

Reverse Active Learning for Optimising Information Extraction Training Production

Dung Nguyen and Jon Patrick

School of IT, University of Sydney
1 Cleveland, Sydney 2006, NSW, Australia
{nguyend,jonpat}@it.usyd.edu.com

Abstract. When processing a noisy corpus such as clinical texts, the corpus usually contains a large number of misspelt words, abbreviations and acronyms while many ambiguous and irregular language usages can also be found in training data needed for supervised learning. These are two frequent kinds of noise that can affect the overall performance of machine learning process. The first noise is usually filtered by the proof reading process. This paper proposes an algorithm to deal with noisy training data problem, for a method we call reverse active learning to improve performance of supervised machine learning on clinical corpora. The effects of reverse active learning are shown to produce results on the i2b2 clinical corpus that are state-of-the-art of supervised learning method and offer a means of improving all processing strategies in clinical language processing.

Keywords: Active Learning, Clinical, Information extraction.

1 Introduction

1.1 Learning from Noisy Gold-Standard

Clinical notes contain valuable information about a patient's health status, however, retrieving information from them is challenging. The popular challenge to supervised machine learning approaches in clinical domain is the level of noise that arises in training data, even if tagging is created by human experts, they may not always be reliable because some instances are implicitly difficult for annotators and them becoming distracted or fatigued over time, introduces variability in the quality of their annotations [1].

The process of removing noisy training data is conventionally known as data selection and data reduction for machine learning [2,3]. The work nearest to our problem with noisy training data is estimating the quality of annotators and then only querying the more reliable annotators in subsequent iterations of active learning (AL) [4]. However, this method only works with an online annotation process and does not analyse the performance of annotators over time. Furthermore, in the case where the gold-standard instances are available but they do not contain any information about the authors, the method of

estimating the performance of annotators is not applicable. Other similar idea come from the Corrective Active Learning which try to clean up the noisy labeled data through human interaction [5]. But using human resources is always costly and this method is only practical when the correctors have intensive knowledge and familiar with the training data. This is not the case when training data usually come from different sources and organizations. Our filtering methods for classification task does not require any human involvement and will work for any source of noisy training data without information about annotators.

1.2 Active Learning

Active learning (AL) is a subfield of machine learning where the learner is allowed to query the most informative instances to retrain the model instead of making a random selection. Based on this approach, with the same number of sample selections, the performance of active learners dominates random learners in most cases. There are three main scenarios for AL: (i) membership query synthesis, (ii) stream-based selective sampling, and (iii) pool-based sampling [1].

All AL scenarios involve evaluating the informativeness of unlabelled instances. There are several strategies to estimate the value of information for each instance. The simplest and most popular strategy is uncertainty sampling which selects the instances that it is least certain to label [6]. Some variations to this query framework used margin and entropy as uncertainty measures [7,8]. Query-by-committee method maintains a list of models trained on the same labelled data but with competing hypotheses and query the instance with lowest agreement [9,10]. The important research question in AL is when to stop querying. Several stopping criteria have been introduced and are based on measures of stability or self-confidence within the learner [11,12].

1.3 The Active Learning's Sampling Bias Problem

A well-known problem with many AL algorithms, especially at the early steps of learning is that they are prone to generate a biased training set rather than be representative of the true underlying data distribution. This is due to the limitations in model initialization and the selection strategy. Many algorithms just randomly select a few instances to train the first model which is usually not a good starting point for the real data distribution. Experiments with various text data sets have shown that an active learner starting from a better initial training set such as clustered-based sampling has better uncertainty estimation than one starting from a randomly generated initial training set [13].

The selection strategies which focus on individual instances (Uncertainty sampling or Query-by-Committee) have a high chance of querying the outliers in the data. To overcome this problem, some algorithms such as Density-Weighted Methods and Balance Exploration and Exploitation are designed based on the main idea that informative instances should not only be those which are uncertain, but also those which are “representative” of the underlying distribution [8]. Expected model change approach has higher priority for instances that will

impact the greatest change to the current model [8]. By focusing on the entire input space rather than individual instances, expected error reduction and variance reduction AL try to minimise the estimated future error and output variance of the model respectively [14]. However, all of these methods are just the approximations of the data distribution over each AL iteration and these approximations are only close to the real distribution when we have enough “representative” in the training set. Before reaching that state, the model is still an approximation and easy to query some outliers.

