s,t)) \qquad (1)$$

where $f(s,t)$ is a feature function, λ is a weight, and Z is a normalization factor. At a minimum, translation scores and a target-side language model score are included; additional feature functions providing e.g., reordering scores or global coherence scores can be added.

Our system maps 'source' (abbreviated) terms to 'target' (expanded) terms according to a phrase table with all pairs of AAs and their expanded forms, trained from the list of term mapping collected in the first step. No entries are included for AAs that are identical to function words such as *is*, *of*, *on*, etc., as initial development experiments showed that these would lead to an overly high number of false alarms. The drawback is that these AAs will never be expanded and will necessarily count as misses.

The language model in the SMT system is a back-off n-gram model trained using modified Kneser-Ney smoothing. The n-gram order was varied between 3 and 5 and optimized on the development set. We compared several language models: one trained on the target side of our term mapping list plus i2b2 data, another on the target side plus Cases data, and a third trained on all three.

The maximum phrase length in our translation system is 5. During decoding, no reordering is permitted. The decoding pass generates up to 100 hypotheses per sentence, in order to explore all possible combinations of AA expansions in a sentence.

5 Self-training

Self-training is a general way of utilizing unsupervised data in a classification system. Starting with a system trained on limited data, the system is applied to unlabeled data. The system's predictions are then filtered according to the probability or confidence of the prediction, and the most likely or confident hypotheses are added back to the training data. This procedure can be iterated. Self-training has been used in NLP for e.g., parsing (McClosky et al., 2006) and machine translation (Ueffing et al., 2007). In the context of AA resolution, (Pakhomov, 2002) has used a similar approach to enrich the training data for a maximum entropy classifier.

Here, we use the top-1 hypotheses of our first-pass SMT system to generate additional training data for both the first and second pass language models. To this end we apply the SMT system to the i2b2 and Cases data. Additionally we utilize up to 2000 nursing notes from the MIMIC-II corpus that do not overlap with our development or evaluation sets. One-best hypotheses are generated from our initial SMT system, and are combined with the target side of the term mapping list. This set is then used to retrain the back-off n-gram model used in the SMT system, and to re-generate the first-pass n-best lists. The automatically expanded data is also used to train the document-level language models described in the following section.

6 Document-Level Context Modeling

The selection of appropriate AA expansions is primarily dependent on the the specific medical domain (nursing, cardiology). AA disambiguation could be aided by a detailed sense inventory with domain labels – however, such a classification was not available from our web sources, and considerable manual

labor would be required for manual annotation.

As an alternative information source it might be advantageous to take into account not only the local sentence context but also the more global document context. For example, the probability of expanding *hr* to *heart rate* rather than *hour* might be boosted by the occurrence of words such as *cardiovascular* or *blood pressure* earlier in the document. Thus, the document context might serve as a proxy for explicit domain or topic models.

To this end we explore document-context language models (DCLMs) as developed by and described in (Ji et al., 2015). DCLMs are neural language models that attempt to predict words based not only on the local n-gram context as in standard back-off language models, but based on the entire history up to the beginning of the document. Various DCLM architectures have been proposed. We provide a concise summary here; details can be found in (Ji et al., 2015).

General recurrent neural language models (RNNLMs) compute the probability of an output vector (probabilities over the output vocabulary) y at time step n as

$$y_n = softmax(W_h h_n + b) \quad (2)$$

$$h_n = g(h_{n-1}, x_n) \quad (3)$$

where W is a weight matrix, b is a bias term, $h \in \mathbb{R}^H$ is a hidden state vector, $x \in \mathbb{R}^K$ is a continuous embedding vector representing the word, and g is a nonlinear activation function. The number of parameters in the network is determined by the dimensionalities of the embedding vector, K, and that of the the hidden vector, H. In "context-to-hidden" DCLMs the hidden state vector in sentence t at time step n is computed not only from the current embedding vector x_n and the preceding state vector $h_{t,n-1}$ but additionally from the last hidden state vector (context vector) of the preceding sentence, $c_{t-1} = h_{t-1,M}$, where M is the last word in the previous sentence. The context vector is simply concatenated with the current embedding vector:

$$h_{t,n} = g(h_{n-1}, x_n \circ c_t) \quad (4)$$

Alternatively, the context vector can be directly combined with the output vector ("context-to-output" model), using its own weight matrix:

$$y_{t,n} = softmax(W_h h_{t,n} + W_c c_{t-1} + b) \quad (5)$$

Due to the addition of a second weight matrix W_c this model has more parameters and may be more difficult to train on limited data. Finally, an "attention-based" architecture has been proposed to address the limits of a fixed-dimensional representation of variably-sized document contexts by formulating the context vector as a linear combination of all hidden states in the previous sentence:

$$c_{t-1,n} = \sum_{m=1}^{M} \alpha_{m,n} h_{t-1}, m \quad (6)$$

Thus, the model can attend to different words in the previous sentence selectively. Moreover, a different context vector is computed for every word n in the current sentence. The context vector is added to both the hidden and the output representation for the current sentence. While this creates a more flexible model, the number of parameters also increases greatly.

