Active Learning for Named Entity Recognition in Clinical Text

By

Yukun Chen

Dissertation
Submitted to the Faculty of the
Graduate School of Vanderbilt University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY
in

Biomedical Informatics
August, 2015
Nashville, Tennessee

Approved:

Joshua C. Denny, M.D., M.S.
Hua Xu, Ph.D.
Thomas A. Lasko, M.D., Ph.D.
Qiaozhu Mei, Ph.D.
Qingxia Chen, Ph.D.

To my father, Dr. Jingde Chen, and my mother, Misha Cao,

for their endless love, support and encouragement

and

To the memory of my grandfather, Yunping Cao

ACKNOWLEDGEMENTS

First and foremost, I would like to thank Professor Hua Xu for his mentorship during the past four years. This dissertation and my other accomplishments at Vanderbilt would not have been possible without his guidance and support throughout my entire PhD journey. He taught me how to become a great informatician, which substantially stimulated my growth as a scientist.

I also want to thank other committee members for their advices on my dissertation research. Dr. Josh Denny guided me to study clinical NLP systems and their applications in the medical domain. He brought me to the world where the collaboration between clinicians and informaticians is so important for the enhancement of healthcare and biomedical research. Dr. Tom Lasko provided me intuitive study instructions on the topic modeling and clustering techniques, which essentially improved the model quality to the next level. Dr. Qiaozhu Mei shared his insights on the development of novel active learning methods, which turned out to be successful. Dr. Qingxia Chen taught me how to use appropriate statistical tools to assess models and gave very useful ideas on the annotation cost modeling. I am very grateful to have such an incredible PhD committee.

I could not have made a significant progress for this dissertation within such a short time without the assistants from the following experts in Dr. Xu's lab at UTHealth at Houston. Jingqi Wang generated and tested a portion of codes to embed an existing annotation interface into our system. Ky Nguyen and Tolulola Dawodu are the two nurse students who dedicated hours of training and annotations to the user studies. Ruiling Liu provided supports in the analysis of the annotation cost models. Dr. Sungrim Moon and Dr. Trevor Cohen participated in discussions for

the study design and result analysis. Dr. Anu Gururaj and Dr. Irmgard Willcockson performed the copy editings on this dissertation. Therefore, this is truly the teamwork accomplishment.

This work was made possible through National Library of Medicine grant 2R01LM010681-05. The annotated datasets were obtained from 2010 i2b2/VA NLP challenge and I would like to thank the organizers for sharing the datasets.

I also want to acknowledge the exceptional study environment at Vanderbilt DBMI. Dr. Cindy Gadd directed me on the path to success in completing the PhD program. Rischelle Jenkins made sure all small steps to graduation were not missing. Terri DeMumbrum and Belinda Ballard also provided great support for my transition from Vanderbilt to UTHealth at Houston. Moreover, students at Vanderbilt DBMI are outstanding. I had a great time at Vanderbilt with Ravi Atreya, Robert Carroll, Yaoyi Chen, Michael Kochen, Dong Wang, Pedro Teixeira, Wiley Laura, Jacob VanHouten, Haresh Bhatia, Lina Sulieman, and more. It was a great experience to work with them in the classes and/or research projects.

Last but not least, I would like to thank my family and friends who are always with me and provide immediate supports whenever needed.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER 1

Introduction

In the medical domain, the rapid growth in the use of electronic health records (EHR) has made a large amount of electronic textual data available for clinical research. It drives the development of new technologies, such as natural language processing (NLP) and machine learning (ML), to unlock important information from clinical text for further analyses. However, statistical NLP systems often require large numbers of annotated samples in order to build high performance ML models. Building large-scale, high-quality corpora is very time consuming and costly in the medical domain, because it often requires manual annotation by domain experts. Therefore, methods that can help build high-performance ML models but require fewer annotations are highly desirable in clinical NLP research.

Active learning (AL), which selects the most informative samples for annotation (as opposed to using random sampling) to iteratively build ML models, could be one of the solutions for addressing the above challenge. AL has been widely studied in the open domain, as well as biomedical NLP tasks, such as assertion classification for clinical concepts [1], word sense disambiguation in biomedical literature [2], and phenotyping from electronic health records [3]. Despite the fact that these studies demonstrated the potential of AL for achieving high-quality ML models with reduced annotation cost, all these studies were conducted in a simulated environment, which assumes that annotation cost for each sample is identical. In reality, however, annotation cost (i.e. the time required by an annotator) can be very different from one sample to another and from one user to another user.

This dissertation research aims to develop novel AL algorithms and practical AL systems for the clinical named entity recognition (NER) task. NER is a fundamental task for many NLP applications. But there is no AL research on NER in the medical domain. We systematically investigated AL for clinical NER and our work consists of three major parts: 1) we conducted a simulation study, using existing annotated datasets, to evaluate existing and new AL algorithms for clinical NER (Chapter 2); 2) we developed an AL-enabled annotation system for clinical NER and conducted a user study to assess the benefit of AL (vs. random sampling) in real-time annotation for building NER models (Chapter 3); and 3) based on results from 2), we further refined our AL algorithms by developing more sophisticated annotation time models and evaluated them using both simulation and user studies (Chapter 4). Our final AL enabled NER system showed better performance than random sampling in the real-world annotation task, demonstrating the potential of AL in clinical NER.

This chapter provides a literature review. We start with NLP and ML-based NER in the medical domain and then introduce relevant aspects of AL for text processing in both open domains and the biomedical domain, including three of our previous studies of AL on other biomedical text processing tasks.

1.1 Natural language processing in the medical domain

NLP converts free text into structured forms to support computational applications. In the open domain, many NLP technologies benefit people's daily lives. For example, web search engines (i.e. Google, Bing, Baidu, etc)[4-6] are currently the most valuable information resources. Using a very simple interface, users type a phrase and get instant returns of the links with the

information they want the most. Apple Siri, an intelligent personal assistant and knowledge navigator, uses a natural language user interface to understand the user's audio question, answer questions, make recommendations, and request information from the internet for the user [7]. IBM's Watson [8] won the Jeopardy! Challenge in 2011 against two of the best human opponents in Jeopardy by applying the technology of automatic question answering in the general domain.

In the medical domain, the rapid growth in the use of clinical notes in EHRs is a strong incentive for the development of clinical NLP [9]. With the large amount of structured knowledge extracted from the narrative using NLP, many clinical studies have been enhanced, for example disease phenotypes and patient cohort identification [10, 11], decision support [5], and drug repurposing [12]. Identification of clinical concepts or clinical NER is an important task to build clinical NLP systems. For example, much work has been done to extract clinically important entities from clinical text, such as diseases, medications, procedures, and laboratory tests [13-15].

Some existing clinical NLP systems, including MedLEE [16, 17], MetaMap [18], cTAKES [19], and KnowledgeMap [20], not only extract various types of clinical entities, but also map them to concepts in the controlled vocabularies the Unified Medical Language System (UMLS) [21]. MedLEE, developed by Friedman et al. in the 1990s at Columbia University, is mainly a semantic rule-based system. It was initially designed to extract clinical attributes from radiological reports [22], and then extended to mammography [23], discharge summaries [24, 25] and pathology [26]. MetaMap was developed initially for biomedical literature mining and has recently also been used for clinical note processing. cTAKES, a comprehensive clinical NLP system, combines both rule-based and machine learning techniques under the IBM UIMA

framework. The KnowledgeMap, developed by Denny et al. [27, 28] at Vanderbilt University, is another clinical NLP system built to extract clinical concepts with their section headers (by SecTag [29]) and negation status (by NegEx [30]) in documents and map them to UMLS concept unique identifiers. In addition, researchers have also developed various tools for clinical NLP tasks for negation detection [30, 31], medication extraction, [32, 33] and temporal expressions [34, 35] in clinical text.

1.2 Machine learning-based named entity recognition in clinical text

NER is a fundamental task for information extraction, which is often used to locate phrases in clinical text and classify them into pre-defined categories of medical concepts, for example medical problem, treatment, and lab test. Many of the clinical NLP systems perform reasonably well at this task by utilizing symbolic NLP or rule-based approaches.

Recent studies have shown that ML-based models, which are trained on annotated datasets, have the potential to achieve better performance in clinical NER tasks. Patrick et al. [36] developed a machine learning model to extract medication-related entities. His system achieved an F-measure of 85.65% for the evaluation of exact match medication entry, which was superior to the other participants in the 2009 i2b2 NLP challenge. Both Brujin et al. [37] and Jiang et al. [38] systematically investigated ML-based approaches for recognizing broader types of clinical entities and presented their promising results of 85.23% and 83.91% in F-measure, respectively, as the top two teams in the clinical concept extraction task in the 2010 i2b2 NLP/VA challenge.

Conditional random field (CRF) [39] and support vector machine (SVM) are the most widely

used ML models in NER tasks. Both build the most effective clinical concept extraction systems [14]. The structural SVM (SSVM) [40, 41] is another NER algorithm that merges the advantages of both CRF and SVM for solving the sequence-labeling problems. Recent studies demonstrated that SSVM performed slightly better in recognizing clinical entities from discharge summaries [42-44].

The ML-based NER uses a classification algorithm (e.g. CRF, SVM, or SSVM) to sequentially label a sentence. The labels of the individual words or tokens in the sentence are commonly represented in the "BIO" format, where "B" represents the label for the beginning of an entity, "I" the inside of the entity, and "O" for outside of the entity. Figure 1 shows an example of "BIO" representation of problem, treatment, and lab test entities for each word/token in a sentence.

She	was	ultimately	changed	to	Levaquin	for	a	possible	early	pneumonia	pending	cultures	.
O	O	O	O	O	B-treatment	O	O	O	B-problem	I-problem	O	B-test	O

Figure 1. An example of "BIO" representation of problem, treatment, and lab test entities for each word/token in a sentence

The training of the ML-based NER model extracts the pattern representing the relationship between the sequential words with their features and their labels. Therefore, the features are extremely important in determining the quality of the NER model. A variety of features extracted from the raw text data were systematically studied for the improvement of the NER model [38]. They include bag of words, prefix and suffix, syntactic features (e.g. part-of-speech tags), and semantic features (e.g. semantic classes in UMLS). In addition, unsupervised analysis for word representation, such as brown clustering [42, 43], has also shown improvement for the clinical

NER task. [37]

1.3 Active learning

ML-based approaches, however, often require large annotated corpora, which are time-consuming to build due to the manual effort required for the task. In the clinical domain, clinicians (e.g. physicians or nurses) are required to conduct the annotation task, thus the cost of annotation could be very high. In the general English domain, pool-based AL strategies [45] have benefited many NLP tasks which require annotation from a large pool of unlabeled data to construct the supervised ML model. Examples include word sense disambiguation [46], text classification [47], and information extraction [48].

In recent years, several studies have also applied AL to text processing tasks in the clinical domain. Figueroa et al. [49] validated AL algorithms as a way to reduce the size of training sets to yield expected performance in medical text classification tasks on five datasets. We also developed and evaluated AL paradigm on multiple biomedical NLP tasks, such as assertion classification of concepts in clinical text [1], supervised word sense disambiguation in MEDLINE [2], and high-throughput phenotyping tasks for EHR data [3]. The conclusion shared by these studies is that AL could reduce annotation cost while improving the quality of the classification model, as compared to the passive learning approach (random sampling).

1.3.1 Pool-based active learning framework

The pool-based AL approach to classification [45] is practical for many real-world learning problem domains, including medicine. The learner can access a large quantity of unlabeled data as a pool with low cost, iteratively select samples from the pool, and request their true labels. An AL system mainly consists of a classification model and an active sample selection or a querying algorithm. The classification model is built by traditional supervised machine learning algorithms. The model is trained by using the labeled instances (training set) and is then applied to the new unlabeled instances (test set) to predict class labels. The second core component of AL is the querying method. In general, there are two types of learners: active learner and passive learner. The passive learner just uses a random sampling method, which queries the labels of instances randomly selected from the pool of unlabeled samples, without considering the information about samples in the pool. The active learner, on the other hand, will select the instances that are the most promising in improving the predictive performance of the model.

An AL protocol is often used for a given dataset and a querying algorithm:

(1) Generate an initial labeled set $L = L_0$, unlabeled set (the pool) $U = U_0$, and a test set T.

(2) Train a predictive model based on L and infer the class label for each sample in U and T.

(3) Score the samples in U based on the querying algorithm and label the top $b(i)$ samples in U, where $b(i)$, the *batch size* of AL, is the number of querying samples at iteration i.

(4) Add the $b(i)$ sample(s) with label(s) to L and remove from U.

(5) Iterate steps (2) to (4) until the stop criterion is met.

(6) Finally, report the classification performance (e.g. AUC, Accuracy, or F-measure) for the prediction of T at each iteration i, generate the learning curve that plots the classification

performance as a function of annotation cost (e.g. number of training samples or annotation time at each iteration *i*), and compute the global score based on the learning curve.

1.3.2 Active learning methods

The main issue for the active learner is how to find the good queries from the pool for better classification performance. Many variations of the AL (querying) algorithms exist, which can be classified into six main types: uncertainty sampling [50], query-by-committee (QBC) [51], expected gradient length (EGL) [52], Fisher information [53], estimated error reduction (EER) [54] and information density [48].

The uncertainty sampling is the simplest and most commonly used query algorithm. The active learner using an uncertainty sampling algorithm tends to query the samples which are least certain about their labels. The QBC algorithm tends to select the samples that generate the most disagreement from a committee of models. The models in the committee are all trained on the same labeled set but represent different hypotheses. The level of disagreement can be computed based on different voting strategies, such as vote entropy [55] and Kullback-Leibler (KL) divergence [56]. The EGL algorithm tends to select the samples that would have produced the greatest change to the current model if their labels were known. The EER algorithm is similar to EGL, in that it tends to query the samples that maximally reduce the generalization error of model. The Fisher information algorithm tends to select the samples, which could indirectly reduce the generalization error by minimizing the output variance. It is equivalent to selecting the sample that could maximize its Fisher information. The information density algorithm tends to query the most representative samples based on the similarity function. It can sometimes be

combined with the uncertainty sampling algorithm to select the most informative samples that are not only uncertain, but also the most representative of the data (e.g. centers of dense regions of data). Detailed information about these algorithms can be found in an AL literature survey [57].

Some of the algorithms are computationally expensive and not practical, such as expected gradient length, Fisher information, and estimated error reduction. QBC is sensitive to the type of classification models selected and also computationally expensive when the number of committees is high and each of the committee is expensive to build. In this study, we mainly focused on uncertainty sampling and diversity sampling (similar to information density), which are more straightforward to implement and fast to compute. In the following sections, we review our previous studies of applying AL to biomedical text processing.

1.3.3 Simulated active learning studies

The majority of the AL studies were based on simulation. They used the pre-annotated dataset, which was split into two sets: 1) a pool of samples to be queried and 2) the evaluation set. The labels of the data in the pool were considered unknown at the beginning of the AL process. When a querying algorithm selected the samples from the pool, their labels were unlocked and the selected samples were added to the training set. In addition, the batch size was pre-set and the AL process stopped when all samples in the pool were queried.

Simulated studies often assume that the cost of training a model is equal to the number of samples in the training set. Some studies assume that the annotation cost per sample is not the

same but a known number (e.g. annotation cost for a sentence is the same for sentences with the same length).

Method assessment in the simulated AL studies is based on a learning curve, which plots the classification performance (e.g. area under the ROC curve score (AUC), accuracy, or F-measure) computed on the evaluation set, as a function of the size of the training set or other types of cost. The area under the learning curve (ALC) score, which was the main evaluation metric in the AL challenge in 2010 [58], is computed as a global score for each querying method. The ALC score is computed based on the following function:

$$\text{ALC score} = \frac{\text{ALC} - Arand}{A\max - Arand}$$

where *A*max is the area under the best achievable learning curve (e.g. 1.00 AUC on all points of the learning curve) and *A*rand is the area under the learning curve obtained by random prediction (e.g. 0.50 AUC on all points of the learning curve). The learning curve of two neighbor points is interpolated linearly.

1.3.4 Active learning in practice

Few prior studies have applied AL to real world tasks in the medical domain and evaluated its performance. In the open domain, researchers have built tools with AL implementations to support different real-world text processing tasks. Relevant studies are described below.

DUALIST [59] is an AL annotation interface that queries and learns from annotations on both features and instances for the classification tasks, such as text word sense disambiguation, twitter

filtering and sentiment analysis. The authors used DUALIST as an example to transform AL to the concept of interAL. In the interAL, a machine can ask questions to obtain information regarding not only the label of instance but also features and utilize the answers from the human users to train a classifier faster. The learning engine is based on the multinomial naïve Bayes, a generative model that can learn from the labels of both the feature and the instance. In addition, DUALIST also implements the Expectation-Maximization (EM) algorithm and semi-supervised learning to exploit the unlabeled data, which could be labeled by the machine and used as the labeled data with zero human effort.

