MEDINFO 2013
C.U. Lehmann et al. (Eds.)
© 2013 IMIA and IOS Press.

doi:10.3233/978-1-61499-289-9-185

Improving Heart Failure Information Extraction by Domain Adaptation

Youngjun Kim[a,c], Jennifer Garvin[b,c], Julia Heavirland[c], Stéphane M. Meystre[b,c]

[a] *School of Computing, University of Utah, Salt Lake City, Utah, U.S.*
[b] *Department of Biomedical Informatics, University of Utah, Salt Lake City, Utah, U.S.*
[c] *VA Health Care System, Salt Lake City, Utah, U.S.*

Abstract

Adapting an information extraction application to a new domain (e.g., new categories of narrative text) typically requires re-training the application with the new narratives. But could previous training from the original domain alleviate this adaptation?

After having developed an NLP-based application to extract congestive heart failure treatment performance measures from echocardiogram reports (i.e., the source domain), we adapted it to a large variety of clinical documents (i.e., the target domain). We wanted to reuse the machine learning trained models from the source domain, and experimented with several popular domain adaptation approaches such as reusing the predictions from the source model, or applying a linear interpolation. As a result, we measured higher recall and precision (92.4% and 95.3% respectively) than when training with the target domain only.

Keywords:

Heart Failure, Ventricular Ejection Fraction, Medical Informatics, Natural Language Processing.

Introduction

Heart Failure (HF) is one of most common diseases in the U.S. and is subject to many treatment quality improvement efforts. Left ventricular ejection fraction (LVEF) qualitative and quantitative assessments are important indicators to monitor the progression and treatment of congestive heart failure. This study was realized in the context of the ADAHF (Automated Data Acquisition for Heart Failure) project, a U.S. Veterans Administration (VA) project aiming at the automated extraction of congestive heart failure treatment performance measures from clinical notes. These performance measures include LVEF assessments (their mention and measured values), medications (angiotensin-converting-enzyme inhibitors and angiotensin receptor blockers), or reasons not to administer these medications. In the study reported here, we focused on left ventricular ejection fraction (EF) mentions and associated qualitative assessments (e.g., ‘decreased’, ‘preserved’) and quantitative values (e.g., ‘35%’, ‘0.5’).

The extraction of the aforementioned information can rely on various methods that include regular expressions and machine learning. The former was chosen for the CUIMANDREef [1] system. We applied the latter, as a sequence tagging task. Sequence tagging based on online learning has been chosen for many Natural Language Processing (NLP) tasks, such as, protein or gene detection in biomedical literatures [2-4], and medical term extraction from clinical notes [5]. Our implementation based on machine learning allowed for higher accuracy than CUIMANDREef when detecting mentions of LVEF and associated values in echocardiogram reports [6].

When adapting our machine learning-based application to the new domain of the ADAHF project, which is a domain with a large variety of clinical notes (Table 2) instead of only echocardiogram reports, we experimented with several popular domain adaptation approaches. Our goal was to reuse the machine learning trained models from the echocardiogram reports corpus, to improve the efficiency of the adaptation.

Domain adaptation of statistical classifications has received increased attention for various NLP problems, such as text classification, sentence parsing, or machine translation. In our study, the echocardiogram reports corpus was the source domain, and the ADAHF project corpus with various note types was our target domain. Many algorithms for efficient domain adaptation with or without labeled target domain data have been proposed. For mention detection, Florian et al. introduced a method that builds on a source domain model and uses its predictions as features to train the target domain model (*Pred* method explained below) [7]. Chelba and Acero used the feature weights of the source domain model as a Gaussian prior for initializing each feature in the target domain model (*Prior*) [8]. They applied their approach to recover the correct capitalization of uniformly cased text. Foster and Kuhn linearly interpolated source and target domain models for machine translation (*LinInt*) [9]. Daumé presented a feature augmentation method that can learn trade-offs between source/target and general feature weights (*Augment*) [10].

In the following sections, we will describe the domain adaptation approaches mentioned above in more details and present our experimental results.

Materials and Methods

Domains Comparison

The echocardiogram reports corpus (i.e., our source domain) consisted of 765 manually annotated notes. We used 275 notes as source training data (EF Train) and 490 notes for testing (EF Test). For more detailed information about this corpus, see [1].