Different DCLMs, as well as standard RNNLMs, and RNNLMs whose context can extend beyond the previous sentence boundary, were implemented[3] and were trained using AdaGrad optimization on the same data sets as the back-off ngram models used in the SMT system. 90% of the data was used for training while 10% were held out as development data. Training was stopped when the difference in development set perplexity between the previous and the current iteration was at most 0.5. Different values were investigated for the number units in the embedding and hidden layers (K and H).

For second-pass rescoring of n-best lists with DCLMs we proceed as follows. For each hypothesis in the n-best list for the current sentence, a new "document" is created by concatenating the hypothesis with the previous document context. Each of these documents is scored with the DCLM. The hypothesis resulting in the lowest per-document perplexity chosen and committed to the growing document context. Since no prior context is available for the first sentence in each document, and all further choices are dependent on the choices for previous

[3]Using https://github.com/jiyfeng/dclm

sentences, we choose the 1-best hypothesis from the first pass SMT system for the first sentence, rather than assuming a "dummy" context. The vocabulary of the models is restricted to those words that occur at least 3 times in the training data; all others are mapped to a generic "unknown word" symbol.

We noticed during training that the attention-based DCLM obtained much higher perplexity on the development data than the other models, most likely as a result of having too little training data in relation to the number of parameters. This model was therefore excluded from further experiments.

7 Experiments and Results

The first evaluation criterion for our method is the correct identification of AAs vs. regular words. Contrary to rule-based or supervised approaches to AA identification (Nadeau and Turney, 2005; Dannélls, 2006; Moon et al., 2015) AAs are not identified explicitly but implicitly through the choices made by the SMT system. AA identification can be considered a binary detection problem and can thus be evaluated by precision, recall, and F1 score. The second evaluation measure is *overall accuracy*, i.e., the overall percentage of correct AA expansions. Finally, we measure the *disambiguation accuracy*, i.e., the percentage of correct expansions of ambiguous AAs only.

Table 3 shows precision (P), recall (R), F1-score (F1), overall accuracy (A) and disambiguation accuracy (DA) on the eval set for several baseline systems. *Random* is a baseline system where one of the sentence hypotheses produced by the SMT system is selected randomly.[4] Precision and recall are high (and generally stable across all different models), since it is only a small number of words not caught by the function word filter that are consistently misinterpreted as AAs. *Oracle* refers to results obtained by a system that always chooses the hypothesis yielding the highest disambiguation accuracy according to the reference annotation – this represents the upper bound on the accuracy that can be achieved given our automatically collected term mappings. The gap between the oracle accuracy and 100% accuracy is due to missing expansions in our term mapping list. Row 3 in Table 3 is the result obtained by the first-pass SMT system. The LM for this system was optimized on the development set and consists of a 4-gram back-off model trained using modified Kneser-Ney smoothing on the combined Cases and i2b2 data and the target side of our term mapping list. Accuracy scores obtained by the SMT model are markedly higher than random scores, though there is still much room for improvement.

[4] A majority sense baseline system is not available due to the lack of a sense inventory with frequency information for this data set.

	System	P	R	F1	A	DA
1	Random	0.95	0.97	0.96	56.6	48.2
2	Oracle	0.93	0.95	0.94	80.0	78.6
3	SMT	0.95	0.97	0.96	72.0	68.0

Table 3: Precision (P), recall (R), F1-score (F1), overall accuracy (A) and disambiguation accuracy (DA) for random baseline, oracle topline, and 1-best output from initial SMT system.

	System	P	R	F1	A	DA
+ self-training						
1	Random	0.95	0.97	0.96	60.4	52.4
2	Oracle	0.96	0.97	0.96	80.9	80.7
3	SMT	0.95	0.97	0.96	72.2	69.4
+ DCLM						
4	DCLM	0.95	0.97	0.96	74.3	71.6

Table 4: Precision (P), recall (R), F1-score (F), overall expansion accuracy (A) and disambiguation accuracy (DA) after self-training and second-pass rescoring with DCLMs.

Table 4 shows the results obtained by an improved system that utilizes self-training and DCLMs. For self-training, the amount of automatically expanded MIMIC-II data and the combination with Cases and i2b2 data was optimized on the development set. Combining the latter two sets with 1,500 expanded documents from MIMIC to train a 4-gram back-off LM was found to be best. Since new n-best lists are generated using the self-trained models, the Random and Oracle results are different (and improved). The accuracy of our SMT system's output is also improved by 1.4% absolute.