This study also includes user experiments, where human annotators use DUALIST to label features and sentences as part of the loop in the AL process. The user study compares AL, interAL, and passive learning in three classification tasks with five users. The results show that interAL generated better learning curves than random sampling did for most of the users. Moreover, features annotations took less time than instance annotations. However, how the annotated features contributed to the classifier training compared to the annotated instances was unknown. Their annotation data shows that users annotated instances more accurately in the case of AL. Users also skipped more instances in interAL and AL than passive learning, indicating that the interactive and active queries could be more ambiguous.

The classification tasks studied in this paper were fairly simple as the actively-trained classifiers were able to reach 90% accuracy after only a few minutes of annotation effort. The study only reported the average annotation time per instance, but its variance is unknown. The difference of annotation time between instances in this study is not large. How AL would perform in a harder task (e.g. clinical NER) when the annotation time per instance is large is also unknown.

In 2008, Settles et al. [60] reported a detailed empirical study of AL with real annotation costs in four real-world domains. In some domains, the annotation cost is known or fixed per sample. This paper studied how AL performed in the domain where annotation cost was unknown in advance.

They conducted user study and simulation experiments over five tasks, including NER and relation extraction in a News corpus, multiple-instance labeling in images, subjectivity handling in speculative text corpus, and contact extraction from an email corpus. The annotation data was collected and the distribution of annotation time per instance over multiple annotators was analyzed. The most interesting questions discussed in this paper are "Can annotation time be accurately predicted?" and "Can we improve AL by utilizing cost information?"

For the first question, they evaluated several regression cost models that achieved reasonable results over different tasks (with correlations ranging from 0.29 to 0.85). For the second question, they tested a cost-sensitive AL approach with a simple querying heuristic that divides the utility measure (e.g. entropy-based uncertainty sampling) by the predicted cost of the instances. They found that AL methods without cost information performed no better than random sampling. However, in some instances, the learning curves could be improved if the annotation cost variables during the AL were taken into account. They did an additional experiment using the true annotation cost in the AL and obtained the best performance in most of the tasks. Moreover, when the actual annotation cost shows considerable variation, an accurate annotation cost model is more helpful for AL. However, further investigation is required.

In the same year, Haertel et al. [61] also presented a practical cost-conscious AL approach based on return on investment (ROI). They evaluated the ROI based-AL on a part-of-speech tagging

task and showed that ROI reduces up to 73% in annotation hours over random sampling. However, they used a fixed heuristic cost model only. The performance with a different cost model and/or human annotator is unknown.

Baldridge and Palmer [62] conducted an interesting AL experiment with one expert and one novice annotators for morpheme annotation in a rare language documentation task. They found that the expert annotator was more efficient with an uncertainty-based active learner, but semi-automated annotations were of little help. On the other hand, the novice annotator was more efficient with a passive learner based on random sampling, but semi-automated annotations were beneficial.

1.4 Active learning in biomedical text processing

Recently, several studies have applied AL to biomedical text processing tasks. We review the relevant studies in the following paragraphs.

Figueroa et al [63] applied AL to two clinical text classification tasks, such as smoking status and depression status extraction, and one non-clinical classification task using SVM as a classifier. They implemented distance-based (DIST), diversity-based (DIV), and a combination of both AL algorithms (CMB), and compared the performance with passive learning. Their results showed that DIST and CMB algorithms performed significantly better than passive learning. They also suggested that DIV is more suitable on data with higher diversity and DIST on data with lower uncertainty.

Miller et al [64] explored various AL methods for clinical coreference resolution that fit more realistically into the coreference annotation workflows. This paper indicated that the traditional AL approach may not be feasible for this task since coreference annotations require contextual information. Their work showed that instance selection worked well for coreference resolution, introduced several metrics for document selection, and proposed a hybrid selection approach that preserves the benefits of instance selection while offering the potential of being applicable to real annotation.

Wallace et al [65] studied an application of AL to the problem of biomedical citation screening for systematic reviews at the Tufts Evidence-based Practice Center. They proposed a novel AL strategy that exploited a priori domain knowledge provided by the expert (specifically labeled features) and extended this model via a Linear Programming algorithm for situations where the expert can provide ranked labeled features. Uncertainty sampling with SVM performed better than random sampling when using accuracy as a model evaluation metric; however, it was not true for evaluating recall of the model, which is important for citation screening. This was due to the imbalanced class and the hasty generalization problem. The result showed that using prior knowledge could positively guide AL.

In addition, our group has carried out studies of AL on clinical and biomedical NLP tasks that have been published. Detailed descriptions of each study are provided below:

1.4.1 Applying active learning to assertion classification in clinical text

The first paper [1] presented one of the earliest applications of AL to clinical text processing. Specifically, we developed new AL algorithms and applied them to the assertion classification task for concepts in clinical text.

We used the manually annotated training set for concept assertion classification provided by the 2010 i2b2/VA NLP challenge [14]. The assertion classification task was to assign one of the six labels ("absent", "associated with someone else", "conditional", "hypothetical", "possible", and "present") to medical problems identified from clinical documents. We converted the multi-class assertion classification task into a binary classification problem, by considering "present" to be the positive class and all others as the negative class. We implemented several existing and newly developed AL algorithms, such as least confidence (LC), least confidence with bias (LCB), least confidence with dynamic bias (LCB2), and the novel model change sampling algorithms (LCMC, LCBMC, LCB2MC). Their uses were assessed with other methods, such as information density and baseline random sampling.

Results showed that when the same number of annotated samples were used, AL strategies could generate better classification models (best ALC – 0.7715 by LCBMC) than the passive learning method (random sampling) (ALC – 0.7411). Moreover, to achieve the same classification performance, AL strategies required fewer samples than the random sampling method. For example, to achieve an AUC of 0.79, the random sampling method used 32 samples, while our best AL algorithm (LCBMC) required only 12 samples, a reduction of 62.5% in the manual annotation effort. This study demonstrated that AL technologies can be effectively applied to clinical text classification tasks, improving performance and reducing annotation effort. New

querying methods developed here also showed good performance on the concept assertion classification task.

1.4.2 Applying active learning to supervised word sense disambiguation in MEDLINE

This was the first study to explore the use of AL in supervised WSD tasks in the biomedical domain [2]. In this study, we developed Support Vector Machines (SVM) classifiers to disambiguate 197 ambiguous words and abbreviations in an existing benchmark dataset, MSH WSD collection, derived from MEDLINE abstracts. Three different uncertainty sampling-based AL algorithms (LC, Margin, and Entropy) were implemented with the SVM classifiers and were compared with a passive learner based on random sampling. For each ambiguous term and each learning algorithm, an average learning curve was generated to plot the accuracy computed from the test set as a function of the number of annotated samples used in the model via a 10-fold cross-validation.

Our experiments showed that active learners significantly outperformed the passive learner, showing better performance for 177 out of 197 (89.8%) WSD tasks. However, there was no significant difference among three active learners for words with more than two senses. Further analysis showed that to achieve an average accuracy of 90%, the passive learner needed 38 samples, while the active learners needed only 24 annotated samples, a 37% reduction of annotation effort. Moreover, we analyzed cases where AL algorithms did not achieve superior performance and discovered three causes: (1) poor model in early learning stage; (2) easy WSD cases; and (3) difficult WSD cases, which provide useful insight for future improvements. This

study demonstrated that integrating AL strategies with supervised WSD methods could effectively reduce annotation cost and improve the disambiguation models.

1.4.3 Applying active learning to high-throughput phenotyping in electronic health record data

This paper investigated the use of AL in ML-based phenotyping algorithms [3]. Generalizable, high-throughput phenotyping methods based on supervised ML algorithms could significantly accelerate use of EHR data for clinical and translational research. However, they often require large numbers of annotated samples, which are costly and time-consuming to produce.

We integrated an uncertainty sampling AL approach with SVM-based phenotyping algorithms and evaluated its performance using three annotated disease cohorts including rheumatoid arthritis (RA), colorectal cancer (CRC), and venous thromboembolism (VTE). We investigated performance using two types of feature sets for each phenotype: 1) unrefined features, which contained all the clinical concepts extracted from the notes and billing codes; and 2) a smaller set of refined features selected by the domain experts. The performance of the AL-based approach was compared with a passive learning approach based on random sampling using area under the learning curve.

Our evaluation showed that AL outperformed passive learning on all three phenotyping tasks. When unrefined features were used in the RA and CRC phenotyping tasks, AL reduced the number of annotated samples required to achieve an area under curve score (AUC) of 0.95 by 68% and 23%, respectively. AL also reduced the number of samples needed to achieve optimal performance for VTE by 68% when using refined features; however, VTE algorithms only

achieved an AUC of 0.70. As expected, refined features improved the performance of phenotyping classifiers and required fewer annotated samples. This study demonstrated that AL can be useful in ML-based phenotyping methods. Moreover, AL and feature engineering based on domain knowledge could be combined to develop efficient and generalizable phenotyping methods.

1.5 Summary

In this chapter, we reviewed relevant literature about AL in biomedical text processing. AL has received increasing attention in the medical domain as a potential solution to address the bottleneck of annotation cost for statistical NLP methods. However, none of the previous studies investigated AL for clinical NER, an important step for many clinical NLP applications. Moreover, none of the studies evaluated the use of AL in real-world annotation processes. Therefore, it is critical to conduct systematic studies to assess all these aspects, in order to demonstrate and propose practical AL solutions for building NER systems.

CHAPTER 2

A Simulation Study of Active Learning Methods for Named Entity Recognition in Clinical Text

2.1 Introduction

The goal of AL for NER would be to select informative sentences from the pool and hopefully save annotation cost. In the literature, some AL studies particularly focused on NER tasks and provided insightful information for approach design. An AL study by Kim et al. [66] presented a new AL paradigm for NER that considered both uncertainty of the classifier and the diversity of the corpus. For uncertainty sampling, they implemented N-best sequence entropy, which was computed based on N the most likely label sequences for the unlabeled samples. For diversity sampling, they considered three levels of information, including NP chunk, Part-of-Speech tag, and the word itself, to compute the similarity between sentences. The combined performance was better than random sampling. However, their diversity-based method alone did not outperform random sampling. Settles and Craven [67] conducted a large-scale empirical study of AL for NER by evaluating seventeen methods in six corpora. They used random sampling and long sentence sampling methods as the two baselines, and multiple AL methods, including six uncertainty-sampling approaches, six query-by-committee methods, and other methods such as information density, Fisher information, and expected gradient length. Most of the AL algorithms performed better than baselines, indicating the promise of AL in NER. One limitation of these existing studies is that they are simulated studies that assumed that the annotation cost for each sentence was the same. In reality, however, annotation cost could be different from one

sentence to another. Informative sentences selected by AL algorithms (using uncertainty sampling for example) could require more annotation time just because they are longer sentences. There have been mixed results for doing cost-sensitive AL in the literature to tackle realistic annotation costs [68-70].

Nevertheless, all the above studies of AL in NER were from open domains and to the best of our knowledge, there is no AL on clinical NER tasks. In this study, we conducted simulated AL experiments using an existing clinical NER corpus with annotated medical problems, treatments, and lab tests in clinical notes. We assessed six existing AL algorithms and developed seven novel AL algorithms for the clinical NER task. In addition to the traditional assumption of same annotation cost per sentence, we also evaluated our methods based on the assumption of same annotation cost per word, which is closer to the real world scenario. The results of our study showed that multiple AL algorithms outperformed passive learning using both evaluations.

2.2 Methods

2.2.1 Dataset

In this study, we used the annotated training corpus from the 2010 i2b2/VA NLP challenge, which contains 349 clinical documents with 20,423 unique sentences. Three types of medical entities: problem, treatment, and test, were annotated in each sentence. Table 1 shows the descriptive statistics of the corpus. The dataset is divided into two pieces: 1) the pool of data to be queried and 2) the independent test set for evaluation. As we used 5-fold cross validation in the experiment, each pool contains 80% of the data randomly selected from the original dataset while the independent test set has the remaining 20% of the sentences.

Table 1. Distribution of words and different types of entities in the corpus of 20,423 unique sentences

	Overall count	Mean of count per sentence	SD of count per sentence
Word	225,670	11.05	9.73
Entity	26,206	1.28	1.65
Problem entity	11,192	0.55	1.03
Treatment entity	8,099	0.40	0.91
Test entity	6,915	0.34	1.02

2.2.2 Active learning experimental framework

In this study, we simulated the practical pool-based AL framework. Although all sentences in our corpus were pre-annotated, we did not utilize their labels unless the querying algorithms selected them. The following is the framework we used in the experiments:

(1) Initial model generation: At the beginning, a small number of samples are queried for annotation to build the initial model. Instead of using random selection, we picked the 8 longest sentences for their entity labels to train the model. The main reason for using this strategy is that selecting the longest sentences could most likely avoid a zero-entity scenario and generate a better initial model or a starting point in the learning curve. Please note that all different methods used the same set of initial samples to ensure a fair comparison.

(2) Querying: The unannotated sentences were then ranked based on the querying algorithm. Some algorithms require the updated CRF model for ranking (e.g. uncertainty sampling) while some do not (e.g. all diversity based algorithms). The top ranked sentences were selected for

annotation, and then added to the annotated set. In our experiment, the batch sizes (the size of top ranked sentences selected for annotation) of each iteration were 8, 16, 32, 64, 128, …, 2^(i+2), where i is the number of iterations. This is one of the standard ways to select the batch size for AL experiments and has been used in an AL challenge [71].

(3) Training: The CRF model was retrained on the updated annotated set.

(4) Iteration: Steps (2) and (3) were repeated until the stop criterion was met (e.g. the limit of annotation cost was reached.).

Multiple measurements were stored during the AL process for evaluation, such as model quality in F-measure, number of words in the annotated set, and number of entities in the annotated set.

As shown above, querying algorithms are critical for an AL system. The following sections discuss three types of querying algorithms that we investigated in our experiments. Some algorithms were developed by previous studies and some algorithms are newly developed in this study (marked as new).

2.2.2.1 Uncertainty-based querying algorithms

The assumption here is that the most uncertain sentences are most informative because identification of their uncertain labels could gain the most utility for the supervised NER learning. We considered a label of a sentence as a sequence of labels of words. In most of our implementations, only the N-best sequence labels were considered since the size of the possible sequence labels grows exponentially as the length of a sentence increases. We also extended the N-best sequence labels to cover most of the highly probable labels. The entropy of words and

entities was also tested in our study. The six methods we implemented to calculate the uncertainty of a sentence are described below:

(1) Least Confidence (*LC*): to take the uncertainty from the best possible sequence label based on the posterior probability output from CRF. The uncertainty of a sentence is equal to $1 - P(y^*|x)$, where y^* is the most likely sequence label.

(2) *Margin*: to take the uncertainty from the best two possible sequence labels. The uncertainty of a sentence is equal to $P(y^*|x) - P(y^{**}|x)$, where y^* and y^{**} are the most likely and second most likely sequence labels, respectively. The smaller difference between the two probabilities represents higher uncertainty.

(3) *N-best sequence entropy*: to take the entropy of the probability distribution over N-best sequence labels predicted by the CRF model. We used N=3 in our experiments.

(4) *Dynamic N-best sequence entropy* (new): to take the N-best sequence labels with the sum of their probabilities being at least 0.9. Here, N ranges from 1 to 20 in our experiments. For example, if the best sequence label has a probability of 0.95, N is equal to 1 (equivalent to LC); if the best 4 sequence labels have probabilities of 0.4, 0.3, 0.1, and 0.1, the sum of the probabilities is 0.9 and therefore, N is 4 and we ignore the 5th and later labels.

(5) *Word entropy*: to take the summation of entropy of individual words given the probability distribution over all possible labels.

(6) *Entity entropy* (new): to take the summation of entropy of the beginning word of the estimated entities (e.g. B-entity; excluding the entropy from the inside "I" and outside "O" of the estimated entities).

2.2.2.2 Diversity-based querying algorithms

Uncertainty sampling is highly dependent on the quality of the model. Therefore, it may not be efficient in a practical setting where updating the model may take time. In this section, we propose diversity-based querying algorithms that consider information other than the model, such as the similarity between sentences.

The idea behind the diversity-based querying algorithms is that we do not want to query the sentences that are similar to those that are already annotated. We applied the vector space model to pre-calculate pair-wise cosine similarity of any two sentences in the corpus. We used complete-linkage (max similarity) to determine the similarity between an unlabeled sentence and a group of labeled sentences. Unlabeled sentences with lower similarity scores would be assigned higher priority for annotation. The advantages of the diversity-based algorithms are (1) it is not dependent on the model and the annotation results; (2) the pair-wise similarity scores between sentences could be pre-computed, thus the querying step could be very efficient.

To find the best similarity measurements, we explored different features at the word, semantic, and syntactic levels for building vectors and calculating similarity scores. We also combined all of them for better similarity assessment.

