# Unsupervised Domain Adaptation for Clinical Negation Detection

**Timothy A. Miller**[1], **Steven Bethard**[2], **Hadi Amiri**[1], **Guergana Savova**[1]
[1] Boston Children's Hospital Informatics Program, Harvard Medical School
{firstname.lastname}@childrens.harvard.edu
[2] School of Information, University of Arizona
bethard@email.arizona.edu

## Abstract

Detecting negated concepts in clinical texts is an important part of NLP information extraction systems. However, generalizability of negation systems is lacking, as cross-domain experiments suffer dramatic performance losses. We examine the performance of multiple unsupervised domain adaptation algorithms on clinical negation detection, finding only modest gains that fall well short of in-domain performance.

## 1 Introduction

Natural language processing applied to health-related texts, including clinical reports, can be valuable for extracting information that does not exist in any other form. One important NLP task for clinical texts is concept extraction and normalization, where text spans representing medical concepts are found (e.g., *colon cancer*) and mapped to controlled vocabularies such as the Unified Medical Language System (UMLS) (Bodenreider and McCray, 2003). However, clinical texts often refer to concepts that are explicitly not present in the patient, for example, to document the process of ruling out a diagnosis. These *negated* concepts, if not correctly recognized and extracted, can cause problems in downstream use cases. For example, in phenotyping, a concept for a disease (e.g., *asthma*) is a strong feature for a classifier finding patients with asthma. But if the text *ruled out asthma* occurs and the negation is not detected, this text will give the exact opposite signal that its inclusion intended.

There exist many systems for negation detection in the clinical domain (Chapman et al., 2001, 2007; Harkema et al., 2009; Sohn et al., 2012; Wu et al., 2014; Mehrabi et al., 2015), and there are also a variety of datasets available (Uzuner et al., 2011; Albright et al., 2013). However generalizability of negation systems is still lacking, as cross-domain experiments suffer dramatic performance losses, even while obtaining F1 scores over 90% in the domain of the training data (Wu et al., 2014).

Prior work has shown that there is a problem of generalizability in negation detection, but has done little to address it. In this work, we describe preliminary experiments to assess the difficulty of the problem, and evaluate the efficacy of existing domain adaptation algorithms on the problem. We implement three unsupervised domain adaptation methods from the machine learning literature, and find that multiple methods obtain similarly modest performance gains, falling well short of in-domain performance. Our research has broader implications, as the general problem of generalizabiliy applies to all clinical NLP problems. Research in unsupervised domain adaptation can have a huge impact on the adoption of machine learning-based NLP methods for clinical applications.

## 2 Background

Domain adaptation is the task of using labeled data from one domain (the *source* domain) to train a classifier that will be applied to a new domain (the *target* domain). When there is some labeled data available in the target domain, this is referred to as *supervised domain adaptation*, and when there is no labeled data in the target domain, the task is called *unsupervised domain adaptation* (UDA). As the unsupervised version of the problem more closely aligns to real-world clinical use cases, we focus on that setting.

One common UDA method in natural language processing is *structural correspondence learning* (SCL; Blitzer et al. (2006)). SCL hypothesizes that some features act consistently across domains (so-called *pivot features*) while others are still informative but are domain-dependent. The SCL method



*Proceedings of the BioNLP 2017 workshop*, pages 165–170,
Vancouver, Canada, August 4, 2017. 

combines source and target extracted feature sets, and trains classifiers to predict the value of pivot *features*, uses singular value decomposition to reduce the dimensionality of the pivot feature space, and uses this reduced dimensionality space as an additional set of features. This method has been successful for part of speech tagging (Blitzer et al., 2006), sentiment analysis (Blitzer et al., 2007), and authorship attribution (Sapkota et al., 2015), among others, but to our knowledge has not been applied to negation detection (or any other biomedical NLP tasks). One difficulty of SCL is in selecting the pivot features, for which most existing approaches use heuristics about what features are likely to be domain independent.

Another approach to UDA, known as *bootstrapping* or *self-training*, uses a classifier trained in the source domain to label target instances, and adds confidently predicted target instances to the training data with the predicted label. This method has been successfully applied to POS tagging, spam email classification, named entity classification, and syntactic parsing (Jiang and Zhai, 2007; McClosky et al., 2006).