In our new approach of applying AL in reverse order, the model initialization and sampling bias problems is no longer need to be considered during the AL querying process and the model is guaranteed to be the best representation of underlying data rather than just the approximation. The detailed explanation is given in section 5.

1.4 Contributions

In this paper, we introduce one new practical consideration of AL that we call Reverse Active Learning (RAL), that is, the use of AL evaluation methods to remove the less informative instances and inconsistencies in the available training data to improve the best performance of supervised machine learning. The quality of the gold-standard and individual instances are evaluated by the system regardless of who annotated them. This approach is time efficient to normal AL especially when we need to select the large amount of training data to guarantee a better performance. More importantly, RAL can overcome the main obstacles in traditional AL: model initialization and biased training subset. These advantages of RAL are discussed in section 5 of this paper.

In much AL research, the experiments are set up by a set of training instances and a test set used to evaluate the performance achieved. The savings estimation are commonly computed as the percentage of the training instances selected during AL to achieve the same performance compared to using all the available training instances, or, as the difference in performance between AL and random selection for a given amount of annotation. This however implies that all the candidate training instances are annotated in advance. The more practical experiment when the annotations are available is how to select the best subset of the training data to build the model.

The traditional AL approach is based on using a small set of training data to build the initial model, and then using the model to query the most informative unlabelled instances to get their labels then retrain the current model. The central idea of RAL is to make use of available training data to generate better performance by removing less informative or inconsistent instances. The noisier the gold-standard is the better improvement the RAL can achieve.

2 Materials and Methods

Figure 1 shows the system architecture. The processing pipeline is described by an Experiment Descriptive Language (EDL) and executed by the Experiment

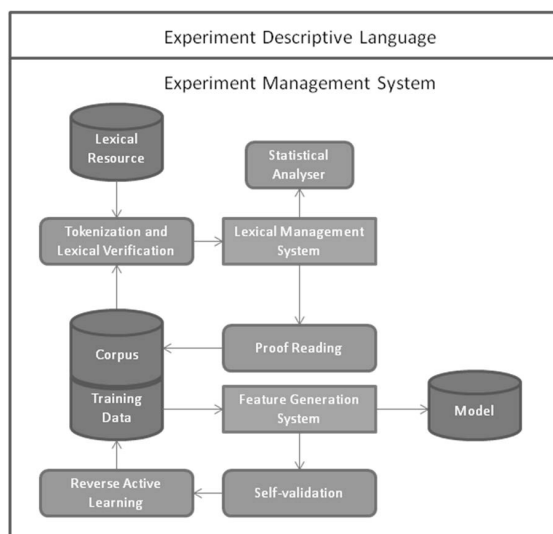


Fig. 1. System Architecture

Management System (EMS). The EMS ensures that a series of experiments and results are described, executed and compared in an expeditious and reusable way. An experiment is the execution of modules in a certain order. Experiment management is primarily the storage of an experiment in a deterministically reproducible form, as well as the storage of results associated with an experiment. A series of experiments need to be executed to compare the results. Instead of manually running the same modules on the original and refined corpora with different tags then comparing the results, using the EMS makes this process automatic and efficient. Using the EDL to describe an experiment allows a flexible execution order of modules within the same program as well as across multiple programs.

The Lexicon Management System (LMS) supports automated and manual resolution of unknown tokens. The LMS is a system developed to store the accumulated lexical knowledge and contains categorizations of spelling errors, abbreviations, acronyms and a variety of non-word tokens. It also has a web interface that supports rapid manual correction of unknown words with a high accuracy clinical spelling suggestor plus the addition of grammatical information and the categorization of such words into gazetteers [15]. The method of the clinical spelling suggestor is based on combining heuristic-based suggestion generation and ranking algorithms based on word frequencies and trigram probabilities. This approach achieved high accuracies on test data sets with over 93.5%.

The proof reading process can be applied directly or indirectly to improve performance of the text extraction system. Firstly, for the available annotation

tools, the proof correcting process is applied directly to the corpus with the expectation of finding more concepts from misspelling corrections, abbreviations and acronym expansions. However, during supervised learning experiments, this transformation should not be made in the original corpus because it would affect the offsets of tokens which in turn will lead to retrieval of incorrect annotations. Consequently, these corrections can be used indirectly as features which will support the model in learning the correct form of misspelt words (“medicla” and “medcial” refer to the same word “medical”) and variations of abbreviations (“amnt” and “amt” are both “amount”), and multiple acronyms of the same term (“ABG”, “ABGs” are both “arterial blood gases”).