The ADAHF project uses clinical notes from inpatients with Congestive Heart Failure (CHF) who were discharged from a selection of 8 VA medical centers in 2008. For this study, we sampled 665 clinical notes (i.e., our target domain), and 275 notes were randomly selected to train a sequence classifier

(ADAHF Train) and the remaining 390 notes (ADAHF Test) were used for testing. The ADAHF Train corpus size was chosen to match the EF Train corpus we used. Table 1 shows the number of concepts and notes contained in the source and target domains.

Table 1 – Corpora characteristics

	EF Train	EF Test	ADAHF Train	ADAHF Test
LVEF mentions	723	1,250	510	844
Quantitative values	430	746	455	743
Qualitative values	439	759	44	66
Number of notes	275	490	275	390

The number of qualitative assessments is very different in the two corpora, probably because qualitative assessments occur more frequently in echocardiogram reports than in other clinical note types, and echocardiogram reports only represented about 4% of the note types found in the ADAHF corpus (Table 2), where the most prevalent note types were progress notes, discharge summaries, history and physical notes, and cardiology consultation notes (Table 2).

Table 2 – Clinical note types in the ADAHF corpus

Type of Note	ADAHF Train	ADAHF Test
Progress note	70 (25.5%)	100 (25.6%)
Discharge summary	38 (13.8%)	43 (11.0%)
History and physical	35 (12.7%)	52 (13.3%)
Cardiology consultation	33 (12.0%)	69 (17.7%)
Echocardiogram report	14 (5.1%)	10 (2.6%)
Pharmacy note	12 (4.4%)	5 (1.3%)
Other consultation note	8 (2.9%)	12 (3.1%)
Nursing note	1 (0.4%)	7 (1.8%)
Other note types	64 (23.3%)	92 (23.6%)

Progress notes include several note subtypes, such as intern note, internal medicine note, nurse practitioner note, physician note, etc.

A few differences between the two domains were also related to differences in the annotation schema and instructions, as explained below with LVEF mentions.

In both corpora (i.e., the source and the target domain), the following concepts were annotated;

LVEF mentions

In the EF corpus, LVEF (e.g., “VISUAL ESTIMATE OF LVEF”, “EF”) and left ventricular systolic function (LVSF) (e.g., “Global LV systolic function”, “systolic dysfunction”) were annotated.

However, in the ADAHF corpus, only LVEF was annotated, and LVSF was excluded.

Quantitative values

In both corpora, quantitative values of LVEF (e.g., “~0.60-0.65”, “0.45”, “50%”) were annotated. . There were no quantitative values of LVSF.

Qualitative assessments

Similarly to LVEF mentions, all qualitative assessments of LVEF and LVSF (e.g., “NORMAL”, “mildly decreased”, “SEVERE”.) were annotated, but qualitative assessments of LVSF were excluded in the ADAHF corpus.

The annotation differences related to LVSF made the direct reuse of the model trained with the EF corpus more difficult. To benefit from the EF corpus, as already annotated by medical expert, and make the corpus compatible with the ADAHF corpus, manual adjustments like removing every LVSF mention and its qualitative assessment would be necessary. Instead of this expensive effort, we wanted to resolve this incompatibility with domain adaptation.

As a first step, we reused the machine learning-based information extraction application with the original features designed for the EF corpus and observed if we could measure reasonable performance with the ADAHF corpus. Then, we combined the two corpora in many different ways, and allowed our classifier to benefit from both source and target domain data. In the next section, we will describe how the feature vectors from clinical notes were extracted and which feature set was utilized for concept detection.

Features extraction

As part of our information extraction application built in UIMA [11, 12], pre-processing includes multiple components depicted in Figure 1.

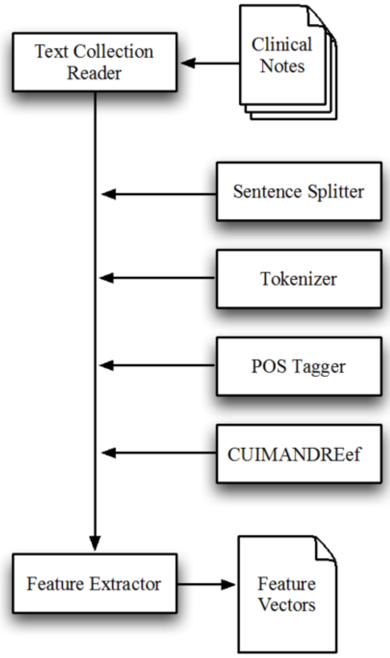

Figure 1- Pre-processing pipeline for features extraction

The sentence splitter detects sentence boundaries and splits the text in sentences; the tokenizer splits sentences in tokens; and the part-of-speech (POS) tagger assigns POS tags to each token. We also used all information extracted by CUIMANDREef [1], the regular expressions-based system.