(1) *Word similarity* (new): A vector of words weighted by the TF/IDF weighting scheme is used to represent each sentence. Then the cosine similarity between two vectors is calculated as the similarity between the two sentences.

(2) *Syntax similarity* (new): Each sentence is parsed by the Stanford parser [72] and the dependency relations derived from the parse tree are used to form the vector. For example, a sentence "She is afebrile with stable vital signs." has six dependencies "nsubj(afebrile-3, She-1)", "cop(afebrile-3, is-2)", "prep(afebrile-3, with-4)", "amod(signs-7, stable-5)", "amod(signs-7, vital-6)", and "pobj(with-4, signs-7)". To generalize the dependency relations, we then replaced the arguments of relations by their corresponding part of speech (POS) tags. The above example was converted into a vector of ["nsubj(JJ, PRP)", "cop(JJ, VBZ)", "prep(JJ, IN)", "amod(NNS, JJ)", "amod(NNS, JJ)", and "pobj(IN, NNS)"]. We weighted each dependency relation in the vector using the TF/IDF weight scheme based on their counts in the sentence and the corpus. Finally, cosine similarity was computed for each pair of sentences, similar to the method of word similarity.

(3) *Semantic similarity* (new): This method is to calculate semantic similarity between two sentences based on concept similarity. We modified an existing semantic similarity method originally based on word similarity [73]. Our approach consisted of two steps: 1) extraction of clinical concepts in each sentence so that each sentence can be represented using a vector of union concepts from the two sentences; and 2) calculation of the similarity between the two sentence vectors of concepts, by measuring similarity scores between any two concepts and computing the cosine similarity of two sentence vectors. For Step 1, we processed each sentence using KnowledgeMap Concept Identifier (KMCI) [74], a general clinical NLP system, that extracts clinical concepts defined in the UMLS. Each sentence was represented by a vector of UMLS concept unique identifiers (CUIs) of the union concepts. For Step 2, the semantic similarity (or distance) between any two UMLS concepts was calculated using the package of *UMLS-interface* and *UMLS-similarity* [75], which computes the similarity between two CUIs by

using the user-selected similarity measurement (i.e. *Path*, *LCH* [76], *WUP* [77], etc) with a specified source (i.e. SNOMED-CT [78] and MeSH [79]). The value of each union concept of a sentence is the max similarity among the similarity scores between each of the concepts from this sentence and this union concept. Once we formed the semantic vector for two sentences, we computed the cosine similarity between them. Figure 2 demonstrates an example of how *semantic similarity* is calculated for two sentences: S1: "You will need to have your uterine bleeding evaluated." and S2: "This continued agitation may be caused by intraparenchymal hemorrhage."

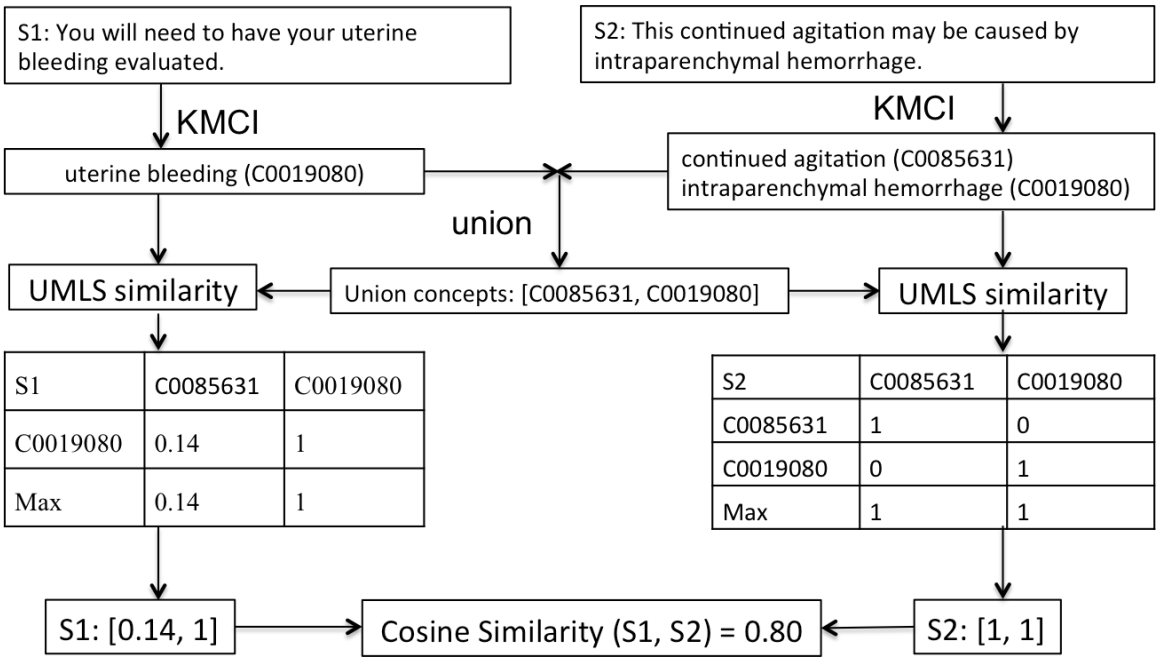

Figure 2. An example of computing similarity between two sentences using *semantic similarity* algorithm

KMCI identified "uterine bleeding" as an UMLS concept (with CUI: C0019080) in S1 and "continued agitation" (C0085631) and "intraparenchymal hemorrhage" (C0019080) in S2. The union UMLS concepts of the two sentences are C0085631 and C0019080. Then we applied

UMLS-similarity package to compute the similarity between the two concepts. The vector for S1 is [0.14, 1], where 0.14 is the UMLS similarity between C0019080 and C0085631 and 1 for the one between C0019080 and C0019080. The vector for S2 is [1, 1] because the UMLS concepts from S2 are exactly the same to the union concepts. Then the similarity between S1 and S2 is 0.80, which is the cosine similarity of these two vectors.

(4) *Combined similarity* (new): This approach combines all word, syntactic, and semantic information for similarity calculation. We first combined words and dependency relations for the same sentence into one vector, and then computed the cosine similarity for each pair of sentences based on the new vectors. The final combined similarity between the two sentences is the average similarity for both the newly computed cosine similarity between word/dependency vectors and the semantic similarity based on UMLS.

In principle, zero similarity score would indicate very diverse sentences, which we want to select. However, after careful analysis, we found that sentences with a zero similarity score to the labeled set were usually short sentences, which contain very few clinical entities. For example, short sentences such as section headers contain few dependency relations and often yield zero syntax similarity. Therefore, we decided to eliminate unlabeled sentences with zero similarity to the labeled set from sample selections for all diversity-based algorithms.

2.2.2.3 Baseline algorithms

In addition, we also included two querying algorithms that simply consider the length of the words or entities in a sentence. As we mentioned in the introduction, one limitation of such simulated AL studies is to assume that each sentence costs the same amount of annotation effort.

By including these two extremely biased methods as additional baselines, we hope to further confirm the effectiveness of AL methods.

(1) *Length-words* is a simple querying method that selects sentences with the largest number of words. The assumption is simply that longer sentences may contain more information for NER than shorter ones.

(2) *Length - concepts* is another simple querying method that selects sentences with the largest number of clinical concepts, as identified by KMCI. The assumption is that sentences with more clinical concepts are more informative sentences for NER.

In addition, we included the typical passive learning method *Random*, which randomly selects samples at each iteration.

2.2.3 Evaluation

Most of the AL studies utilized learning curves that plot F-measure of the model on an independent test set as a function of sample size of the training set as the primary evaluation approach. Following previous studies on open domain NER [66, 67], we first evaluated our AL-enabled clinical NER using the same type of learning curve that plots F-measure versus number of annotated sentences, assuming annotation cost is same for each sentence. However, we think the annotation costs for different sentences could differ greatly in reality; thus simply assuming the equal annotation cost of each sentence, as is traditionally done, could induce an inaccurate estimation about the benefit of AL in reality. Therefore, we also generated the learning curve of F-measure versus number of words in the annotated sentences as a new assessment approach. The new method of word-based evaluation assumes that the annotation cost is proportional to the length of a sentence, and is therefore a better way to estimate the real annotation cost. We also

computed the area under the learning curve (ALC) as a global score for both evaluation methods, which was a major metric to evaluate AL methods in the challenge [71]. The ALC scores for the learning curves of F-measure vs. sentences and F-measure vs. words are labeled as ALC1 and ALC2, respectively. To further demonstrate some characteristics of different querying methods, we plotted additional curves, including the *entity count curve* that plots the number of entities versus the number of sentences and the *sentence length curve* that plots the number of words (length of sentences) versus the number of annotated sentences.

Our evaluation results were based on 5-fold cross validation (CV). For each iterative experiment, one fold was used as an independent test set and four other folds were used as the pool of querying and training set. The results on the learning curves were averaged over the five runs. For the experiments using random sampling, we repeated the experiments of 5-fold CV five times and averaged their results.

2.3 Results

All methods were tested in the same AL framework and cross validation setting (e.g. the same initial queries and model, pool, batch size, parameters of CRF model, and test set). Table 2 shows the ALC scores based on two types of learning curves for twelve AL algorithms in three categories and *Random* that represents passive learning.

Table 2. Two types of ALC scores for all AL algorithms versus passive learning

Categories	Methods	Existing or New	ALC1 score	ALC2 score
			F-measure vs. Sentences	F-measure vs. Words
Uncertainty based sampling methods	*LC*	Existing	0.83	0.84
	Margin	Existing	0.83	0.84
	N-best sequence entropy	Existing	0.81	0.85
	Dynamic N-best sequence entropy	New	0.82	0.84
	Word entropy	Existing	0.83	0.84
	Entity entropy	New	0.83	0.84
Diversity based sampling methods	*Word similarity*	New	0.77	0.82
	Syntax similarity	New	0.72	0.80
	Semantic similarity	New	0.79	0.83
	Combined similarity	New	0.76	0.82
Baseline methods	*Length – Words*	Existing	0.82	0.81
	Length – Concepts	New	0.82	0.81
Passive Learning	*Random*	Existing	0.74	0.82

Note: ALC1 is the ALC (area under the learning curve) score for the learning curves of F-measure vs. number of sentences; ALC2 is the ALC score for the learning curves of F-measure vs. number of words.

For ALC1 that is based on learning curves of F-measure vs. number of sentences, all AL algorithms, except *syntax similarity*, were better than random sampling. Among the three types of algorithms, uncertainty-based sampling methods (0.83 in average ALC1) outperformed two

baselines (0.82 in average ALC1), which outperformed diversity-based methods (0.76 in average ALC1).

For ALC2 that is based on learning curves of F-measure vs. number of words, three types of querying algorithms performed differently: all six uncertainty-based methods outperformed random sampling; in the diversity sampling category, only *semantic similarity* achieved better performance than *Random*; ALC2 of baseline methods (*Length – Words* and *Length - Concepts*) did not exceed random sampling because the tendency of selecting longest sentences was penalized in this evaluation.

We generated two types of learning curves for all thirteen methods. However, for ease of interpretation, we selected the best-performing method in each category to display its learning curve versus *Random*. Figure 3 shows the traditional learning curves based on F-measure versus number of annotated sentences for methods of *LC*, *semantic similarity*, *Length-concepts*, and *Random*. The method of *Length-concepts* had the best performance at the very early stage, but was surpassed by *LC* at the later stages, which outperformed the other methods. Figure 4 shows the new type of learning curves based on F-measure versus number of words in the annotated sentences for the methods of *N-best sequence entropy*, *semantic similarity*, *Length-concepts*, and *Random*. *N-best sequence entropy* led all the stages of AL.

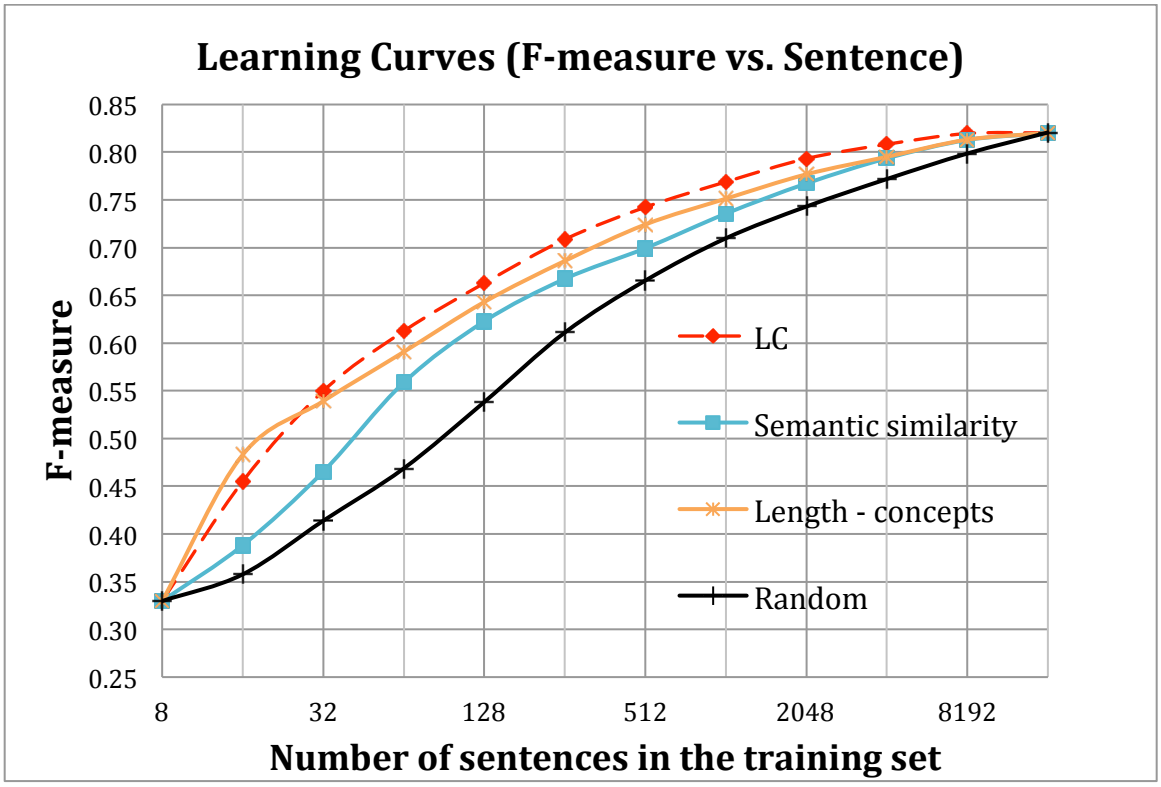

Figure 3. Learning curves for F-measure versus number of sentences in the training set

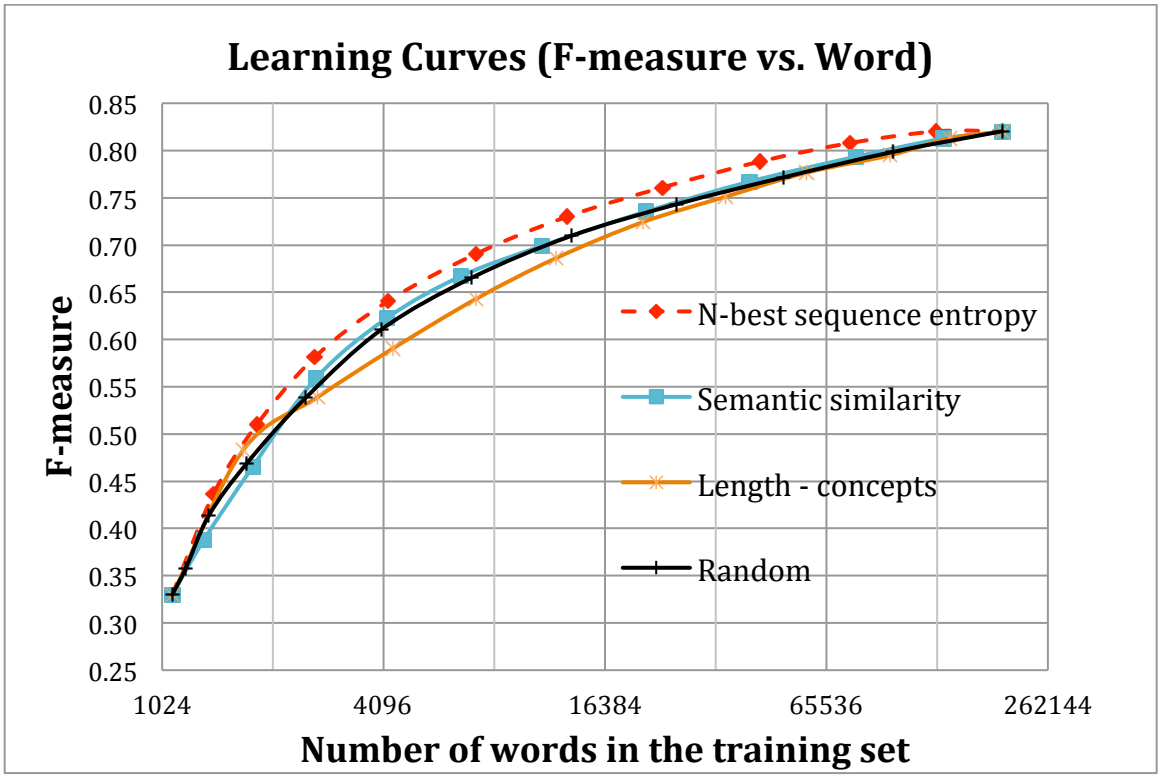

Figure 4. Learning curves for F-measure versus number of words in the training set

Based on learning curves in Figure 3 and 4, we also computed the number of annotated sentences and words required to achieve a fixed F-measure for each method. We used linear interpolation to estimate the points from learning curves that were not actually available. Furthermore, we estimated the extent of annotation cost saving achieved by AL as compared to passive learning for achieving the same performance. For example, to achieve 0.80 in F-measure, *LC* used 2,971 sentences or 61,238 words, *N-best sequence entropy* required 3,249 sentences or 62,486 words, *semantic similarity* needed 5,468 sentences or 98,075 words, *Length-concepts* required 5,201 sentences or 109,580 words, and *Random* queried 8,702 sentences or 105,340 words. Compared to *Random* with respect to cost saving in sentences, *LC* saved 5,731 sentences (66%), *semantic similarity* saved 3,234 sentences (37%), and *Length-concepts* saved 3,501 sentences (40%). With

respect to cost saving in words, *LC* reduced 42,854 words (41% saving of annotation cost in words), *N-best sequence entropy* could save more - 44,102 words (42%). However, *semantic similarity* saved only 7,265 words (7%), and *Length-concepts* actually required annotating 4,240 additional words (4% increase of annotation cost in words).