The automated proof reading process is applied to clean the noisy corpus while RAL is used for removing less informative or inconsistencies in gold-standard training data. Traditional supervised learning generates the model based on training data with the best feature set. The popular methods to achieve the best feature set is by running experiments with n-fold cross-validation on training data with different feature sets and select the model with highest performance. In our method, active de-sampling is applied after the best feature sets are selected to refine the gold-standard and improve the best performance on the current training data. Two different refining strategies for two popular annotation tasks are developed and evaluated; these are self-validation for Named-entity Recognition (NER) and RAL for relationship classification.

3 Methods

3.1 Named-Entity Recognition Self Validation

Conditional Random Fields (CRFs) have been used to recognize clinical entities [16]. In clinical NER, the gold standard was created by manual annotation, which usually contains minor errors and inconsistencies. The self-validation (or reflexive validation) process can estimate the level of these inconsistencies in training data and extract them for further analysis and correction. The gold standard can be corrected for inconsistencies between annotations by using this process, which we also denote as “100% train and test”. This involves using 100% of the training set to build a model and then testing on the same set. As theoretically all data items used for training should be correctly identifiable by the model, any errors represent either inconsistencies in annotations or weaknesses in the computational linguistic processing. The former faults identify training items that are rejected, and the latter faults identify where to concentrate efforts to improve the preprocessing system. The self-validation is an automated process to identify inconsistent candidates and these candidates are then reviewed by human to remove noise that come from the training data.

3.2 Text and Relationship Classification

The base learner used in AL for text classification is a support vector machine (SVM) [17]. We have developed the AL framework for clinical data on top of the

Algorithm 1. Algorithm MCAL

Input

- A pool $U = \{u_1, u_2, \dots, u_n\}$.
- randomly selected training set L (at least one instance for each of m classes).
- number of query q for each iteration.

For each class c in m classes:

1. Init:
 - Set class value of c in L as 1 (positive).
 - Set all other classes in L as -1 (negative).
 - Using L to train the model M .
2. For $t = 1, 2 \dots q$
 - Get label v for the most informative instance u_i from U .
 - If u_i not in S : add u_i to S with value v .
 - If $v == c$: Add instance u_i to L with value of 1.
Else: Add instance u_i to L with value of -1.
 - Retrain model with L .
 - Remove u_i from U .

Return: Selected instances list S .

choice of the online AL algorithm (COMB) which contains 4 AL algorithms (Simple, Self confident, KFF and Balanced EE) [18]. Two enhancements of COMB have been implemented: (i) multi-class AL from COMB's AL binary classifier, (ii) Reverse Active Learning (RAL).

The "one versus the rest" method to build multi-class classifier (MCAL) is applied and can be executed in a parallel environment. The main purpose of AL is to select the most informative instances to train the model. In our method, by obtaining the union of selected instances over binary active learners, the best training set for multi-class text classification are generated so that the available multi-class SVMs tool such as libsvm can be used to build the final model [19]. This process is simpler than normal binary SVM AL combinations which usually require an additional probabilistic comparison step to resolve inconsistencies between different predictions on the same instance. The pseudo code for multi-class AL is illustrated in algorithm 1.

The m -class initial training data must have at least one instance for each class. Because the selection step for each class is independent, this process can be executed concurrently so that the nary AL will have similar running time to binary AL. The pseudo code for the RAL is presented in algorithm 2. The RAL algorithm follows similar steps in AL but shows modifications in the initial configuration and output selection strategy. The RAL starts with a large training data and outputs the list of instances that should be removed to achieve a better gold-standard to train the model.

Algorithm 2. Algorithm Reverse Active Learning

Input

- m-class full training set $L = x_1, x_2, \dots, x_n$.
- number of query q for each iteration.

Init

- Build index on L to store original class value.

For each class c in m classes:

1. Init:
 - Set class value of class c in L as 1 (positive).
 - Set all other classes in L as -1 (negative).
 - Using L to train the model M .
2. For $t = 1, 2 \dots q$.
 - Get the least informative instance x_i from L .
 - Remove instance x_i from L .
 - If $\text{value}(x_i) == 1$: $\text{original_class_value} = c$.
 Else: $\text{original_class_value} = \text{get_original_class}(x_i)$.
 - If x_i not in S : add x_i to S with $\text{original_class_value}$.
 - Retrain model with L .