Clinical negation detection has a long history because of its importance to clinical information extraction. Rule-based systems such as Negex (Chapman et al., 2001) and its successor, ConText (Harkema et al., 2009) contain manually curated lists of negation cue words and apply rules about their scopes based on word distance and intervening cues. While these methods do not learn, the word distance parameter can be tuned by experts to apply to their own datasets. The DepNeg system (Sohn et al., 2012) used manually curated dependency path features in a rule-based system to abstract away from surface features. The Deepen algorithm (Mehrabi et al., 2015) algorithm also uses dependency parses in a rule-based system, but applies the rules as a post-process to Negex, and only to the concepts marked as negated.

Machine learning approaches typically use supervised classifiers such as logistic regression or support vector machines to label individual concepts based on features extracted from surrounding context. These features may include manually curated lists, such as those from Negex and ConText, as well as features intended to emulate the rules of those systems, as well as more exhaustive contextual features common to NLP classification problems. The 2010 i2b2/VA Challenge (Uzuner et al., 2011) had an "assertion classification" task, where concepts had mutually exclusive *present, absent (negated), possible, conditional, hypothetical, and non-patient* attributes, and this task had a variety of approaches submitted that used some kind of machine learning. The top-performing system (de Bruijn et al., 2011) used a multi-level ensemble classifier, classifying assertion status of each word with three different machine learning systems, then feeding those outputs into a concept-level multi-class support vector machine classifier for the final prediction. In addition to standard bag of words features for representing context, this system used Brown clusters to abstract away from surface feature representations. The MITRE system (Clark et al., 2011) used conditional random fields to tag cues and their scopes, then incorporated cue information, section features, semantic and syntactic class features, and lexical surface features into a maximum entropy classifier. Finally, Wu et al. (2014) incorporated many of the dependency features from rule-based DepNeg system (Sohn et al., 2012) and the best features from the i2b2 Challenge into a machine learning system.

## 3 Methods

In this work, we apply unsupervised domain adaptation algorithms to machine learning systems for clinical negation detection, evaluating the extent to which performance can be improved when systems are trained on one domain and applied to a new domain. We make use of the (Wu et al., 2014) system in these experiments, as it is freely available as part of the Apache cTAKES (Savova et al., 2010)[1] clinical NLP software, and can be easily retrained.

Unsupervised domain adaptation (UDA) takes place in the setting where there is a *source* dataset $D_s = \{\mathbf{X}, \vec{y}\}$, and a *target* dataset $D_t = \{\mathbf{X}\}$, where feature representations $\mathbf{X} \in \mathbb{R}^{N \times D}$ for $N$ instances and $D$ feature dimensions and labels $\vec{y} \in \mathbb{R}^N$. Our goal is to build classifiers that will perform well on instances from $D_s$ as well as $D_t$, despite having no gold labels from $D_t$ to use at training time. Here we describe a variety of approaches that we have implemented.

The baseline cTAKES system that we use is a support vector machine-based system with L1 and L2 regularization. Regularization is a penalty term added to the classifier's cost function during training that penalizes "more complex" hypotheses, and

[1] http://ctakes.apache.org

is intended to reduce overfitting to the training data. L2 regularization adds the L2 norm to the classifier cost function as a penalty and tends to favor smaller feature weights. L1 regularization adds the L1 norm as a penalty and favors sparse feature weights (i.e., setting many weights to zero).

Before attempting any explicit UDA methods, we evaluate the simple method of increasing regularization. While regularization is already intended to reduce overfitting, it may still overfit on a target domain since its hyper-parameter is tuned on the source domain. In a real unsupervised domain adaptation scenario it is not possible to tune this parameter on the target domain, so for this work we use heuristic methods to set the adapted regularization parameter. We first find the optimal regularization hyperparameter $C$ using cross-validation on the source data, then increase it by an order of magnitude and retrain before testing on target data. For example, if we find that the best F1 score occurs when $C = 1$ for a 5-fold cross-validation experiment on the source data, we retrain the classifier at $C = 0.1$ before applying to target test data.[2] Changing this parameter by one order of magnitude is purely a heuristic approach, chosen because that is how we (the authors) typically would vary this parameter during tuning. Future work may explore whether this parameter on target data without supervision, perhaps by using some information about the data distribution in the target domain.