Figure 5 shows the entity count curves for *Random* and other methods (*LC*, *semantic similarity*, *Length-concepts*) that achieved the highest entity count per sentence in their categories. Figure 6 shows the sentence length curves for *Random* and other methods (*LC*, *semantic similarity*, *Length-words*) that queried the longest sentences in their categories.

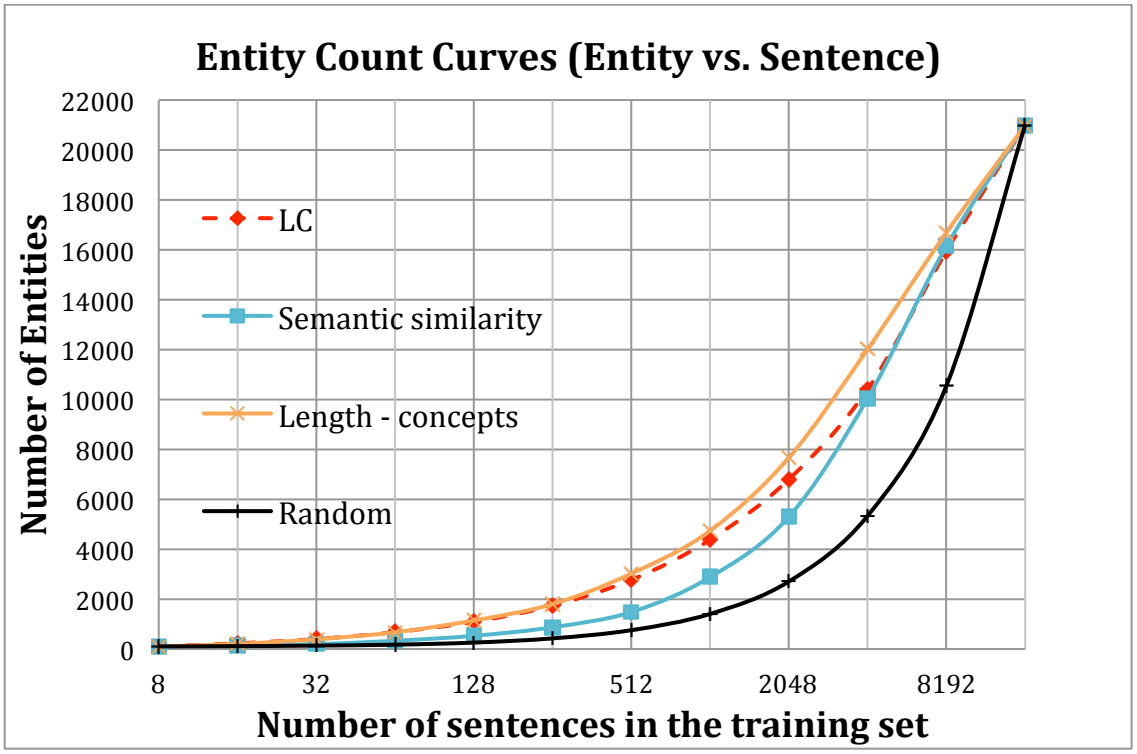

Figure 5. Entity count curves that plot number of entities versus number of sentences in the training set

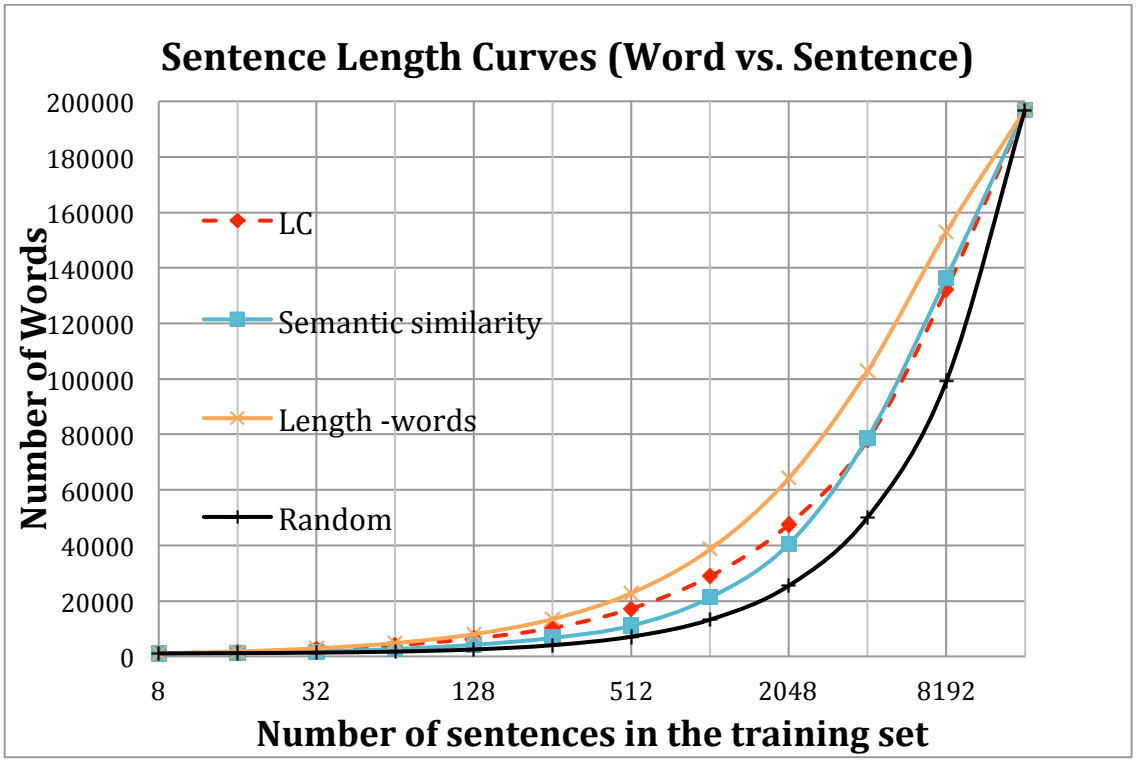

Figure 6. Sentence length curves that plot number of words versus number of sentences in the training set

Both *entity count curves* and *sentence length curves* present a very similar pattern. *Length-concepts* and *Length-words* queried the most number of concepts per sentence and the longest sentences, respectively, at all stage of AL, while random sampling did the least. Both *LC* and *semantic similarity* fall between the curves mentioned above, but they both performed better than *Length-concepts*, *Length-words*, and *Random* in terms of ALC2.

2.4 Discussion

In this study, we conducted simulated AL experiments for a clinical NER task and demonstrated that AL has the potential to reduce annotation cost for building clinical NER models. To the best of our knowledge, this is the one of the earliest studies on AL for clinical NER. According to

Figure 3 to 6, AL algorithms (e.g. uncertainty sampling) did query longer sentences with higher number of entities per sentence, which could contribute to higher ALC1 and ALC2 scores. However, simply selecting sentences with high number of entities (e.g. *Length-concepts*) or longest sentences (e.g. *Length-words*) failed to surpass passive learning in ALC2 score, which we consider as an evaluation metric closer to the real-time situation. This finding suggests that active leaning does select informative samples that can help build better clinical NER models quickly.

Uncertainty-sampling based algorithms outperformed all other methods in both ALC1 and ALC2, because they queried the most informative sentences using the knowledge of trained models. Among these methods, *LC*, *Margin*, *Word Entropy*, and *Entity Entropy* had very similar results (0.83 in ALC1 and 0.84 in ALC2). *N-best sequence entropy* gained highest ALC2 (0.85), indicating that it is probably more efficient in reality. However, one concern of applying uncertainty sampling based methods to real-world annotation tasks is that they rely on the updated NER models, which may take time when the annotated data set is getting bigger. For example, it would take several minutes to fully train a model based on 1000 annotated sentences in our experiment. In reality, it may not be feasible to ask annotators to wait such a long time for the next iteration of queried samples.

The diversity sampling methods, on the other hand, do not depend on the CRF model and most processes can be pre-computed before the annotation process starts, which makes it more appealing. However, the current diversity-based methods implemented in this study did not perform as well as the uncertainty sampling. One possibility to improve the diversity-based methods is to integrate clustering algorithms (e.g. *k-means* and *affinity propagation* [80]) to find the most representative samples. In addition, we could also investigate better feature

representation methods such as the topic modeling method (e.g. *Latent dirichlet allocation* [81]). Another research direction is to combine uncertainty and diversity methods, e.g., using the linear function from Kim et al. [66].

Another contribution of this work is to introduce a new evaluation metric for simulated AL studies for NER. Instead of assuming that each sentence requires the same amount of annotation effort, we assume each word requires the same amount of annotation effort. Therefore, the estimated savings of annotation cost in our study would be closer to reality, where longer sentences probably need more annotation time than the shorter sentences. Our results seem to support this intuition. For example, to achieve an F-measure of 80%, the *LC* method could save 66% sentences; but the saving would be only 42% if we consider words instead of sentences. The 24% drop of savings indicates that the traditional evaluation could overestimate the effectiveness of AL methods in NER, when compared to passive learning. Moreover, other AL methods such as the diversity sampling methods, which could outperform passive learning in ALC1, did not achieve the same performance when ALC2 was used in evaluation. For example, the *semantic similarity* method showed a saving of 37% in ALC1 evaluation; but it had a saving of only 7% in ALC2 evaluation. These findings suggest that we should be more cautious about results from simulated experiments of AL on clinical NER. The actual benefit of AL should be further evaluated using real-time settings of NER tasks.

As described above, the main limitation of this study is that it is a simulated study of AL for clinical NER. To assess the real value of AL for clinical NLP, we will have to evaluate it in a real-world setting. There are a few machine learning systems with integrated AL components, such as the DUALIST system [59] for word sense disambiguation in open domains. However, to our knowledge, there is no clinical NLP system that integrates a practical AL module. Therefore,

our next step is to develop a clinical NER system, which consists of an annotation interface and an AL component that actively selects samples for annotation. We will then conduct formal user studies to compare AL vs. passive learning in terms of annotation time and model quality.

2.5 Conclusion

We conducted a simulated study to compare different AL algorithms for a clinical NER task. Our results showed that most AL algorithms outperformed the passive learning method when we assume equal annotation cost for each sentence. However, savings of annotation by AL were reduced when the length of sentences was considered. We suggest that the effectiveness of AL for clinical NER needs to be further evaluated by developing AL enabled annotation systems and conducting user studies.

CHAPTER 3

An Active Learning-enabled Annotation System for Building Clinical Named Entity Recognition Models

3.1 Introduction

Most of AL studies for biomedical text processing, including our study in Chapter 2, were conducted in a simulated setting, which assumes that annotation cost for each sample is identical. In reality, however, annotation cost (i.e. the time required to annotate one sample by an annotator) can be very different from one sample to another, or from one annotator to another. The estimated cost savings by AL in simulated studies may not be applicable in reality. Settles et al. [60] conducted a detailed empirical study to assess the benefit of AL in terms of real-world annotation costs and their analysis concludes that a reduction in the number of annotated sentences required does not guarantee a real reduction in cost. Therefore, to better understand how AL works within the real time annotation process and to demonstrate the utility of AL in real-world tasks, we should integrate AL technologies with annotation systems and validate its effectiveness by recruiting users to conduct real-world annotation tasks.

In this study, we aimed to evaluate performance of AL versus passive learning in annotating problems, treatments, and lab tests in clinical notes to build ML-based NER systems in real-time. Our work consists of two main parts: 1) develop an AL enabled annotation system, called *Active LEARNER* (or *A-LEARNER*), for clinical NER; and 2) conduct a user study to evaluate the performance of *Active LEARNER* in practice.

The front end of *Active LEARNER* is a graphic user inference that allows users to mark clinical entities in a sentence supplied by the system using a particular querying engine. In the back end, the system iteratively trains CRF models based on users' annotations and selects the most useful sentences based on the querying engine. The system implements a multi-thread processing scheme to allow a no-waiting annotation experience for users. Meanwhile, we proposed novel AL algorithms, which query sentences that are not only the most uncertain samples but also most representative of the corpus based on sentence clusters. The algorithm is called Clustering And Uncertainty Sampling Engine (*CAUSE*), which served as the querying engine in the *Active LEARNER*.

In the user study, we compared the performance of *CAUSE* against *RANDOM* (random sampling), which represents passive learning. Two nurses were recruited to use *Active LEARNER* to annotate sentences and build NER models for both *CAUSE* and *RANDOM*, with a rest period between sessions. To ensure that the results from the two methods are comparable, we provided intensive user training, which includes review of annotation guidelines, review of sentence-by-sentence annotations, and multiple rounds of practice sessions. Once users completed the training and achieved consistent annotation performance, they were asked to annotate sentences for a fixed time period that consists of four 30-minute sessions for each method. We evaluated the performance of each method by generating learning curves (i.e. F-measure of the NER model on the test dataset vs. annotation time) for each user.

Our results show that *CAUSE* did not guarantee less annotation time than *Random* across different users, at a given performance point of the model (e.g., F-measure of 0.7). We then discuss other findings in our experiments and the limitations of the *CAUSE* method, with suggestions to future improvements, which are partially addressed in the Chapter 4.

3.2 Methods

The clinical NER task in this study was same as the previous one in Chapter 2, which is to extract problem, treatment, and lab test concepts from clinical notes. We first developed an AL-enabled annotation system, which iteratively builds the NER model based on already annotated sentences and selects the next sentence for annotation. Multiple new querying algorithms were developed and evaluated using the simulated studies. For the user study, the best querying algorithm from the simulation was implemented in the system. Two nurses were then recruited and participated in the real-time annotation experiments using the system for both *CAUSE* and *RANDOM* modes.

3.2.1 Development of the active learning-enabled annotation system

Practical AL systems such as DUALIST [59] have been developed to allow user and computer to iteratively interact for building supervised ML models for different NLP tasks, such as text classification and word sense disambiguation. For sequence labeling tasks such as NER, however, there is no existing interactive system available. In this study, we designed and built a system named *Active LEARNER* (also called *A-LEARNER*), which stands for Active Leaning Enabled AnnotatoR for Named Entity Recognition. To the best of our knowledge, it is the first AL enabled annotation system for clinical NER tasks.

Active LEARNER uses the unlabeled corpus as the input and generates NER models on the fly, while iteratively interacting with the user who annotates sentences queried from the corpus. The *Active LEARNER* system consists of three components: 1) the annotation interface, 2) the ML-based NER module, and 3) the AL component for querying samples. For the annotation

interface, we adopted the existing *BRAT* system, a rapid annotation tool developed by Stenetorp P et al. [82]. We modified the original *BRAT* interface to allow users to mark entities more efficiently. The ML-based NER module was based on the *CRF* algorithm implemented by *CRF++* [83], as described in [38]. The AL component implemented some existing and novel querying algorithms (described in later sections) using a multi-thread framework. More details of the *Active LEARNER* system are described below.