Return: Removed instances list S .

4 Results

4.1 Clinical Concept Recognition

The experiments in this section were conducted to attempt the Clinical Concept Recognition task in the i2b2 2010 challenge on Clinical Information Extraction and CRF++ toolkit¹ was used to build the NER model [20]. During the training experiment, the best feature sets were selected after many cycles of n-fold cross validation. This section focuses on evaluate the effect of Self-validation process for refining the training data in order to boost up performance of supervised NER even when the best feature sets were decided.

With a self-validation process, approximately 150 errors in the training data were recognized and manually corrected. The three most frequent error types in concept annotation were: (1) missing modifier (any, some) within the noun phrase; (2) including punctuation at the end of concepts (full stop, comma, and hyphen); (3) missing annotation (false negative). When the model is trained with both correct and incorrect examples on some concepts, the probability of identifying the correct instances is definitely reduced and this might lead to series of wrong annotations on these concepts. Thus, automated detection and manual correction of errors in the training data contributed to improve the consistency and overall performance of the current model.

¹ CRF++. Yet another CRF toolkit. Software. [Http://crfpp.sourceforge.net](http://crfpp.sourceforge.net)

Proof reading and self-validation of the gold-standard brought approximately 1.5% improvement to the overall F-score of concept annotation supervised learning. This contribution was significant to make our result attain the state of the art of supervised learning method at i2b2 2010 challenge which was evaluated and reported by the organizers. Because the training and testing data come from the same source of corpora and annotators, the same kind of noise on testing data is unavoidable. Based on this assumption, we executed the same process on released testing data and over 200 similar errors were discovered. After correcting all of these errors for a better testing set, our system performance increased another 0.5%. Overall, the improvement of RAL for NER is approximately 2.0% as recorded in Table 1.

Table 1. System performance before and after Self-validation process

CONCEPT	TP	FP	FN	P	R	F
Problem	14423	2896	4127	83.28%	77.75%	80.42%
	14486	2752	3664	84.40%	80.25%	82.27%
Test	9945	1629	2954	85.93%	77.10%	81.27%
	10233	1504	2666	87.19%	79.33%	83.07%
Treatment	10134	1877	3426	84.37%	74.73%	79.26%
	10542	1722	3018	85.96%	77.74%	81.64%
OVERALL	34502	6402	10507	84.35%	76.66%	80.32%
	35661	5978	9348	85.64%	79.23%	82.31%

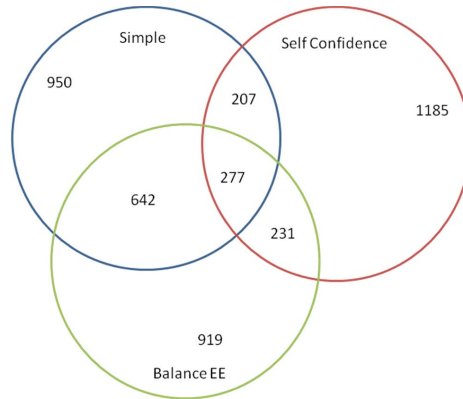
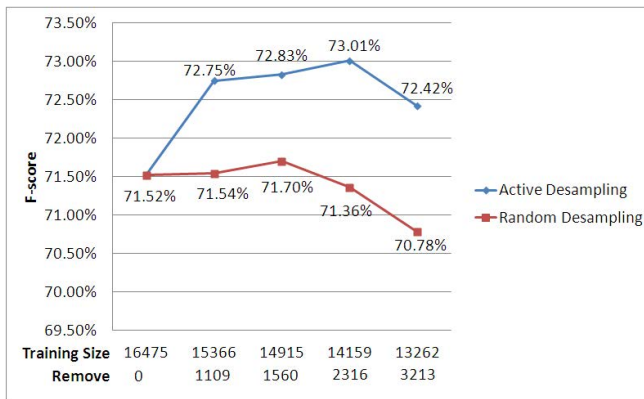
4.2 Text Classification

A multi-class AL algorithm was tested on the i2b2 2010 9-class relationship classification test data (no-relation is treated as one class). The simple Libsvm classifier is built mainly based on local context features (bag of words) and semantic features (concept and assertion type from the ground truth) supplied by the organizer. With these minimal and simple features, the classifier achieved an F-score of 68.86% test accuracy by using all 16475 instances of the training data (including no-relation instances). After adding proof reading as features, the result increased to 70.02% which was reported as equal second performance for relationship classification task at the challenge.