Features extracted from pre-processors included words, POS tags, morphology, infixes, the output of CUIMANDREef, and their combinations – as listed below:

- Word: 0 (current word), -1 (one previous), +1 (one following), -2, +2, -3, +3, -4, and +4
- Bi-grams of words: [-2, -1], [-1, 0], [0, 1], and [1, 2]
- POS tag: 0 (current word POS tag), -1, +1, -2, +2, -3, +3, -4, +4 and [-2, -1], [-1, 0], [0, 1], and [1, 2]
- Word shape information (for example, "EF" is normalized to "AA", "ejection fraction" to "aa"): 0, -1, +1, -2, +2, -3, +3 and [-2, -1], [-1, 0], [0, 1], and [1, 2]
- Prefix and suffix: up to length of 3
- Morphology: alpha-numeric characters, punctuations, etc.
- CUIMANDREef tag: 0 (the output of CUIMANDREef for the current word), -1, +1, -2, and 2

Concept detection

We reformatted the training data with BIO tags (B: at the beginning, I: inside, or O: outside of a term) and trained a sequential classifier using Miralium [13], a Java implementation of the Margin Infused Relaxed Algorithm (MIRA) [14]. MIRA, also called passive-aggressive algorithm, is an extension of the perceptron algorithm with a margin-dependent learning rate for multiclass classification. It is used to learn the feature weights by processing the training instances one-by-one and seeking the smallest update to the feature weights when the output label of the instance is different than the desired label.

Domain Adaptation

We implemented three baselines and four domain adaptation methods, as explained below:

Baselines

- Source data only (*SrcOnly*): a model trained only with the source data, and then tested with the target data (training with EF Train; testing with ADAHF Test).
- Target data only (*TgtOnly*): a model trained and tested only with the target data (training with ADAHF Train; testing with ADAHF Test).
- Union of data (*Union*): a model trained with the union of the source and target data, and then tested with the target data (training with EF Train and ADAHF Train; testing with ADAHF Test). For simplicity, we did no instance weighting from both datasets. By cross-validation, the weight of instances for each domain can be re-assigned.

Other Methods

- Predictions (*Pred*): a model trained with the target data and the predictions from the *SrcOnly* model, following these steps:
  - Train the *SrcOnly* model with the source data.
  - Run the *SrcOnly* model on the target data.
  - Use the resulting predictions as new feature to augment training of the new model with the target data.
  - Test with the target test data.

  Any combination of predictions by multiple pre-existing models can be used in this method. The identical feature set for both dataset is not necessary while it is required for other approaches.
- Linear interpolation (*LinInt*): the predictions of the *SrcOnly* and the *TgtOnly* models are linearly interpolated, as follows:
  - Train the *SrcOnly* model with the source data.
  - Train the *TgtOnly* model with the target data.
  - Interpolate the probabilistic predictions of both models.

  For each word x with label y in the test data, the probability $Ps(y/x)$ assigned by the *SrcOnly* model and the probability $Pt(y/x)$ assigned by the *TgtOnly* model are combined to obtain the interpolated probability as defined in Equation (1).

  $$P(y|x)=\alpha Ps(y|x)+(1-\alpha)Pt(y|x) \quad (1)$$

  The α value can be between 0 and 1, which can be optimized by testing on a development set or cross validation. For this study, we did not adjust α but assigned it a value of 0.5 (uniform weights) as the simplest setting.
- Prior based (*Prior*): the feature weights learned from the source data are used as a prior to train with the target data. To initialize the weight of each feature in target data, we did as follows:
  - If the feature exists in the *SrcOnly* model, assign the corresponding weight from the *SrcOnly* model.
  - Otherwise, assign a zero weight.
  - Train the target model starting with those feature weights.