3.2.1.1 System workflow

Initial design: The original plan was to follow the traditional pool-based AL framework [45]. Figure 7 shows the initial design workflow of the *Active LEARNER* system. Once the system starts, the pool of unlabeled data is loaded into the memory. At the initial iteration or before the CRF model is generated, all sentences are randomly ranked. The top sentence in the ranked unlabeled set is queried and displayed on the interface. The annotator then highlights clinical entities in the sentence via the labeling function on the interface. When the user submits the annotated sentence, the *labeled set* and the *unlabeled set* are updated and the *learning process* is activated. Specifically, the *learning process* includes CRF model encoding based on the current labeled set and sentence ranking by the querying engine. The CRF model encoding is straightforward; however, it could take time to rebuild the CRF model when the labeled data set gets bigger. Sentence ranking consists of two steps: 1) CRF model decoding, which is to make predictions for each unlabeled sentence based on the current model; and 2) ranking sentences by the querying algorithm, which considers both the probabilistic prediction of each sentence from step 1, and other information about the unlabeled sentences (i.e. clustering results). The *learning*

process is complete when the ranked unlabeled set is updated. The next iteration starts when the annotator starts reading the new top unlabeled sentence on the interface. The program is stopped when the user either clicks the quit button or a pre-set cutoff time runs out. The drawback of the initial design, however, is that the annotator sometimes has to wait for the next sentence, because the learning process could take time, as CRF model encoding/decoding could be slow with large samples.

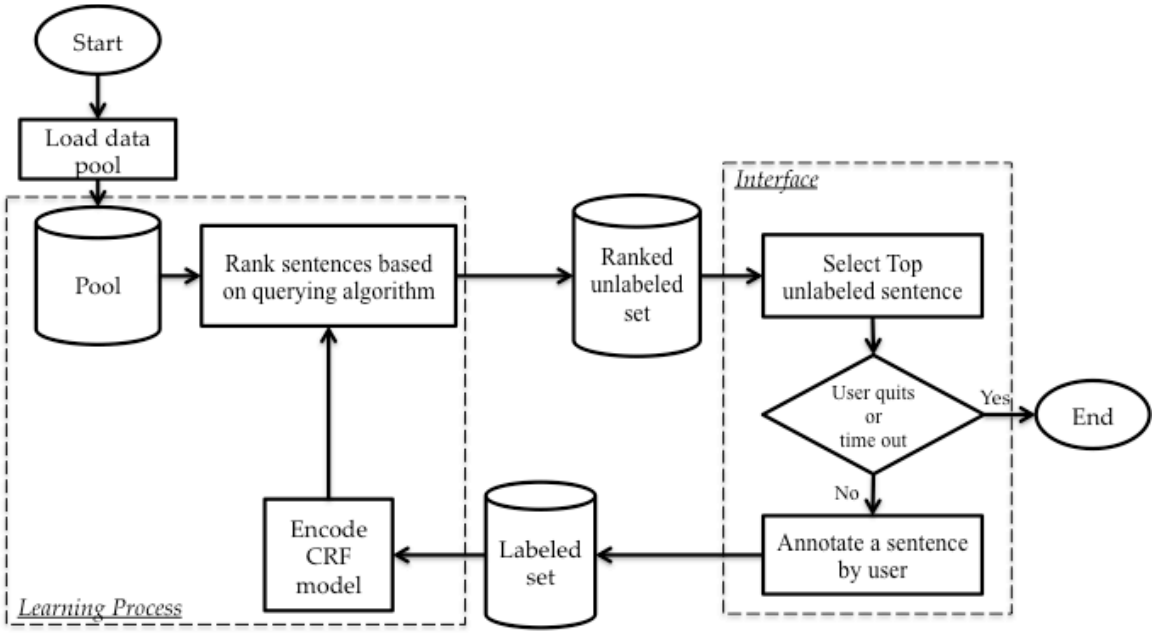

Figure 7. Workflow of Active LEARNER - initial design

Final design: To avoid delay in the workflow, we separate the annotation and learning processes by paralleling two threads: the annotation thread and the learning thread. The final design workflow is shown in Figure 8.

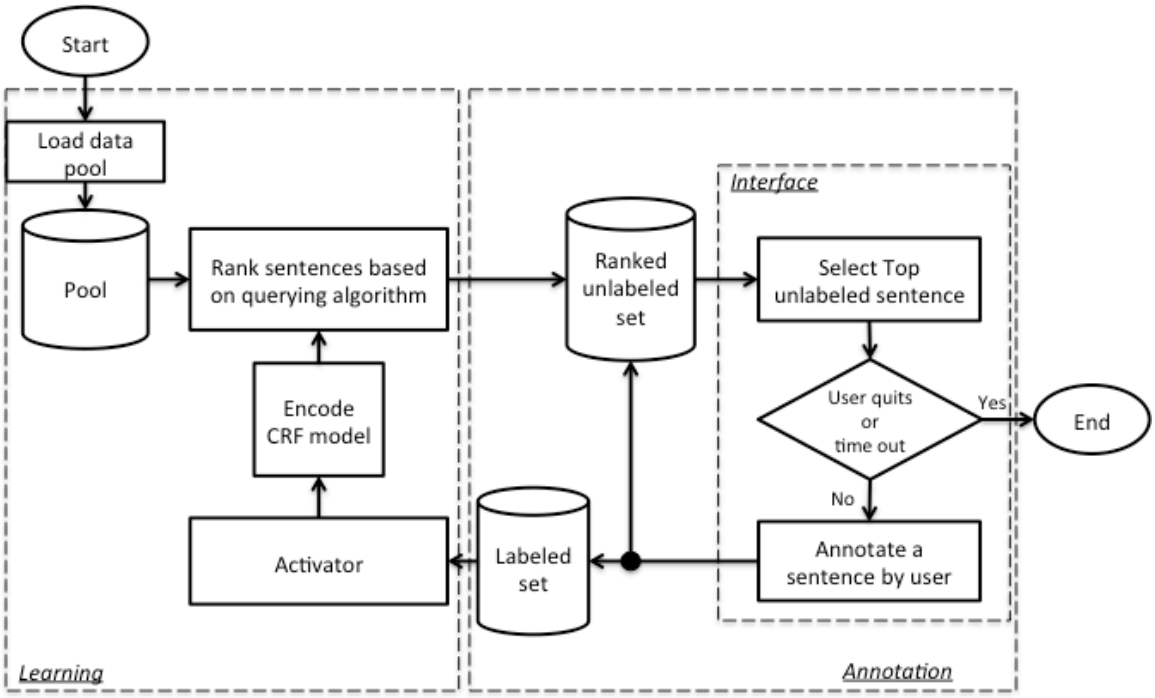

Figure 8. Workflow of Active LEARNER - final design

In the annotation thread, the black circle in the figure splits the flow into two, which run simultaneously. One sub-flow runs back to the *ranked unlabeled set* and interface. Therefore, the user can immediately read the next sentence on the interface right after the annotation of the previous sentence is submitted. The other sub-flow adds the newly annotated sentence to the labeled dataset and pushes the newly updated labeled set to the learning thread. In the learning thread, the process starts from an activator. A new *learning process* will be activated if the encoding or querying process in the learning thread is not busy and the number of newly annotated sentences is greater or equal to a threshold (equal to five in our study), which is for the update frequency control. When the *learning process* is activated, it runs in parallel with the annotation thread and it updates the *ranked unlabeled set* whenever the new rankings are generated. This design allows a user to continuously annotate the top unlabeled sentence from

the ranked list, which is generated by either the current or previous learning process in the learning thread. The stop criteria are the same as those described in the initial design.

To better record and manage the user study, we also integrated additional functions to the Active LEARNER system:

Log Function: We collect and record various types of information during annotation, including 1) user annotation activities such as marking, changing, or deleting entities; 2) detailed time information such the start and end annotation time stamp for every sentence; and 3) model performance information, such as intermediate NER models files and querying score for each unlabeled sentence for each update, so that we can report the precision, recall, and F-measure of models over time. All of the logging information is analyzed after the annotation task is completed, to provide additional insights to the annotation and learning processes.

Session Manager: we divide the entire annotation task into different sessions so that users can take a break between sessions. The time for each session can be pre-set on the interface for 15, 30, 45, or 60 minutes. When the user clicks the start button, Session #1 starts and the timer activates. When the time of the session is up, a pop up window will interrupt the annotation and remind the user to take a break. After the break, the user can click the "Resume" button to continue the annotation for the next session (e.g. session #2, session #3, and so on). The system also automatically saves everything so that the annotation task can be resumed in case it is paused in the middle of a session.

Prerequisites for running the Active LEANER include: (1) Corpus should be pre-processed for tokenization and sentence separation; (2) Features for CRF encoding and decoding should be

pre-extracted for every sentence; (3) The entities of interest need to be pre-defined (e.g. problem, treatment, and test).

3.2.1.2 Querying methods (clustering and uncertainty sampling engine)

In Chapter 2, we have described multiple AL querying algorithms and shown that uncertainty based sampling methods are more promising than other methods to reduce the annotation cost (in terms of the number of sentences or words) in the simulated studies. In this study, we further developed a novel AL algorithm that considers not only the uncertainty but also the representativeness of sentences. The AL methods were compared to a passive learning method based on random sampling (*Random*) in both the simulation and the user studies.

Uncertainty sampling (*Uncertainty*) assumes the most informative sentences are the ones with the highest uncertainty to be labeled by the model. We implemented Least Confidence (*LC*), which takes the uncertainty from the best possible sequence label based on the posterior probability output from CRF, in the annotation system. *LC* is the least computationally expensive and therefore the fastest among the six uncertainty sampling methods presented in Chapter 2. Moreover, *LC* achieved the best ALC score based on equal cost per sentence assumption (ALC1 in Chapter 2) and the second best ALC score based on equal cost per word assumption (ALC2 in Chapter 2).

Uncertainty based sampling methods are promising for selecting the most informative sentences from the pool for the clinical NER modeling. However, these methods could not distinguish the most representative sentences with respect to their similarity. As similar sentences could share very close uncertainty scores, the batch of the top ranked sentences could possibly contain

multiple similar sentences with repeated clinical concepts. These concepts may be annotated more than once in these similar sentences. Obviously, annotating such similar sentences is not the most efficient for building NER models although these sentences are most informative.

Table 3 shows a scenario where two sentences are most informative but occurred back-to-back when we ran *Active LEARNER* with *LC* as the querying engine. After the user submitted the annotation of the first sentence, the second sentence was shown. Then the user had to mark many repeated entities in the second sentence. The duplicated effort on the second sentence did not really improve the NER model and therefore decreased the AL efficiency.

Table 3. A scenario of two most informative sentences that occurred back-to-back when *Active LEARNER* was tested with *LC* as the querying engine

Sentence 1	Coronary Artery Disease, Hypertension, Hyperlipidemia, Diabetes Mellitus, Hypothyroid, h/o Bilateral DVT's (on chronic coumadin therapy), Pleural disorder? Sarcoidosis, Gastritis, B12 deficiency, Chronic renal insufficiency, s/p Appendectomy, s/p Lap cholectomy, s/p Total abdominal hysterectomy
Sentence 2	PMH: Hypertension, Hyperlipidemia, Diabetes Mellitus, Hypothyroid, h/o Bilateral DVT's, Pleural disorder? Sarcoidosis, Gastritis, B12 deficiency, Chronic renal insufficiency, s/p Appendectomy, s/p Lap cholectomy, s/p Total abdominal hysterectomy

Note: The duplicated words from two sentences were underlined

Here, we propose the clustering and uncertainty sampling engine (*CAUSE*) that combines clustering technique and uncertainty sampling to query both informative and representative sentences. This method guarantees that the top ranked sentences in a batch are from different clusters and thus dissimilar with each other.

The algorithm of *CAUSE* is described as the following:

Input:

(1) Clustering results of sentences

(2) Uncertainty scores of sentences

(3) Batch size (x)

Steps:

(1) Cluster ranking: score each cluster based on the uncertainty scores of sentences and select the top x cluster(s) based on the cluster scores, where x is the batch size; (e.g. the score of a cluster could be the average uncertainty score of sentences in this cluster.)

(2) Representative sampling: in each selected cluster, find a sentence with the highest uncertainty score as the cluster representative.

Output: x cluster representative sentences in the order of their cluster ranking.

Initial sampling: When the NER model and uncertainty scores of sentences are not available, we used random sampling to select a cluster and the representative within the selected cluster.

The following sections describe how exactly the *CAUSE* algorithm was implemented in this study.

3.2.1.2.1 Sentence clustering with topic modeling

Clustering is a required pre-processing step in *CAUSE* for the pool of data to be queried. The clustering process consists of Latent Dirichlet Allocation (*LDA*) [81], a topic modeling technique, for feature generation, and affinity propagation (*AP*) [80] for clustering. In this clinical concept extraction task, we need to group semantically similar sentences together. We applied a C implementation of *LDA* (*LDA-C*) [13] to extract the hidden semantic topics in the corpus of clinical notes. Since using document-level samples for topic modeling could generate more meaningful topics than sentences, we ran *LDA* topic estimation on the entire dataset from the 2010 i2b2/VA NLP challenge (826 clinical documents). Given the K estimated topics, the *LDA* inference process was performed to assign probabilistic values of topics for every sentence. Eventually, each sentence was coded in a K dimensional vector with a probability at each of the K topics as value. Cosine similarity was used to calculate the similarity between every sentence pair. Next, we applied a python package of *AP* [84] that takes the M x M pair-wise similarity matrix as the input and outputs the clustering result for the M sentences.

3.2.1.2.2 Cluster ranking

Each cluster is assigned a score based on one of the following schemas: (a) Maximum uncertainty cluster sampling (MUCS): assign the cluster the highest uncertainty score among all the sentences in the cluster; (b) Average uncertainty cluster sampling (AUCS): assign the cluster the average uncertainty score from all the sentences in the cluster; (c) Random cluster sampling (RCS): assign the cluster a random score (assuming that each cluster is equally important). According to our experiments, AUCS performed the best in terms of learning curve

performance. The cluster with a higher score will be ranked higher among all clusters, thought to contribute most to the NER modeling.

3.2.1.2.3 Representative sampling

From the top ranked cluster, we select the sentence that has the highest uncertainty score as the representative of the cluster. We also find the representative sentences from the second ranked cluster, third ranked cluster, and so on. We keep sampling until the batch is filled up with representatives. The ranking of the representatives follows the ranking of their clusters. We assume that the number of clusters is greater than or equal to the batch size so that the batch cannot contain more than one sentence from a cluster.

The assumption here is that cluster representative sentences can improve the NER model by helping identify entities from other sentences in the same cluster. Table 4 shows an example of a cluster that contains multiple sentences about medications. The cluster representative is the first sentence, where “Dulcolax” is tagged as the medication treatment. When the NER model is trained on the annotated cluster representative, the model could identify other medications (e.g. “Amaryl”, “Nortriptyline”, “Metformin”, etc.) from additional sentences in the same cluster based on their similar context (e.g. “mg”, “p.o.”, and “q.”) as the cluster representative.

Table 4. An example of a cluster that contains multiple sentences about prescription

Cluster representative	Sentences in a cluster
X	14. ***Dulcolax*** 10 mg p.o. or p.r. q. day p.r.n.
	9. Amaryl 4 mg p.o. q. day .
	3. Nortriptyline 25 mg p.o. q. h.s.
	2) Metformin 500 mg p.o. q. 8 hours .
	…

3.2.2 The user study

The user study is to evaluate the performance of AL versus passive learning in the real-world annotation processes for building NER systems. The annotation cost in the user study is not number of sentences or words but the actual annotation time by an annotator; the annotations (i.e. clinical entities) are done by users on-the-fly, instead of from a pre-annotated gold standard. Two nurses are recruited to use Active LEARNER to annotate sentences and tested both *CAUSE* and *RANDOM* modes.

3.2.2.1 Study design

3.2.2.1.1 Training annotators

We understand that there are human factors influencing the user study, such as annotation speed and annotation quality, in addition to querying methods. To make the results of two methods

comparable, we rigorously trained two users in the annotation process, to ensure they will perform consistently in both experiments. The user-training phase included the following steps:

Guided training: The first step of training is to study the annotation guidelines, which were generated by the 2010 i2b2/VA NLP challenge organizer [85]. Both nurses had some experience on similar chart review tasks. At the very first training session, the NLP expert discussed the annotation guidelines with two nurses for 15-30 minutes, particularly focusing on the annotation boundaries of the clinical concepts. The next step was to review annotations sentence-by-sentence. The objective of this training session was to train users to be more familiar with both the annotation guidelines of the task and the *Active LEARNER* interface. Users were shown two interfaces on the left and right half of the screen. A user annotates a sentence on the left-side interface. When the annotation is finished, the user could review the i2b2 gold standard of the annotation for the same sentence on the right-side interface. If there was discrepancy between the user's annotation and the gold standard, we discussed the possible reasons that support either gold standard or user annotation. A user could either stick to the original decision or change the annotation based on the discussion.