For rescoring hypotheses with document-level language models we investigated the DCLM architectures described in Section 6, minus the attention-based model, well as standard RNNLMs and RNNLMs whose context can extend in the past beyond the sentence boundary. The number of parame-

ters for each model (K and H) was optimized on the development set. Different models trained on different automatically expanded data sources (Cases, i2b2, and MIMIC-II) and their combinations were investigated. It was found that the combined data as well as the Cases and i2b2 data sets in isolation actually resulted in a *worse* performance of the rescored system compared to the first-pass SMT system. While our mismatched data sources did help in training the SMT system, DCLMs, which attempt to model the entire document structure, seem to be very sensitive to mismatched data. By contrast, DCLMs trained on the automatically expanded MIMIC-II data only did achieve an improvement over the first-pass system. The best model (obtained by development set optimization) was a "context-to-hidden" DCLM with a hidden layer size of 48 and a word embedding layer size of 128. The best final overall accuracy on the evaluation set is 74.3%; the disambiguation accuracy is 71.6%. This is fairly close to the topline disambiguation accuracy of 80.2% that can be achieved given our term inventory; however, there is further room for improvement. Of the different document context models tested, all performed in a similar range – e.g., the best models with other architectures ("context-to-output" and RNNLMs without sentence boundary) achieved between 70.2% and 71.1% disambiguation accuracy on the eval set. Furthermore, an RNNLM model with only the current sentence as context achieves 70.5%. Thus, while DCLMs seem to provide slight improvements, our text sample is currently too small to assess statistically significant differences between different architectures or context lengths. Rather, the benefit seems to derive from the neural probability estimation technique used in RNNLM-style models.

Figure 2 shows the automatically expanded version of the sample in Figure 1. While most expansions were acceptable, our term mapping list did not contain a domain-appropriate entry for *a&a*, which was therefore expanded incorrectly to *arthroscopy and arthrotomy* rather than *albuterol and atroven*.

8 Discussion

In this paper we have explored unsupervised and self-supervised resolution of AAs in nursing notes. Contrary to most previous work, which has utilized supervised classifiers, AA resolution was achieved using web mining to extract term mappings, statistical machine translation, and document-level neural language modeling. With the exception of a small set of hand-annotated documents used to evaluate different models, no ground truth labels were required. Results demonstrated positive effects from self-training and neural language models. Future work will include leveraging additional sources for term mappings, the development of statistical models to improve syntactic readability, and readability experiments with lay human readers.

respiratory care note: patient on non-rebreather mask and 6l nasal cannula required nasotracheal suction due inability to clear secretions.
suction copious thick yellow sputum .
patient oxygen/blood saturation level didn't recover after suction and arthroscopy and arthrotomy therapy.

Figure 2: Expanded version of nursing note.

Acknowledgments

The 'Cases' dataset used in this project was downloaded from iDASH repository (https://idash-data.ucsd.edu) supported by the National Institutes of Health through the NIH Roadmap for Medical Research, Grant U54HL108460. Deidentified clinical records used in this research were provided by the i2b2 National Center for Biomedical Computing funded by U54LM008748 and were originally prepared for the Shared Tasks for Challenges in NLP for Clinical Data organized by Dr. Ozlem Uzuner, i2b2 and SUNY. This work was supported in part by the National Library of Medicine (NLM) under award number R01 10432704 and by the UW Provosts Office through a grant to the first author.

References

A.G. Ahmed, F.F.A. Hady, E. Nabil, and A. Badr. 2015. A language modeling approach for acronym expansion disambiguation. In *Computational Linguistics and Intelligent Text Processing. Lecture Notes in Computer Science vol. 9041*, pages 264–278.

W. Ammar, K. Darwish, A. El Kahki, and K. Hafez. 2011. ICE-TEA: in-context expansion and translation of English abbreviations. In *Proceedings of CICLing*, pages 41–54.

H. Ao and T. Takagi. 2005. ALICE: an algorithm to extract abbreviations from MEDLINE. *JAMIA 12(5)*, pages 576–586.

D.B. Bracewell, S. Russell, and A.S. Wu. 2005. Identification, expansion and disambiguation of acronyms in biomedical text. In *Proceedings of ISPA Workshops*, pages 186–195.

D. Dannélls. 2006. Automatic acronym recognition. In *Proceedings of EACL*, pages 167–170.

T. Delbanco, J. Walker, S.K. Bell, J.D. Darer, J.G. Elmore, N. Faraq, et al. 2012. Inviting patients to read their doctors notes: Quasi-experimental study and a look ahead. *Annals of Internal Medicine*, 15(7):461.

T. Esch, R. Mejilla, M. Anselmo, B. Podtschaske, T. Delbanco, and J. Walker. 2016. Engaging patients through open notes: an evaluation using mixed methods. *BMJ Open*, 6:e010034.