The first UDA algorithm we implement is structural correspondence learning (SCL) (Blitzer et al., 2006). Following Blitzer et al. we select as pivot features those features that occur more than 50 times in both the source and target data. Then, for each data instance $i$ in $\mathbf{X}_c = \{\mathbf{X}_s \cup \mathbf{X}_t\}$, and each pivot feature $p$, we extract the non-pivot features of $i$ (non-pivot features are simply all features not selected as pivot features), $\vec{x_i} = \mathbf{X}_c[i, \textit{non-pivots}]$, and a classification target, $y_i[p] = [\![\mathbf{X}_c[i, p] > 0.5]\!]$.[3] For each pivot feature $p$, we train a linear classifier on the $(\vec{x_i}, y_i[p])$ classification instances, take the resulting feature weights, $w_p$, and concatenate them into a matrix $W$. We decompose $W$ using singular value decomposition: $W = U\Sigma V^T$, and construct $\theta$ as the first $d$ dimensions of $U$. This matrix $\theta$ represents a projection from non-pivot features to a reduced dimensionality version of the pivot-feature space. At training and test time, features are extracted normally, and non-pivot feature values are multiplied by $\theta$ to create *correspondence features* in the reduced-dimensionality pivot space. Following Sapkota et al. (2016), we experiment with two methods of combining correspondence features with the original features: *All+New*, which combines all the original features with the correspondence features, and *Pivot+New* which combines only the pivot features from the original space with the correspondence features.

[2]Note that since C is the cost of misclassifying training instances, increasing regularization means lowering C.

[3]We use $[\![expr]\!]$ to denote the indicator function, which returns 1 if $expr$ is true and 0 otherwise.

| | Test corpus | | | |
|---|---|---|---|---|
| Train corpus | Seed | Stratified | Mipacq | i2b2 |
| Seed | **0.88** | 0.76 | 0.65 | 0.79 |
| Stratified | 0.66 | **0.83** | 0.67 | 0.79 |
| Mipacq | 0.73 | 0.78 | **0.75** | 0.85 |
| i2b2 | 0.65 | 0.59 | 0.64 | **0.93** |

Table 1: Results (F1 scores) of baseline cross-domain experiments. Bold diagonals indicate in-domain results, which were obtained with 5-fold cross-validation. Off-diagonal elements were trained on source data and tested on target data.

The next UDA algorithm we implement is bootstrapping. Jiang and Zhai (2007) introduced a variety of methods for UDA, under the broad heading of *instance weighting*, but the method they call *bootstrapping* was the only one which does not rely on any target domain labeled data. This method creates pseudo-labels for a portion of the target data by running a classifier trained only on source data on the target data, and adding confidently classified target instances to the training data, labeled with whatever the classifier decided. Jiang and Zhai experiment with the weights of these instances, either giving higher weights to target instances or weighting them the same as source instances. We implemented a simpler version of bootstrapping that does not modify instance weights, and adds instances based on the initial classifier score (rather than iteratively re-training and adding additional instances). We allow up to 1% of the target instances to be added.

In addition to adding the highest-scoring instances, we also experiment with adding only high-scoring instances from the minority class. In many NLP tasks, including negation detection, the label of interest has low prevalence, and there is a danger that the classifier will be most confident on the majority class and only add target instances with that

| Source | Target | None | 10xReg | SCL A+N | SCL P+N | BS-All | BS-Minority | ISF |
|---|---|---|---|---|---|---|---|---|
| Seed (L1) | Strat | 0.76 | **0.8** | **0.8** | 0.69 | 0.79 | 0.79 | **0.8** |
| | Mipacq | 0.65 | 0.66 | 0.69 | 0.6 | 0.69 | **0.7** | 0.69 |
| | i2b2 | 0.79 | **0.83** | **0.83** | 0.71 | **0.83** | **0.83** | **0.83** |
| Strat (L1) | Seed | 0.66 | 0.66 | 0.66 | 0.58 | 0.66 | **0.67** | 0.66 |
| | Mipacq | 0.67 | **0.68** | **0.68** | 0.65 | **0.68** | 0.66 | **0.68** |
| | i2b2 | 0.79 | 0.79 | 0.79 | 0.71 | 0.79 | **0.8** | 0.79 |
| Mipacq (L2) | Seed | **0.73** | 0.59 | **0.73** | 0.71 | **0.73** | 0.71 | **0.73** |
| | Strat | 0.78 | 0.76 | 0.78 | 0.71 | 0.78 | **0.79** | 0.78 |
| | i2b2 | **0.85** | 0.77 | **0.85** | 0.84 | 0.84 | **0.85** | **0.85** |
| i2b2 (L1) | Seed | 0.65 | **0.72** | **0.72** | 0.67 | **0.72** | **0.72** | **0.72** |
| | Strat | 0.59 | 0.68 | **0.69** | 0.62 | 0.68 | 0.68 | 0.68 |
| | Mipacq | 0.64 | **0.69** | **0.69** | 0.68 | **0.69** | **0.69** | **0.69** |
| Average | | 0.71 | 0.72 | **0.74** | 0.68 | **0.74** | **0.74** | **0.74** |

Table 2: Results of unsupervised domain adaptation algorithms (F1 scores). None=No adaptation, 10xReg=Regularization with 10x penalty, SCL A+N is structural correspondence learning with all features in addition to projected (new) features, SCL P+N is SCL with pivot features and projected features, BS-All=Bootstrapping with instances of all classes added to source, BS-Minority=Bootstrapping with only instances of minority class added to source, ISF=Instance similarity features.

label. We therefore experiment with only adding minority class instances, enriching the training data to have a more even class distribution.