Practice: The practice process consists of two parts: 1) a shorter session with two to three 15-minute of annotation; and 2) a longer session with four half-hour annotation, which was the same as the main user study discussed in the later section. The users conducted this part of training independently without breaks. We collected user's annotation speed and annotation quality at each session so that we could track if the user achieved consistent annotation performance.

3.2.2.1.2 The main study design

Warm up training section: In the second and third week of the user study, we conducted a shorter version of the training called warm up training. This served to refresh users on both annotation guidelines and interface usage. We also measured the speed and the quality of the annotation with user's current status. The warm up training also consisted of two parts. The first part was sentence-by-sentence annotation review. It took at least 15 minutes and up to 45 minutes. This part could be stopped when user was making annotations consistent with the i2b2 gold standard. The second part was two 15-minute sessions of annotation. We used this opportunity to measure the user's current speed and quality of annotation.

After users were well trained on the annotation task, we asked users to start the real experiments. As shown in Table 5, both users tested the *Random* method in week 1 and then the *CAUSE* method in week 2. The reason to separate the user studies for two methods by a one-week gap is to allow users to forget the previous annotation. In each week, a user was required to go through a warm up training first, and then to complete the annotations of four half-hour sessions. The annotation time for each session was set to 30 minutes. A break of at least 10 minutes and up to 15 minutes was required between two sessions. During one session, each user was asked to continuously work without break. With respect to the user study environment, each user was isolated in a conference room with minimum interruption.

Table 5. Schedule of the user study

Time	Event	Task	Duration
Week 0	Guided Training	1. Annotation guideline review	30 minutes
		2. Sentence-by-sentence annotation and review using the interface	45 minutes
	Practice	1. Three quarter-hour sessions of annotation practice	45 minutes
		2. Four half-hour sections of annotation using *Random*, with 15-minute break between sessions	3 hours
Week 1	Annotation warm up training	1. Sentence-by-sentence annotation and review using the interface	15 - 30 minutes
		2. Two 15 min sessions of annotation practice	30 minutes
	Main study for method *Random*	Four 30 min sessions of annotation using Method 2	3 hours
		15-minute break between sessions	
Week 2	Annotation warm up training	1. Sentence-by-sentence annotation and review using the interface	15 - 30 minutes
		2. Two 15 min sessions of annotation practice	30 minutes
	Main study for method *CAUSE*	Four 30 min sessions of annotation using Method 2	3 hours
		15-minute break between sessions	

3.2.2.2 Datasets

In this chapter, we used the same annotated training corpus from the 2010 i2b2/VA NLP challenge as described in Chapter 2. The clinical named entity recognition task is to identify the medical concepts of problem, treatment, and lab test from the corpus. The dataset with 20,423 unique sentences was randomly split into five folds, each of which has either 4,084 or 4,085

unique sentences. In the simulation, we performed 5-fold cross validation so that four out of five folds were used as the pool of data to be queried and the remaining fold was the independent test set for evaluation. In the user study, we used fold 1 with 4,085 unique sentences as the independent test set and the remaining 16,338 unique sentences as the pool for data querying. In the annotation warm up training, the reviewed sentences are from the independent test set. Table 6 shows the characteristics (counts of sentences, words, and entities, words per sentence, entities per sentence, and entity density) in five folds of the dataset and the pool of querying data.

Table 6. Characteristics (counts of sentences, words, and entities, words per sentence, entities per sentence, and entity density) in five folds of the dataset and the pool of querying data

	Sentence count	Word count	Entity Count	Words per sentence	Entities per sentence	Entity density*
Fold 1	4,085	44,403	5,395	10.87	1.32	0.25
Fold 2	4,085	45,588	5,183	11.16	1.27	0.24
Fold 3	4,084	45,355	5,201	11.11	1.27	0.24
Fold 4	4,085	45,141	5,263	11.05	1.29	0.25
Fold 5	4,084	44,834	5,177	10.98	1.27	0.24
Pool (Fold 2+3+4+5)	16,338	180,918	20,824	11.07	1.27	0.24

Note: Entity density is the number of words of the entities divided by the total number of words.

3.2.2.3 Evaluation

In the simulation study, we used number of words in the annotated sentences as the estimated annotation cost. The learning curves that plot F-measures vs. number of words in the training set

were generated to visualize the performance of different methods. For each method, the five learning curves from the 5-fold cross validation were averaged to general a final learning curve.

In the user study, actual annotation time was used as the annotation cost. We also generated the learning curves that plot F-measure vs. actual annotation time to compare both AL and passive learning. Moreover, there are human factors that would affect the learning curve as well, such as user *annotation speed* and *annotation quality*. The most intuitive annotation evaluation metric to determine the *annotation speed* is the entity tagging speed (e.g. number of entities or entity annotations per minute). Obviously, if a user can contribute significantly more annotations in a given time, the learning curve of NER models could be better regardless of querying methods. In addition, the *annotation quality*, which is measured by F-measure based on gold standard, is another important factor for training a clinical NER model. If we fix the *annotation speed*, higher *annotation quality* would help build better NER models. Therefore, we also evaluated users' annotation by generating the annotation speed curve (annotated entities per annotation time) and annotation quality curve (F-measure per section). Additional curves were also generated to illustrate some characteristics of methods over annotation time, including: sentence count, average sentence length, and annotated word count. Table 7 summarizes all the analysis curves generated to measure user's annotation performance and characteristics of methods in the user study.

Table 7. Summarization of analysis curves for the measurements of annotation performance of users and characteristics of methods in the user study

Objective	Curve name	Description
Annotation performance of users	Annotation speed curve	Count of annotated entities over annotation time
	Annotation quality curve	Annotation quality in F-measure over sections
Characteristics of methods	Sentence count curve	Count of annotated sentences over annotation time
	Average sentence length curve	Words per sentence over annotation time
	Reading speed curve	Words in the annotated sentence over annotation time

To globally assess different learning curves, we computed the area under the learning curve (ALC) as a global score for each method, which is calculated as the area under the given learning curve (*actual area*) divided by a *maximum area* that represents the maximum performance. The *maximum area* is equal to the ultimate cost spent in training (e.g. number of words in the final training set or the actual annotation time) times the best possible F-measure. Ideally, the best F-measure is 1.0. However, the NER models could never achieve perfect under only 120-minute annotation. At this study, we used an F-measure of 0.75 as the best possible F-measure in 120-minute annotation.

3.3 Results

3.3.1 The Active LEARNER system

Figure 9 shows the first page when opening the *Active LEARNER* system, where we can specify the parameters used in the user study, such as user name, algorithm name, section time, and

selection of training mode and dataset. Then a profile folder is created for each unique parameter set. All annotation actions and models are stored in the corresponding profile folder.

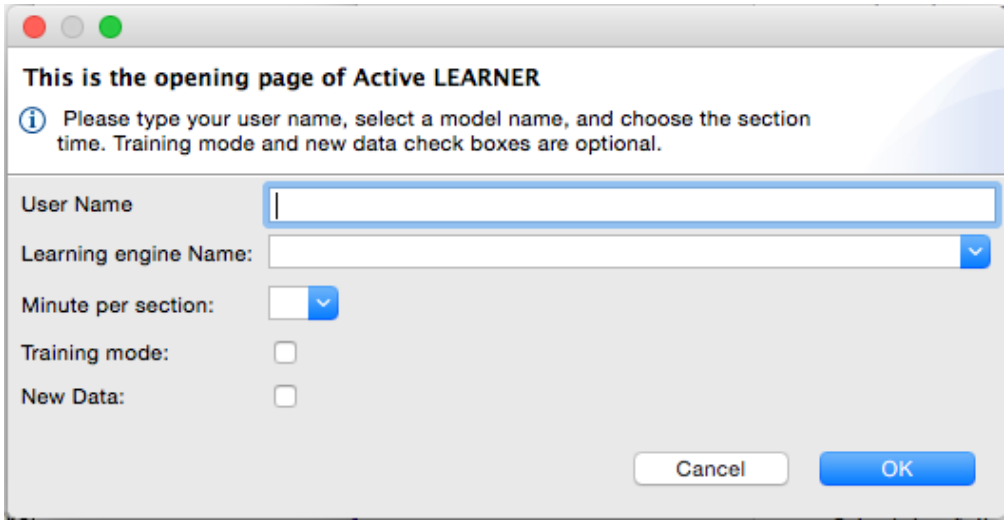

Figure 9. The initial interface to select parameters, such as user name, algorithm name, section time, training mode, and dataset

Figure 10 shows a screenshot of the main annotation interface, which consists of three parts: toolbar, sentence annotator, and document viewer. Toolbar is located at the top of the interface, which provides basic buttons for the user to control the program. “NEXT (Hot key: space)” or “Next” is for user to submit the current annotation and request the next sentence to annotate. “Prev” is to return to the previously annotated sentence. “Pause” is to stop the clock when the user needs to take a break from annotation. “Quit” is to close the program. The user can also use the drop-down list (in Figure 11) to select a sentence among all previously annotated sentences for modifications.

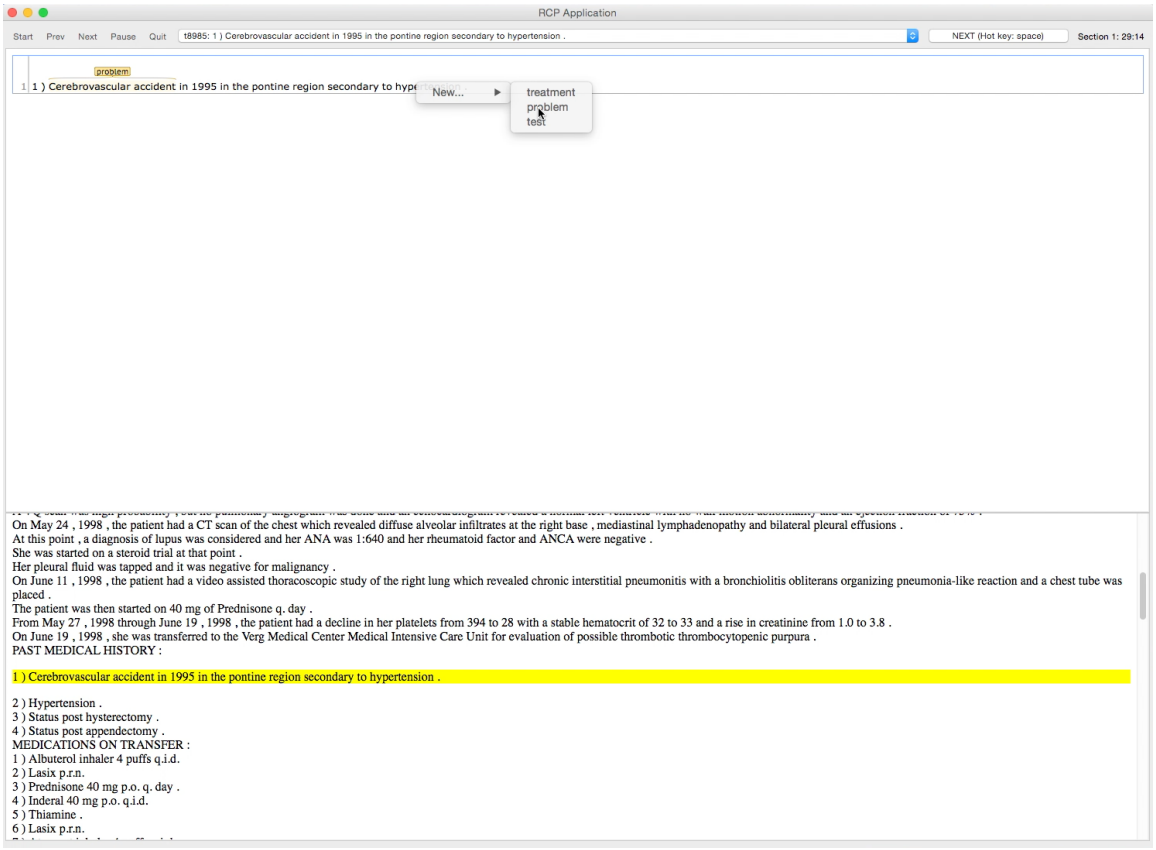

Figure 10. A screenshot on the main annotation interface

Figure 11. A screenshot of the drop-down list to select a sentence among all previously annotated sentences

The sentence annotator in the central part of the interface displays the sentence with annotation and provides functions to mark entity types for selected phrases in the sentence. We embedded and modified *BRAT* to support these functions for both displaying and labeling. Figure 12 demonstrates the marking process. The user can also modify the annotation of an entity by deleting the one previously marked and generating a new one.

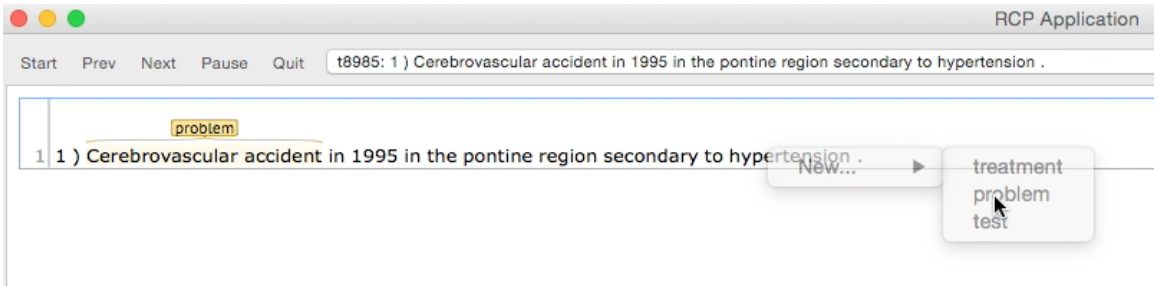

Figure 12. An enlarged screenshot on the annotation interface at the time when user was tagging "hypertension" as "problem" after "Cerebrovascular accident" was tagged as "problem"

Document viewer in the lower part of the interface displays the clinical document that contains the current sentence shown in the annotator. The target sentence is placed in the center of the document viewer and highlighted in yellow; but users can scroll up or down to read the whole document. However, the user cannot modify anything shown in the document viewer.

3.3.2 Simulated results

In the simulation, we evaluated methods of *Random*, *Uncertainty*, and *CAUSE* assuming same cost per word. Both *Uncertainty* and *CAUSE* utilized *LC* as the uncertainty measurement. The training process of *Uncertainty* and *LC* started from 5 initially selected sentences based on random sampling. *CAUSE* used random cluster and representative sampling (described in

Section 2.2.2.1) to select the initial 5 sentences. The batch size is 5 so that the model was updated with every additional 5 newly queried sentences. The AL process stopped at the point where there are as close as 7,200 words in the training set. This stopping criterion is to mimic the 120-minute (7,200 seconds) long user study per method, assuming the user would annotate approximately one word per second (see words per minute in Table 11).

Figure 13 shows the learning curves of *Random*, *Uncertainty*, and *CAUSE* in the same graph. Obviously, *CAUSE* outperformed Random and Uncertainty most of the time at all stages during the AL process. In terms of ALC score, *CAUSE* achieved 0.839, *Uncertainty* did 0.782, and *Random* did 0.812. At the point where there are ~7,200 words in the training set, *CAUSE* generated NER models with 0.713 in F-measure on average, while *Random* and *Uncertainty* achieved 0.696 and 0.697 in F-measure, respectively.