A. Henriksson, H. Moen, M. Skeppstedt, V. Daudaravicius, and M. Duneld. 2014. Synonym extraction and abbreviation expansion with ensembles of semantic spaces. *Journal of Biomedical Semantics 5(6)*.

Y. Ji, T. Cohn, L. Kong, C. Dyer, and J. Eisenstein. 2015. Document context language models. *CoRR*, abs/1511.03962.

M. Joshi, T. Pedersen, R.Maclin, and S. Pakhomonov. 2006. Kernel methods for word sense disambiguation and acronym expansion. In *Proceedings of AAAI*.

D. McClosky, E. Charniak, and M. Johnson. 2006. Effective self-training for parsing. In *Proceedings of HLT-NAACL*, pages 152–159.

S. Moon, S. Pakhomov, and G. Melton. 2012. Automated disambiguation of acronyms and abbreviations in clinical texts: Window and training size considerations. In *Proceedings of AMIA*, pages 1310–1319.

S. Moon, B. McInnes, and G.B. Melton. 2015. Challenges and practical approaches with word sense disambiguation of acronyms and abbreviations in the clinical domain. *Healthc Inform Res*, 21(1):35–42.

D. Movshovitz-Attias and W. Cohen. 2012. Alignment-HMM-based extraction of abbreviations from biomedical text. In *Proceedings of the BioNLP*, pages 47–55.

D.L. Mowery, B.R. South, L. Christensen, and J. Leng et al. 2016. Normalizing acronyms and abbreviations to aid patient understanding of clinical texts: ShaRe/CLEF ehealth challenge 2013, task 2. *J Biomed Semantics*, 7(43).

D. Nadeau and P. Turney. 2005. A supervised learning approach to acronym identification. In *Canadian Society Conference on Advances in Artificial Intelligence*, pages 319–329.

N. Okazaki and S. Ananiadou. 2006. Clustering acronyms in biomedical text for disambiguation. In *Proceedings of LREC*, pages 959–962.

N. Okazaki, S. Ananiadou, and J. Tsuji. 2010. Building a high-quality sense inventory for improved abbreviation disambiguation. *Bioinformatics*, 26(9):1246–1253.

S. Pakhomov, T. Pedersen, and C.G. Chute. 2005. Abbreviation and acronym disambiguation in clinical discourse. pages 589–593.

S. Pakhomov. 2002. Semi-supervised maximum entropy based approach to acronym and abbreviation normalization in medical texts. In *Proceedings of ACL*, pages 160–167.

M. Saeed, M. Villaroel, A.T. Reisner, and G. Clifford et al. 2011. Multiparameter intelligent monitoring in intensive care II (MIMIC-II): A public access ICU database. *Critical Care Medicine*, 39(5).

M. Stevenson, Y. Guo, A. Al Amri, and R. Gaizauskas. 2009. Disambiguation of biomedical abbreviations. In *Proceedings of the Workshop on BioNLP*, pages 71–79.

B. Tevana, T. Cheng, K. Chakrabarti, and Y. He. 2013. Mining acronym expansions and their meanings using query click log. In *Proceedings of WWW*, pages 1261–1271.

N. Ueffing, G. Haffari, and A. Sarkar. 2007. Transductive learning for statistical machine translation. In *Proceedings of ACL*.

O. Uzuner, U. Juo, and P. Szolovits. 2007. Evaluating the state-of-the-art in automatic de-identification. *JAMIA 14(5)*, pages 550–63.

J. Walker, M. Meltsner, and T. Delbanco. 2015. US experience with doctors and patients sharing clinical notes. *BMJ*, page 350:g7785.

J.L. Wolff, J.D. Darer, A. Berger, and D. Clarke et al. 2016. Inviting patients and care partners to read doctors' notes: OpenNotes and shared access to electronic medical records. *JAMIA*, to appear.

Y. Wu, B. Tang, M. Jiang, S. Moon, J.C. Denny, and H. Xu. 2013. Clinical acronym/abbreviation normalization using a hybrid approach. In *Proceedings of CLEF*.

Y. Wu, J. Xu, Y. Zhang, and H. Xu. 2015. Clinical abbreviation disambiguation using neural word embeddings. In *Proceedings of the 2015 Workshop on Biomedical Natural Language Processing (BioNLP)*, pages 171–176.

Y. Xia, X. Zhong, P. Liu, C. Tan, S. Na, Q. Hu, et al. 2013. Normalization of abbreviations/acronyms: THCIB at CLEF eHealth 2013 Task 2. In *CLEF 2013 Evaluation Labs and Workshops: Working Notes*.

W. Zhou, V.I.Torvik, and N.R. Smalheiser. 2006. ADAM: another database of abbreviations in MEDLINE. *Bioinformatics*, 22(22)::2813–2818.