The final UDA algorithm we experiment with uses instance similarity features (ISF) (Yu and Jiang, 2015). This method extends the feature space in the source domain with a set of similarity features computed by comparison to features extracted from target domain instances. Formally, the method selects a random subset of $K$ *exemplar instances* from $D_t$ and normalizes them as $\hat{\vec{e}} = \frac{\vec{e}}{||\vec{e}||}$. Similarity feature $k$ for instance $i$ in the source data set is computed as the dot product $\mathbf{X}_t[i] \cdot \hat{\vec{e}}[k]$. Following Yu and Jiang, we set $K = 50$ and concatenate the similarity features to the full set of extracted features for each source instance at training. These exemplar instances must be kept around past training time, so that at test time similarity features can be similarly created for test instances.

## 4 Evaluation

Our evaluation makes use of four corpora of clinical notes with negation annotations – i2b2 (Uzuner et al., 2011), Mipacq (Albright et al., 2013), SHARP (Seed), and SHARP (Stratified). We first perform cross-domain experiments in the no adaptation setting to replicate Wu et al.'s experiments.[4] One difference to Wu et al. is that we evaluate on the training split of the target domain – we made this choice because the development and test sets for some of the corpora are quite small and the training data gives us a more stable estimate of performance. We tune two hyperparameters, L1 vs. L2 regularization and the values of regularization parameter C, with five-fold cross validation on the source corpus. We record results for training on all four corpora, testing on all three target domains, as well as a cross-validation experiment to measure in-domain performance. Table 1 shows these results, which replicate Wu et al. in finding dramatic performance declines across corpora.

In our domain adaptation experiments, we also use all four corpora as source domains, and for each source domain we perform experiments where the other three corpora are target domains. This result is reported in Table 2.

## 5 Discussion and Conclusion

These results show that unsupervised domain adaptation can provide, at best, a small improvement to clinical negation detection systems.

Strong regularization, while not obtaining the highest average performance, provides nominal improvements over no adaptation in all settings except when the source corpus is Mipacq, in which case performance suffers severely. Mipacq has two unique aspects that might be relevant; first, it is the largest training set, and second, it pulls docu-

[4] See that paper for an discussion of corpus differences.

ments from a very diverse set of sources (clinical notes, clinical questions, and medical encyclopedias), while the other corpora only contain clinical notes. Perhaps because the within-corpus variation is already quite high, the regularization parameter that performs best during tuning is already sufficient to prevent overfitting on any target corpus with less variation, and increasing it leads to underfitting and thus poor target domain performance. Future work may explore this hypothesis, which must include some attempt to relate the within- and between-corpus variation.

Four different systems all obtain the highest average performance, with BS-All (standard bootstrapping), BS-Minority (bootstrapping with minority class enrichment), structural correspondence learning (SCL A+N), and instance similarity features (ISF) all showing 3% gain in performance (71% to 74%). While the presence of some improvement is encouraging, the improvements within any given technique are not consistent, so that without labeled data from the target domain it would not be possible to know which UDA technique to use. We set aside the question of "statistical significance," as that is probably too low of a bar – whether or not these results reach that threshold, they are still disappointingly low and likely to cause issues if applied to new data.

In summary, selecting a method is difficult, and many of these methods have hyper-parameters (e.g., pivot selection for SCL, number of bootstrapping instances, number of similarity features) that could potentially be tuned, yet in the unsupervised setting there are no clear metrics to use for tuning performance. Future work will explore the use of unsupervised performance metrics that can serve as proxies to test set performance for optimizing hyperparameters and selecting UDA techniques for a given problem.