For a large-scale classification problem with many thousands instances and features as in i2b2 relationship classification task, linear kernel is usually a promising learning technique for this data [21]. Because of this assumption, further experiments are taken with the use of Liblinear as a base classifier rather than Libsvm with non-linear kernels [22]. Liblinear is an open source library for large-scale linear classification with ease of solvers and parameters selection. The use of Liblinear classifier improved the F-score by 1.5%. Three of the 4 AL algorithms that have high performance on the i2b2 clinical data set are Simple, Self confidence and Balance EE while the KFF algorithm returns a significantly lower result. Table 2 shows the comparisons of the AL algorithms with a maximum of 500 queries for each class. The combined training data is a union of all

Table 2. Performance of multi-class AL

Method	Training size(AL instances)	Training percent	F-Score
Simple	2076	12.60%	66.25%
Self-confidence	1900	11.53%	64.58%
Balance EE	2069	12.56%	65.76%
Combined	4361	26.47%	68.03%

**Fig. 2.** The distribution of selected instances over 3 AL algorithms: Simple, Self Confidence and Balance EE**Fig. 3.** Active desampling and random desampling on test data

instances selected by three AL algorithms. The overlapping selected instances between the 3 algorithms are shown in Figure 2 where Simple and Balance EE show the greatest overlap in selected instances.

RAL applied with the Simple method can improve the best result on supervised learning (1.5%) by removing over 1500 less informative instances. The final result after using Liblinear and applied RAL is comparable to the current state

of the art of the dataset who also using Liblinear as base classifier. Furthermore, the i2B2 challenge papers have been studied and there is no AL approach for relationship classification used in any of the top submitted processing systems. Figure 3 shows that the F-scores of active desampling dominates random desampling with the same number of removed instances.

5 Discussion

Our methods of applying RAL can improve the supervised learning performance from 1.5% to 2% on two popular clinical information extraction tasks: clinical NER and relationship classification. These improvement may not be significant in general, but these are the additional improvements to the best performance of the available training data. In supervised learning, experiments are set up in order to select the best feature sets to build the model. The closer a model is to the best possible performance, the harder (or smaller) is the improvement that can be made. In our experience, after the best feature sets are determined for the available training data, the contribution of 1.5% to 2% is remarkable. Furthermore, the improvement of one percent and above was considered as statistical difference in the i2b2 challenge’s ranking system [20].

From the experiment’s results, with equivalent number of selected instances during AL processes (approximately 2000 instances), the performance of RAL is at least 4.84% higher than any single AL setting and 3% higher than the union of all three popular AL methods with double selected instances (table 2). The most important aspect is RAL can produce a better result than using all available training data right at the very first removing instances as seen in Figure 3. In order to normal AL to achieve the same performance with this clinical dataset, the number of data evaluation and selection process might be up to 14,000 cycles which will take much more time to complete.

For the relationship classification, our classifier is only based on very simple, easy implemented feature sets and mainly using resources supplied by the organizer while the winner used a wide range of complicated features from external syntactic (chunking and parsing) tools; medical ontology and Wikipedia knowledge [23]. Without using any external and huge resources as features, our classifiers is more practical to be installed and evaluated in real clinical environment.

RAL starts with a full training set which is best in initial model representation and informativeness for instance evaluation. Then the model sequentially removes small portions of less useful instances which causes minor impact on the data distribution. Consequently, the initial model building and training bias are no longer considered as obstacle to RAL. In this situation, complicated algorithms to deal with these problems are also no longer needed. The original and “Simple” AL method based on Uncertainty estimation is a reasonable choice in the RAL setting. As shown in the result, with a noisy gold standard such as in clinical domain, application of RAL is preferred to normal AL because the simple RAL can generate better results than supervised learning right from the first steps of removal.

6 Conclusions

In this study, a general system architecture for refining the training data is developed and evaluated. RAL is used as a final step to improve the quality of training data after the best feature sets are decided. Our work focuses on the clinical domain which suffers from a high level of noise in both the corpus and training data. However, our method and system architecture are easy to adapt to any domain with modification of resources and feature sets for supervised learning.

We also introduce and illustrate a simple combination of binary active learners into multi-class AL as well as the application of AL in reverse order to improve the best performance of supervised learning. To sum up, we recommend the use of proof reading and RAL in supervised learning for a noisy domain such as clinical natural language processing. RAL is not only simple and suitable but also can generate better performance and is less prone to biased training problem.