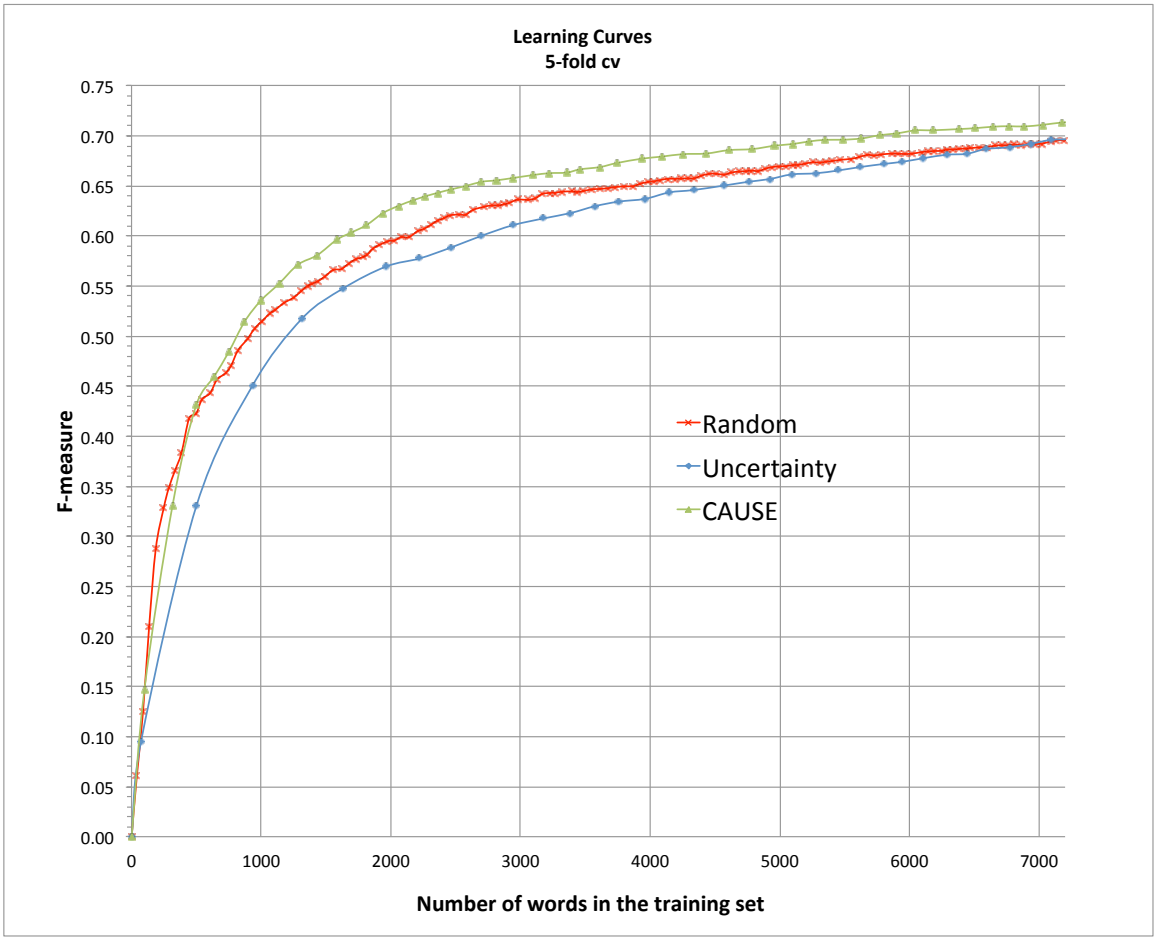

Figure 13. Simulated learning curves by 5-fold cross validation that plot F-measure vs. number of words in the training set for random sampling (*Random*), least confidence (*Uncertainty*), and *CAUSE* that used least confidence to measure uncertainty

3.3.3 User study results

For the user study, there are 16,338 unique sentences in the pool for querying and 4,085 unique sentences in the test set for evaluating NER models. Based on the simulated results, *CAUSE* performed better than Uncertainty. Therefore, we used *CAUSE* to represent AL in the user study and compared it with *Random* in the user study. The initial sentence selection schemas used in the user study were the same as the simulation. The batch size was set at 5, meaning the new

learning process would be activated when there were at least 5 newly labeled sentences added to the *labeled set*.

Table 8 reports the assessment of annotation information from the main studies. As we can see, two methods (*Random* and *CAUSE*) have very similar *annotation speed* and *annotation quality*, indicating both users' performances are stable and two methods could be comparable.

Table 8. Annotation counts, speed, and quality comparison in the 120-minute main study

Users	Methods	Annotated Entity count	Annotation speed (Entities per min)	Annotation quality (F-measure)
User 1	*Random*	945	7.88	0.82
	CAUSE	926	7.72	0.83
User 2	*Random*	882	7.35	0.81
	CAUSE	948	7.90	0.82

Figure 14 and 15 show the learning curves of F-measure versus annotation time in minutes by *Random* (in week 1) and *CAUSE* (in week 2) from two users. The experimental results for the two users were different. *Random* performed better than *CAUSE* for user 1; while *CAUSE* was superior to *Random* for user 2. Table 9 shows the ALC scores and F-measure of the final NER model at the end of 120 minutes annotation for *Random* and *CAUSE* from both users.

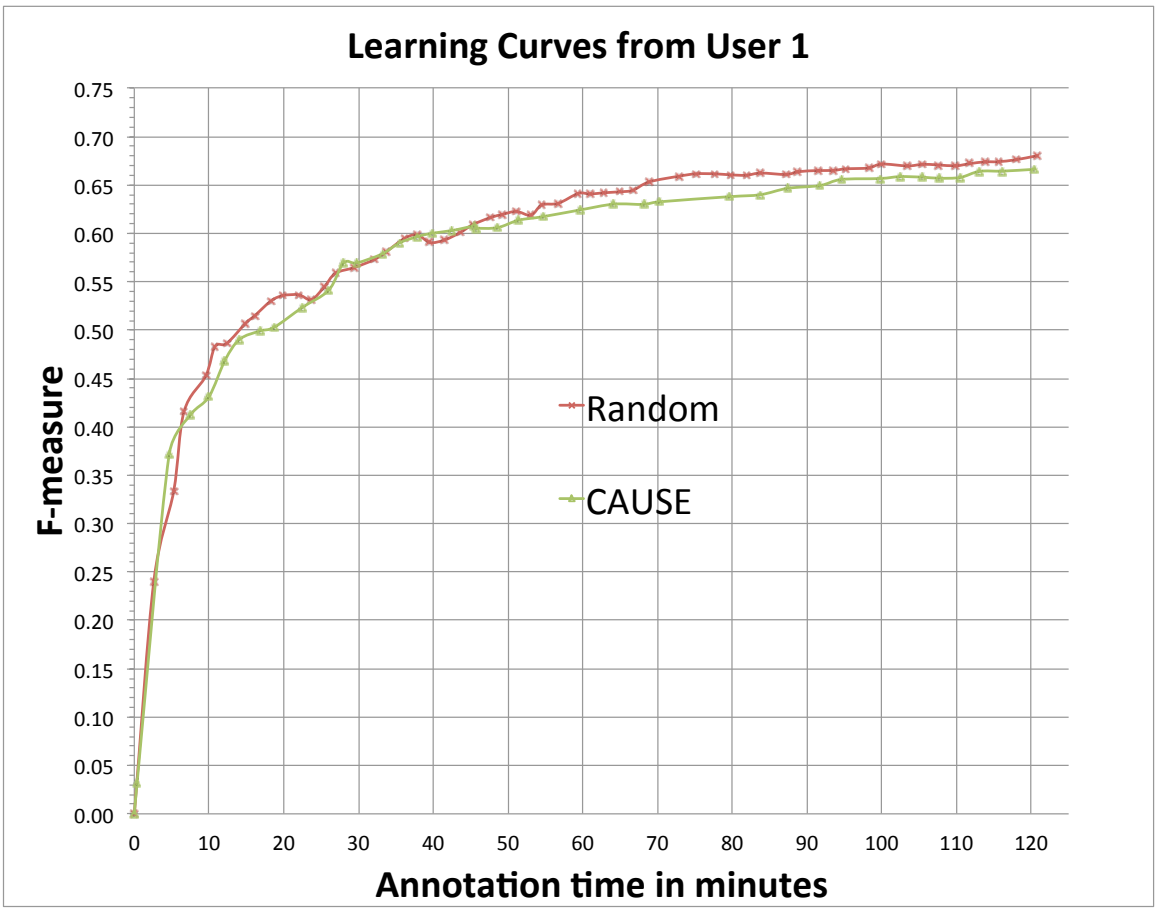

Figure 14. Learning curves of F-measure vs. annotation time in minutes by *Random* and *CAUSE* from user 1

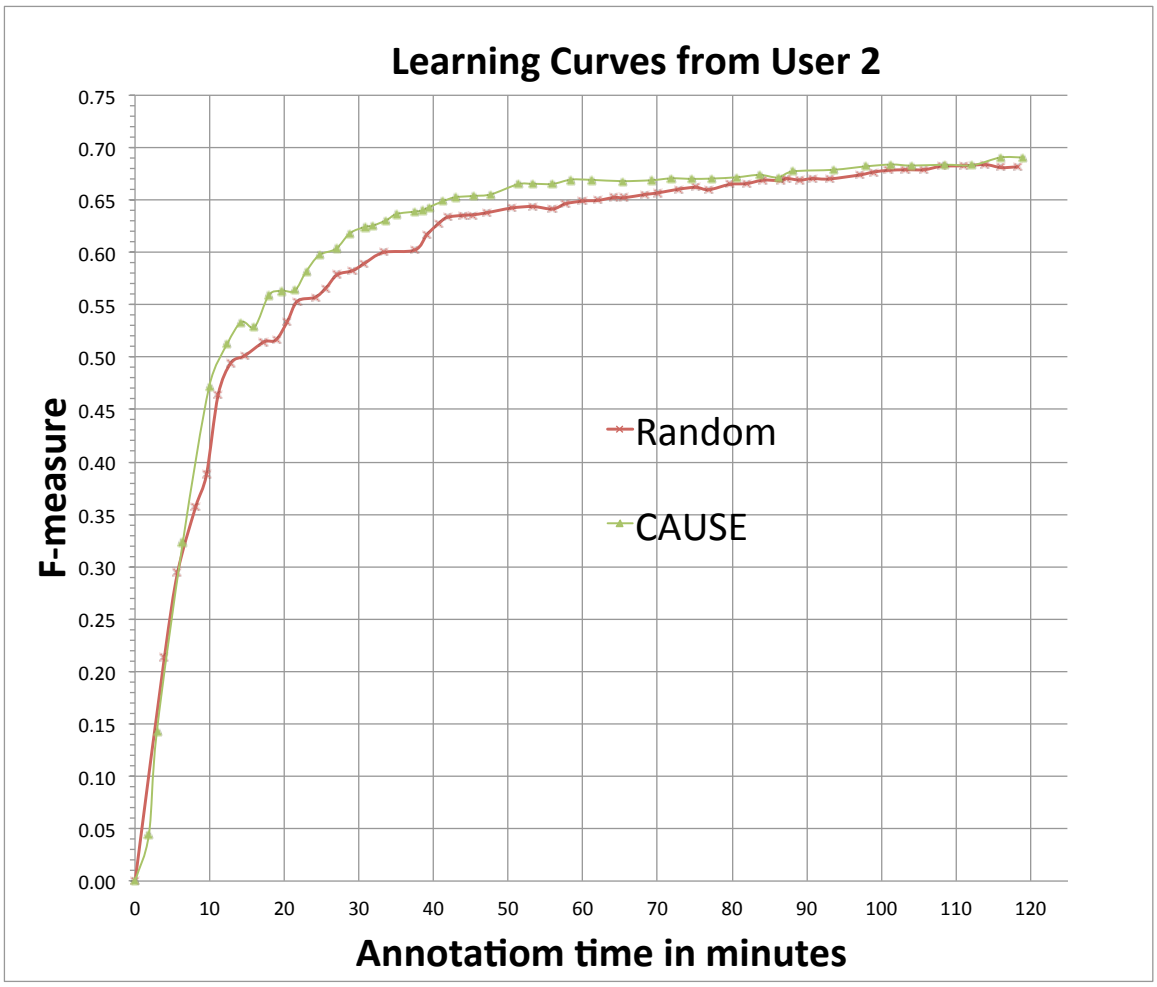

Figure 15. Learning curves of F-measure vs. annotation time in minutes by *Random* and *CAUSE* from user 2

Table 9. Comparison between *Random* and *CAUSE* in ALC score and F-measure of the last model in the 120-minute main study

User Index	Evaluated method	ALC scores	F-measure of models at 120 minutes
User 1	*Random*	0.812	0.680
	CAUSE	0.783	0.666
User 2	*Random*	0.820	0.682
	CAUSE	0.831	0.691

We also analyzed additional measures for both *CAUSE* and *RANDOM*, which are described in Table 7. Figure 16 shows the sentence count curves and it clearly shows that both users annotated many more sentences in *Random* than in *CAUSE*. The average sentence length curves (Figure 17) shows that sentences picked by CAUSE were almost two times longer than that by RANDOM, which explains why users can annotate more sentences queried by *Random* than *CAUSE*.

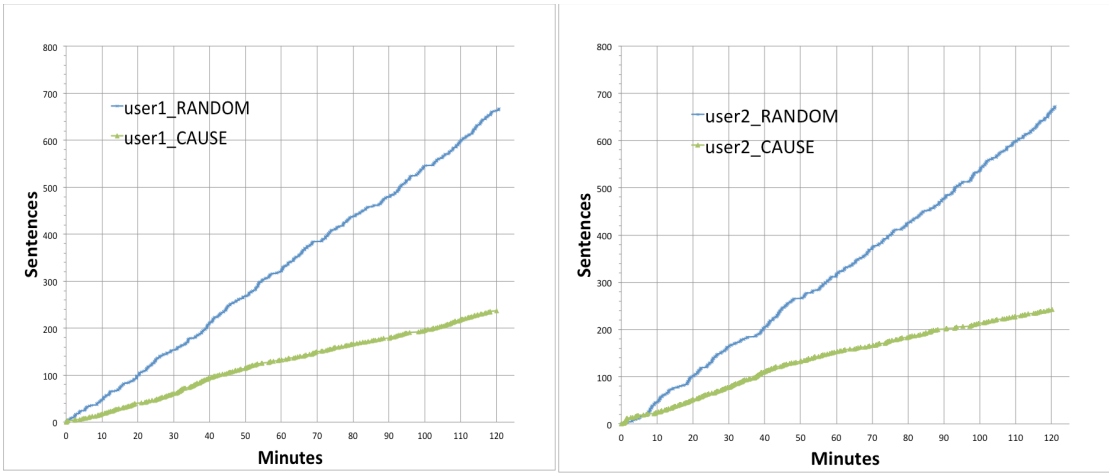

Figure 16. Sentence count curves of the number of annotated sentences over the annotation time in minute from the main studies of *Random* and *CAUSE* by user 1 and user 2

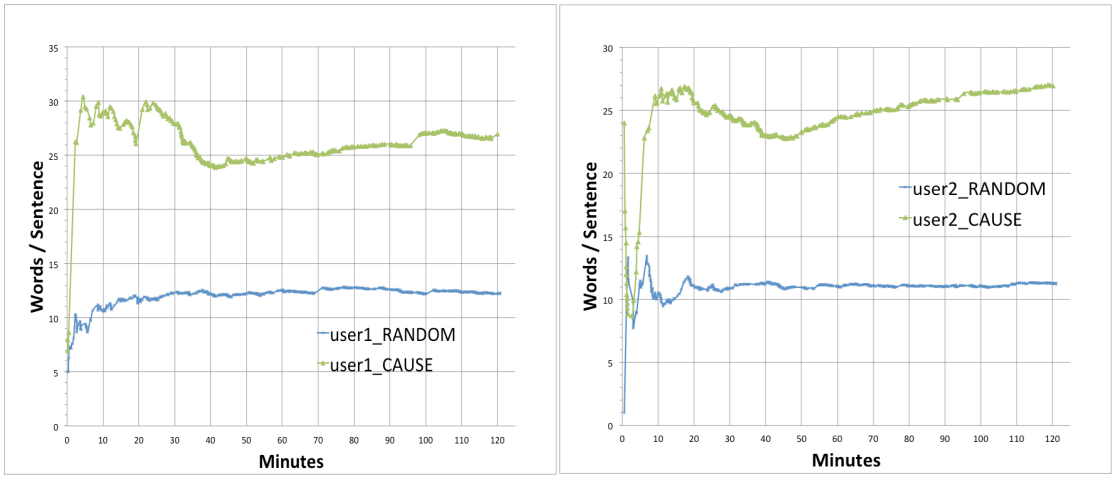

Figure 17. Sentence length curves (words per sentence over the annotation time) from the main studies of *Random* and *CAUSE* by user 1 and 2

Figure 18 shows the reading speed curves that plot the number of words in the annotated sentences over time. This figure suggests that users could review sentences queried by *Random* faster than that by *CAUSE*, especially for user 1.

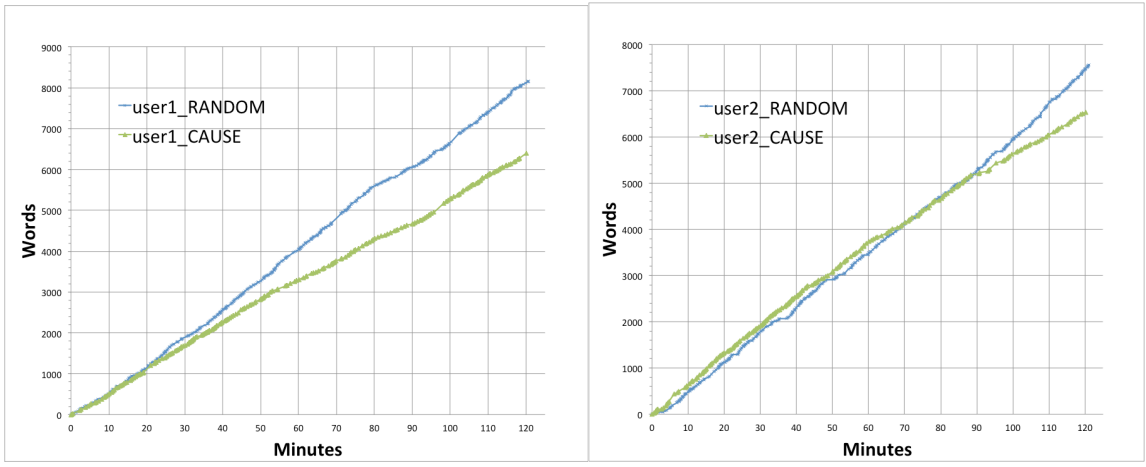

Figure 18. Reading speed curves that plot the number of words in the annotated sentences over annotation time in minute from the main studies of *Random* and *CAUSE* by user 1 and user 2

Figure 19 presents the annotation speed curves. Overall the annotation speeds for both users were relatively consistent. User 1's *annotation speeds* for *CAUSE* and *Random* were very similar. For the annotation by user 2, *CAUSE* showed higher annotation speed than *Random*. It seems that user 2 performed more efficiently in the *CAUSE* study than in the *Random* study.