## Acknowledgments

This work was supported by National Institutes of Health grants R01GM114355 from the National Institute of General Medical Sciences (NIGMS) and U24CA184407 from the National Cancer Institute (NCI). The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

## References

Daniel Albright, Arrick Lanfranchi, Anwen Fredriksen, Will F Styler, Colin Warner, Jena D Hwang, Jinho D Choi, Dmitriy Dligach, Rodney D Nielsen, James H Martin, Wayne Ward, Martha Palmer, and Guergana K Savova. 2013. Towards comprehensive syntactic and semantic annotations of the clinical narrative. *Journal of the American Medical Informatics Association: JAMIA* 20(5):922–930. https://doi.org/10.1136/amiajnl-2012-001317.

J Blitzer, M Dredze, and F Pereira. 2007. Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In *ACL 2007*. page 440. http://www.cs.brandeis.edu/ marc/misc/proceedings/acl-2007/ACLMain/pdf/ACLMain56.pdf.

J Blitzer, R McDonald, and F Pereira. 2006. Domain adaptation with structural correspondence learning. *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)* http://dl.acm.org/citation.cfm?id=1610094.

Olivier Bodenreider and Alexa T. McCray. 2003. Exploring semantic groups through visual approaches. *Journal of Biomedical Informatics* 36(6):414–432. https://doi.org/10.1016/j.jbi.2003.11.002.

W W Chapman, W Bridewell, P Hanbury, G F Cooper, and B G Buchanan. 2001. A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of biomedical informatics* 34(5):301–310. https://doi.org/10.1006/jbin.2001.1029.

Wendy W. Chapman, David Chu, and John N. Dowling. 2007. ConText : An Algorithm for Identifying Contextual Features from Clinical Text. *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing* (June):81–88.

Cheryl Clark, John Aberdeen, Matt Coarr, David Tresner-Kirsch, Ben Wellner, Alexander Yeh, and Lynette Hirschman. 2011. MITRE system for clinical assertion status classification. *Journal of the American Medical Informatics Association : JAMIA* 18(5):563–567. https://doi.org/10.1136/amiajnl-2011-000164.

Berry de Bruijn, Colin Cherry, Svetlana Kiritchenko, Joel Martin, and Xiaodan Zhu. 2011. Machine-learned solutions for three stages of clinical information extraction: the state of the art at i2b2 2010. *Journal of the American Medical Informatics Association* 18(5):557–562. https://doi.org/10.1136/amiajnl-2011-000150.

Henk Harkema, John N. Dowling, Tyler Thornblade, and Wendy W. Chapman. 2009. ConText: an algorithm for determining negation, experiencer, and temporal status from clinical reports. *Journal of biomedical* 42(5):839–851.

J Jiang and CX Zhai. 2007. Instance weighting for domain adaptation in NLP. *ACL* .

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective Self-Training for Parsing. In *Proceedings of {NAACL HLT} 2006*. New York City, USA, pages 152–159.

Saeed Mehrabi, Anand Krishnan, Sunghwan Sohn, Alexandra M. Roch, Heidi Schmidt, Joe Kesterson, Chris Beesley, Paul Dexter, C. Max Schmidt, Hongfang Liu, and Mathew Palakal. 2015. DEEPEN: A negation detection system for clinical text incorporating dependency relation into NegEx. *Journal of Biomedical Informatics* 54:213–219. https://doi.org/10.1016/j.jbi.2015.02.010.

U Sapkota, T Solorio, M Montes-y Gómez, and S Bethard. 2016. Domain Adaptation for Authorship Attribution: Improved Structural Correspondence Learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. pages 2226–2235. https://www.aclweb.org/anthology/P/P16/P16-1210.pdf.

Upendra Sapkota, Steven Bethard, and Manuel Montes-y g. 2015. Not All Character N -grams Are Created Equal : A Study in Authorship Attribution. In *Proceedings of NAACL 2015*. pages 93–102.

GK Savova, JJ Masanz, and PV Ogren. 2010. Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association* .

Sunghwan Sohn, Stephen Wu, and Christopher G Chute. 2012. Dependency Parser-based Negation Detection in Clinical Narratives. *AMIA Joint Summits on Translational Science proceedings AMIA Summit on Translational Science* 2012:1–8.

Özlem Uzuner, Brett R South, Shuying Shen, and Scott L DuVall. 2011. 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. *Journal of the American* http://jamia.oxfordjournals.org/content/18/5/552.short.

S Wu, T Miller, J Masanz, M Coarr, and S Halgrim. 2014. Negation's not solved: generalizability versus optimizability in clinical natural language processing. *PloS one* .

J Yu and J Jiang. 2015. A Hassle-Free Unsupervised Domain Adaptation Method Using Instance Similarity Features. *ACL (2)* .