References

1. Settles, B.: Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin-Madison (2010)
2. Blum, A.L., Langley, P.: Selection of relevant features and examples in machine learning. *J. Artif. Intell.* 97(1-2), 245–271 (1997)
3. Fung, G., Mangasarian, O.L.: Data selection for support vector machine classifiers. In: *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2000*, pp. 64–70. ACM, New York (2000)
4. Donmez, P., Carbonell, J.G., Schneider, J.: Efficiently learning the accuracy of labeling sources for selective sampling. In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2009*, pp. 259–268. ACM, New York (2009)
5. Nallapati, R., Surdeanu, M., Manning, C.: Correlative learning: Learning from noisy data through human interaction. In: *IJCAI Workshop on Intelligence and Interaction* (2009)
6. Lewis, D., Gale, W.: Training text classifiers by uncertainty sampling. In: *Proceedings of ACM-SIGIR Conference on Information Retrieval*, pp. 3–12 (1994)
7. Scheffer, T., Decomain, C., Wrobel, S.: Active Hidden Markov Models for Information Extraction. In: Hoffmann, F., Adams, N., Fisher, D., Guimarães, G., Hand, D.J. (eds.) *IDA 2001. LNCS*, vol. 2189, pp. 309–318. Springer, Heidelberg (2001)
8. Settles, B., Craven, M.: An analysis of active learning strategies for sequence labeling tasks. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2008*, pp. 1070–1079. Association for Computational Linguistics, Stroudsburg (2008)
9. Seung, H.S., Oppen, M., Sompolinsky, H.: Query by committee. In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory, COLT 1992*, pp. 287–294. ACM, New York (1992)
10. McCallum, A., Nigam, K.: Employing em and pool-based active learning for text classification. In: *Proceedings of the Fifteenth International Conference on Machine Learning, ICML 1998*, pp. 350–358. Morgan Kaufmann Publishers Inc., San Francisco (1998)

11. Vlachos, A.: A stopping criterion for active learning. *Computer Speech and Language* 22(3), 295–312 (2008)
12. Olsson, F., Tomanek, K.: An intrinsic stopping criterion for committee-based active learning. In: *Proceedings of the Thirteenth Conference on Computational Natural Language Learning, CoNLL 2009*, pp. 138–146. Association for Computational Linguistics, Stroudsburg (2009)
13. Kang, J., Ryu, K., Kwon, H.C.: Using Cluster-Based Sampling to Select Initial Training Set for Active Learning in Text Classification. In: Dai, H., Srikant, R., Zhang, C. (eds.) *PAKDD 2004. LNCS (LNAI)*, vol. 3056, pp. 384–388. Springer, Heidelberg (2004)
14. Roy, N., McCallum, A.: Toward optimal active learning through sampling estimation of error reduction. In: *Proceedings of the Eighteenth International Conference on Machine Learning, ICML 2001*, pp. 441–448. Morgan Kaufmann Publishers Inc., San Francisco (2001)
15. Patrick, J., Sabbagh, M., Jain, S., Zheng, H.: Spelling correction in clinical notes with emphasis on first suggestion accuracy. In: *2nd Workshop on Building and Evaluating Resources for Biomedical Text Mining*, pp. 2–8 (2010)
16. Lafferty, J., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data (2001)
17. Cristianini, N., Shawe-Taylor, J.: *An introduction to support Vector Machines: and other kernel-based learning methods*. Cambridge Univ. Pr. (2000)
18. Baram, Y., El-Yaniv, R., Luz, K.: Online choice of active learning algorithms. *Journal of Machine Learning Research* 5, 255–291 (2004)
19. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2, 27:1–27:27 (2011), Software <http://www.csie.ntu.edu.tw/>
20. Uzuner, O., South, B.R., Shen, S., DuVall, S.L.: 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *J. Am. Med. Inform. Assoc.* 18(5), 552–556 (2011)
21. Chih-Wei Hsu, C.C.C., Lin, C.J.: A practical guide to support vector classification. Technical report (2003–2010)
22. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: A library for large linear classification. *J. Mach. Learn. Res.* 9, 1871–1874 (2008)
23. Rink, B., Harabagiu, S., Roberts, K.: Automatic extraction of relations between medical concepts in clinical texts. *J. Am. Med. Inform. Assoc.* 18(5), 594–600 (2011)