  This method makes the feature weight close to the priors from the *SrcOnly* model and keeps the weight when the target data is similar to the source data. In other cases, the weight will be updated from the prior to the new target data.
- Feature augmentation (*Augment*): a model jointly learned from three different versions of original features (common, source-specific, and target-specific). The augmented training instances of the source domain data contain common and source-specific versions while the augmented instances of the target domain contain common and target-specific versions. The transformation of features are as follows:
  - Source data: $x \rightarrow <x, x, 0>$
  - Target data: $x \rightarrow <x, 0, x>$

  0 means zero vectors and the order of segments is common, source-specific, and target-specific. During a single supervised learning process with these augmented features extracted from both data sets, the trade-off regularization between source/target and general weights is possible.

Results

We first present results when training and testing our application with the EF corpus, and then present results of the various approaches for domain adaptation. All trainings were finished after ten iterations and no feature pruning was done for this study.

EF corpus training and testing

As seen in Table 3, our sequential tagger using online learning obtained good performance on every concept type when trained with the EF training corpus and tested with the EF testing corpus. Precision was about 3% higher than recall on average, except with quantitative values. Recall was lower with qualitative assessments, because of the higher variability of these concepts, and difficulty to capture all possible ways to express them in narrative text.

Table 3 – Results of EF Test Data (%)

	Recall	Precision	F_1-measure
LVEF mentions	93.8	97.1	95.4
Quantitative	97.1	96.9	97.0
Qualitative	91.8	96.7	94.2
All concepts	94.1	96.9	95.5

Domain adaptation to ADAHF results

All domain adaptation approaches used the same feature set than with the EF corpus, and were tested with the ADAHF testing corpus. All results are listed in Table 4.

Statistical analysis was realized with multiple Student t-tests and the Šidák correction for multiple hypotheses [15].

As expected, *SrcOnly* performed very poorly, especially with qualitative assessments: precision was only 22.8% with numerous LVSF qualitative assessment false positives. All other approaches obtained significantly better performance (except with qualitative assessments; *p* between 0.00126 and < 0.00001)

As already explained, the *TgtOnly* model was trained and tested with the ADAHF corpus (i.e., the target domain). Even though performance was not as good as when trained and tested with the EF corpus, we confirmed that our application could be utilized with a variety of clinical note types. For qualitative assessments, this method obtained the best precision.

The *Union* approach obtained the best recall with qualitative assessments among all approaches, but also saw precision drop in the same category. It was only significantly better than the *SrcOnly* approach.

The *Pred* approach did not work well with quantitative values and qualitative assessments. However, it obtained the best performance with LVEF mentions detection. For example, with a phrase containing a LVEF mention like “ESTIMATED EF: 15%”, the *Pred* approach was able to extract the correct term “ESTIMATED EF” but the *TgtOnly* approach only captured “EF”. Recall and precision were both significantly better than with the *LinInt* approach (*p* values 0.00366 and 0.00003) and very significantly better than the *SrcOnly* approach (*p* values < 0.00001).

We didn’t observe any significant improvement with the *LinInt* approach. Overall, recall and precision dropped, probably because of the equal weights to predictions from both the *SrcOnly* and *TgtOnly* models that we assigned with this method.

Prior was the only approach that allowed for better overall recall (significantly better than *LinInt* and *SrcOnly*; *p* values 0.00132 and <0.00001), and the best recall with quantitative values. *Prior* and *Augment* seemed the most useful with categories that had the same annotation schemata and guidelines in both domains, like quantitative values.

Augment allowed for results similar to *Prior,* and the highest precision with quantitative values, significantly better than *LinInt*, *Union*, and *SrcOnly* (*p* values 0.00009, 0.00679, and <0.00001). Compared to *TgtOnly*, the *Augment* approach avoided false positives that were wrongly detected by *TgtOnly*: “GFR > 50%” (GFR is the glomerular filtration rate) “fib ~ 40% of time” (fib means atrial fibrillation here).

In summary, the domain adaptation approaches we implemented were efficient to overcome the difference in annotation schemata and allowed for a better performance with categories like LVEF mentions and quantitative values, but they did not outperform the baselines (*TgtOnly* and *Union*) when important difference in prevalence of concepts exist between domains, such as qualitative assessments in our case.

Even if our experimental setting was quite different from traditional domain adaptations with large quantities of source data and small target data, the improvements were promising for the extension of our application to a larger variety of clinical note types.

Conclusion

This study showed that NLP-based information extraction methods could be successfully applied to the detection of men-

Table 4 – Recall (R) and precision (P) results for all domain adaptation approaches [The highest recall and precision for each concept are bolded.]

	SrcOnly		TgtOnly		Union		Pred		LinInt		Prior		Augment	
	R	P	R	P	R	P	R	P	R	P	R	P	R	P
LVEF mentions	81.6	83.5	94.8	95.8	95.5	97.0	**96.5**	**98.0**	94.9	95.0	95.7	96.8	94.9	96.0
Quantitative values	87.2	61.7	91.7	93.3	90.7	91.3	90.0	92.3	90.2	87.0	**92.1**	93.2	91.7	**94.2**
Qualitative values	51.5	22.8	54.6	**97.3**	**57.6**	90.5	53.0	94.6	45.5	58.8	54.6	94.7	54.6	78.3
All concepts	82.9	67.7	91.8	94.7	91.8	94.2	91.8	**95.3**	90.8	90.2	**92.4**	95.1	91.8	94.7

tions of LVEF and associated qualitative assessments and quantitative values. We showed that our application performed well in both corpora with various linguistic features. The domain adaptation results are promising and will allow us to reduce the manual adjustment effort with already annotated data.

Acknowledgments

We thank Hal Daumé III for his valuable advices on domain adaptation and Benoit Favre for his help with *miralium*. Research supported by VA HSR&D IBE 09-069 and by HSR&D HIR 08-374 (Consortium for Healthcare Informatics Research) and HIR 09-007 (Translational Use Case – Ejection Fraction). The views expressed in this article are those of the authors and do not necessarily represent the views of the Department of Veterans Affairs or the University of Utah School of Medicine.

References

[1] Garvin JH, Duvall SL, South BR, Bray BE, Bolton D, Heavirland J, Pickard S, Heidenreich P, Shen S, Weir C, Samore M, Goldstein MK. Automated extraction of ejection fraction for quality measurement using regular expressions in Unstructured Information Management Architecture (UIMA) for heart failure. J Am Med Inform Assoc. 2012: 19: 859-866.

[2] Finkel J, Dingare S, Manning CD, Nissim M, Alex B, and Grover C. Exploring the Boundaries: Gene and Protein Identification in Biomedical Text. BMC Bioinformatics. 2005: 6: S5.

[3] McDonald R and Pereira F. Identifying Gene and Protein Mentions in Text Using Conditional Random Fields. BMC Bioinformatics. 2005: 6: S6.

[4] Zhou G, Shen D, Zhang J, Su J, and Tan S. Recognition of Protein/Gene Names from Text Using an Ensemble of Classifiers. BMC Bioinformatics. 2005: 6: S7.

[5] Bruijn BD, Cherry C, Kiritchenko S, Martin J, and Zhu X. Machine learned Solutions for Three Stages of Clinical Information Extraction: the State of the Art at i2b2 2010. J Am Med Inform Assoc. 2011: 18 (5): 557-562.

[6] Meystre SM, Kim Y, Garvin JH. Comparing Methods for left Ventricular Ejection Fraction Clinical Information Extraction. AMIA Summits Transl Sci Proc, CRI. 2012: 138.

[7] Florian R, Hassan H, Ittycheriah A, Jing H, Kambhatla N, Luo X, Nicolov N, and Roukos S. A Statistical Model for Multilingual Entity Detection and Tracking. Proc. Conf. NAACL and HLT. 2004.

[8] Chelba C and Acero A. Adaptation of Maximum Entropy Capitalizer: Little Data Can Help a Lot. Proc. Conf. EMNLP. 2004.

[9] Foster G and Kuhn R. Mixture-Model Adaptation for SMT. Workshop on Statistical Machine Translation, ACL. 2007.

[10] Daumé H III. Frustratingly Easy Domain Adaptation. Proc. ACL. 2007.

[11] Ferrucci D and Lally A. UIMA: An Architectural Approach to Unstructured Information Processing in the Corporate Research Environment. Journal of Natural Language Engineering. 2004: 10 (3-4): 327-348.

[12] Apache UIMA 2008. Available at http://uima.apache.org.

[13] miralium. http://code.google.com/p/miralium/

[14] Crammer K, and Singer Y. Ultraconservative Online Algorithms for Multiclass Problems. Journal of Machine Learning Research. 2003: 3: 951–991.

[15] Šidák Z. Rectangular confidence regions for the means of multivariate normal distributions. Journal of the American Statistical Association. 1967: 62 (31): 626-633.

Address for correspondence

Youngjun Kim,
School of Computing, University of Utah,
50 S. Central Campus Drive, Salt Lake City,
Utah 84112, USA;
youngjun@cs.utah.edu