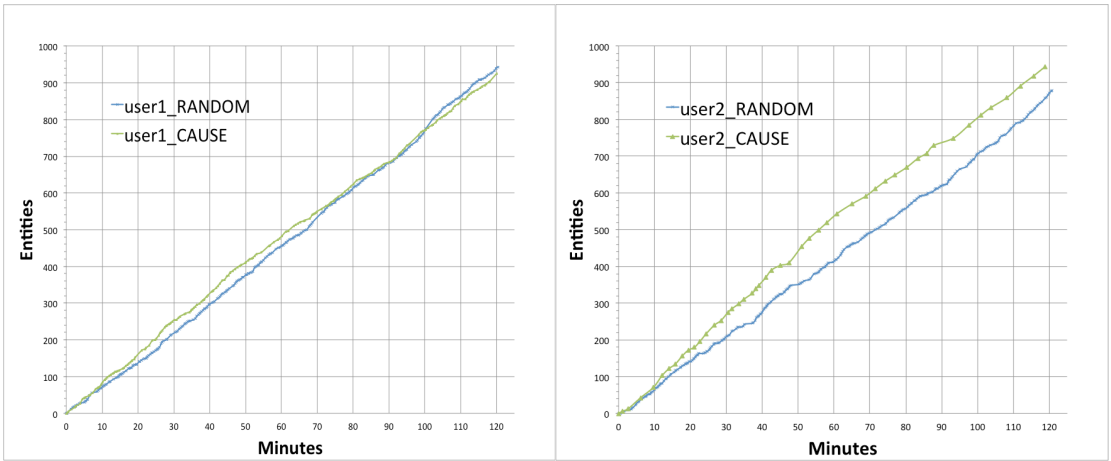

Figure 19. Annotation speed curves that plot the entity annotations over the annotation time in minute from the main studies of *Random* and *CAUSE* by user 1 and user 2

Table 10 and 11 summarize the characteristics of *Random* and *CAUSE* in each 120-minute main study from both users. Both users annotated more sentences in the *Random* mode than that in the *CAUSE* mode, very likely due to shorter length of sentences selected by *Random*. Moreover, users seemed to read the words queried by *Random* faster than *CAUSE*. The entity number per sentence by *CAUSE* is about 3 times higher than that in *Random*. Entity density by *CAUSE* is also higher than that by *Random*.

Table 10. Characteristics of *Random* and *CAUSE* in each 120-minute main study from user 1 and 2 (part 1)

User	Method	Annotated Sentences	Words in annotated sentences	Entities in annotated sentences	Words in entities
User 1	*Random*	655	8,023	945	1,915
	CAUSE	232	6,333	926	2,145
User 2	*Random*	651	7,325	882	1,952
	CAUSE	240	6,455	948	2,404

Table 11. Characteristics of *Random* and *CAUSE* in each 120-minute main study from user 1 and 2 (part 2)

User	Method	Sentences per min	Words per sentence	Words per min	Entities Per Sentence	Entity Density
User 1	*Random*	5.53	12.24	67.70	1.44	0.24
	CAUSE	1.97	27.00	53.30	3.99	0.34
User 2	*Random*	5.55	11.25	62.44	1.35	0.27
	CAUSE	2.01	26.98	54.33	3.95	0.37

3.4 Discussion

This is the first study that integrates AL with annotation processes to build clinical NER systems and evaluates it in a real-world task by engaging users. Although many previous AL studies showed substantial savings of annotation in terms of number of samples in simulation, our real world experiments showed that current AL methods did not guarantee savings of annotation time for all users in practice.

This finding could be due to multiple reasons. First, although AL selected more informative sentences and required fewer sentences for building NER models, it often selects longer sentences with more entities, which take a longer time to annotate. Users annotated ~240 sentences queried by *CAUSE* in 120 minutes (~2.0 sentences per minute) versus ~660 sentences by *Random* in the same time (~5.5 sentences per minute). Our results suggest that the increased information content of actively selected sentences is strongly offset by the increased time required to annotate them. Moreover, it seems that users may have different behaviors for sentences selected by different methods. For example, it seemed that users read randomly sampled sentences faster (62-68 words per minute) than AL selected sentences (53-54 words per minute). All these results demonstrate that AL in practice could be very different from simulation studies and it is critical to benchmark AL algorithms using real-world practical measurements (such as annotation time), instead of theoretical measurements (such as the number of training sentences and the number of words in training sentences).

There are many other factors that may affect users, thus contributing to the final results. First of all, different users have different behaviors when annotating clinical text. For example, one annotator reviewed sentences once only and very quickly; but the other often reviewed a sentence twice after marking the entities. In addition, users' responses to AL-selected sentences could also be different in terms of annotation speed and annotation quality. Moreover, for the same user, the annotation behavior could have varied during the study. Our study design allowed user to make more consistent annotation in every session by adding breaks between sessions. However, as the whole process for evaluating one method could take over 4 hours including the warm-up session, users could be very exhausted for the last one or two sessions.

We assessed the *annotation speed* and *annotation quality* curves over four sessions for both users (Figure 20 and 21). In terms of annotation speed, both users conducted the annotation faster in the first two sessions (Session 1 and 2) than the last two sessions (Session 3 and 4) in the *CAUSE mode* (according to green curves in Figure 20). Conversely, in the *Random* model, both users' annotation speed achieved the highest point at the last section, although it was fairly consistent in the first three sections (according to red curves in Figure 20). With respect to annotation quality (Figure 21), both users showed relatively high variance in F-measure, ranging from 0.75 to 0.87 without obvious patterns.

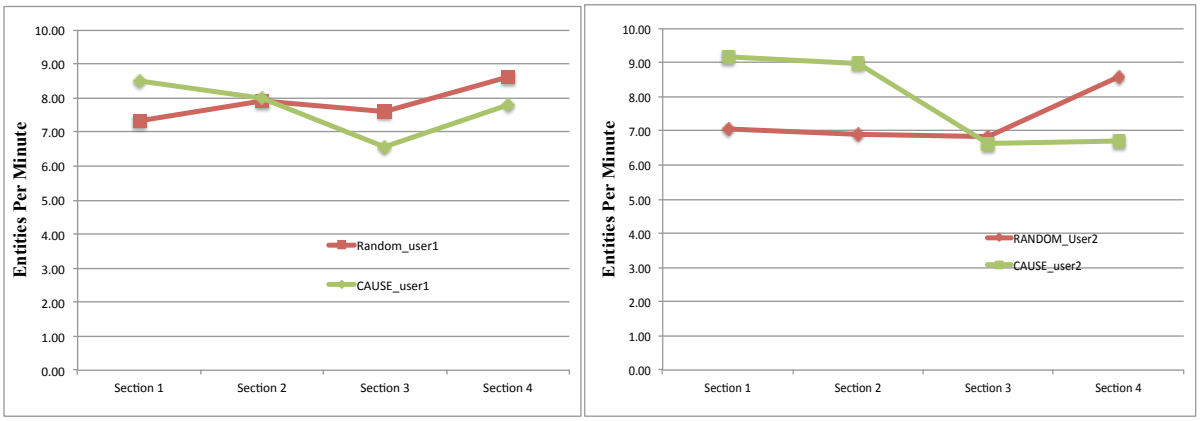

Figure 20. Annotation speeds per section in the main studies of *Random* and *CAUSE* from user 1 and 2

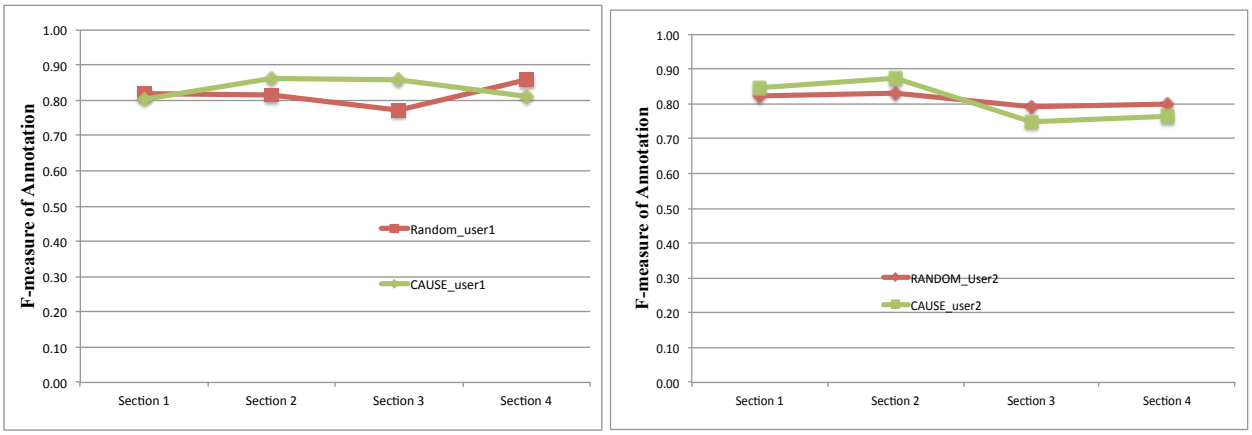

Figure 21. Annotation qualities per section in the main studies of *Random* and *CAUSE* from user 1 and 2

Although the results in this user study showed that the current AL methods could not be guaranteed to save annotation time, compared to passive learning, we gained valuable information about why it happened. If the querying algorithm accounts for the actual annotation time in the model, we believe AL could perform better. Therefore, the next phase of our work will include improving our AL algorithms against the practical measures (i.e., annotation time). One of our plans is to use annotation data collected in this study to develop regression models, which can more accurately estimate annotation time of unlabeled sentences, thus optimizing the AL algorithms for actual annotation time instead of number of samples.

CHAPTER 4

Annotation Time Modeling for Active Learning in Clinical Named Entity Recognition

4.1 Introduction

In the previous chapter, we describe an AL-enabled annotation system for NER (*Active LEARNER*), where we developed a novel AL algorithm called *CAUSE* that considers both uncertainty and representativeness of sentences. In a simulation study where the annotation cost was estimated based on the number of words in the annotated sentences, *CAUSE* showed superior performance than baseline methods including the uncertainty sampling method and random sampling. In the user study, however, we found mixed results: *CAUSE* generated a better learning curve than random sampling for one user, but not the other. This finding indicates that the *CAUSE* algorithm is not guaranteed to save actual annotation time for each user, when compared to random sampling.

One potential direction to improve the *CAUSE* algorithm is to develop better models for estimating actual annotation cost (i.e., time), instead of simply assuming the length of a sentence is the only factor that accounts for annotation time. In the AL field, there are several studies that investigated cost-sensitive AL. Settles et al. [60] reported an empirical study of AL's impact on real annotation costs. One of their conclusions is that AL approaches that ignore cost information may perform no better than random sampling. However, improved learning curves are achievable if the cost variables can be appropriately taken into account. Haertel et al. [61] also presented a practical cost-conscious AL approach based on return on investment (ROI). They

evaluated the ROI based AL on a part-of-speech tagging task and showed that ROI achieved as high as a 73 % reduction over random in hourly cost.

Inspired by these studies, here we developed a new querying engine in *Active LEARNER* called *CAUSE* 2.0 (*CAUSE2*), which queries the sentences that are most informative based on annotation time models. Annotated data from both users collected from the previous user study were used to generate the annotation time predictive models. To assess these models, we conducted three types of evaluations: 1) we fitted the models to the training data and reported the coefficient of determination (R^2); 2) we conducted simulation studies using another pre-annotated dataset to systematically evaluate fifteen methods from four different categories (i.e. four in the category of uncertainty sampling, four in *CAUSE*, six in *CAUSE2*, and one in passive learning) using the estimated annotation time predicted by the models as the cost; and 3) we integrated the best performing algorithm *CAUSE2* with *Active LEARNER* and conducted another user study to compare *CAUSE2* and random sampling in the real-time annotation task.

Based on the regression results, the proposed annotation time models achieved R^2 of 0.79 and 0.53 for user 1 and user 2, respectively; while the baseline models achieved 0.67 and 0.46 for user 1 and user 2, respectively. The proposed method outperformed baseline for both users by an improvement of 15-16% in R^2.

The simulated results based on learning curves showed that *CAUSE2* outperformed *CAUSE,* uncertainty sampling, and random sampling. Uncertainty sampling, which was previously considered as a promising AL strategy, required more time when the new annotation time model was used. Among all AL methods based on the new time estimation model, *CAUSE2* was the only algorithm that was significantly better than random sampling.

The same two users participated in the new user study to evaluate random sampling and *CAUSE2*. The results showed that *CAUSE2* performed globally better than random sampling in terms of the area under the learning curve scores for both users. However, the advantage margin of *CAUSE2* vs. random sampling is different between user 1 and user 2. Finally, we discuss the advantages and the limitations of the proposed methods and lay out the direction for future work.

4.2 Methods

4.2.1 Active learning with annotation time models

Previously developed AL methods including *CAUSE* only account for the informativeness of samples for building ML models (e.g., uncertainty and representativeness). However, the user study shows that without considering the annotation cost in the model, AL is not able to reduce actual annotation cost (i.e., time).

Inspired by Settles et al. [60], who developed a simple heuristic that divides the utility measure by the predicted cost, and Haertel et al. [61], who suggested the ROI AL method, we propose a similar AL strategy that queries sentences that are most informative and least costly. Basically, we use the ratio between the informativeness of a sentence *s* - *Informativeness (s)* and the estimated annotation time of *s – Cost (s),* instead of *Informativeness (s)* only, to rank sentences. Any previous querying algorithm, such as *CAUSE*, can determine the *Informativeness* (s). The *Cost(s)* could be an annotation time model, which is further explained in following paragraphs. In the case of *CAUSE*, we name the new algorithm that considers the estimated annotation time as *CAUSE2*. Therefore, *CAUSE2* is the first kind of AL algorithm that considers three aspects to

rank sentences: (1) uncertainty based on the NER model, (2) representativeness based on the clustering, and (3) annotation cost based on the annotation time model.

We developed a simple linear regression model to predict the annotation time of unlabeled sentences. This model attempts to mimic the annotator's thought process in general, which was observed in the previous user study. We divide the annotation process into three procedures, represented by the following features in a sentence: number of words, number of words as part of the concepts, and number of concepts. The linear model for annotation time estimation is described in the following formula:

$$AT(s) = w_0 + w_1 * X(s) + w_2 * Y(s) + w_3 * Z(s),$$

where $AT(s)$ is the estimated annotation time in second for an input sentence s; $X(s)$ is the number of words in sentence s; $Y(s)$ is the number of words tagged as part of the entities in sentence s; and $Z(s)$ is the number of entities in sentence s. The weights for the three features can be interpreted as the following: w_1 is the time in seconds for the annotator to scan the sentence word by word; w_2 is the time in seconds for the annotator to determine the boundaries of potential entities; w_3 is the time in seconds for the annotator to mark the identified entities through the interface. The intercept in this linear model, w_0, is an unrealistic time in seconds to annotate a sentence with zero words, zero words as entity, and zero entity. It could be interpreted as the "idle time" during annotation when users are taking a tiny break for 1 – 3 seconds per sentence. In summary, this simple model can capture the general annotation process and estimate the annotation time as the summation of the time of initial word-by-word reading, the time of boundary identification for the entities, and the time of marking the identified entities.

We also generated a baseline linear regression model to predict annotation time by using only number of words as a single feature. Here is the baseline model:

$AT(s) = w_0 + w_1 * X(s)$,

where $AT(s)$ is the estimated annotation time in second for an input sentence *s*; $X(s)$ is the number of words in sentence *s*. The weight of $X(s)$, w_1, is basically the *reading speed* as we defined in Chapter 3. We compared the proposed and the baseline linear regression models here.

As annotation time for the same sentence is different from one annotator to another, we trained a regression model for each annotator based on existing individual annotated data from the previous study in Chapter 2.

4.2.2 Datasets

4.2.2.1 Training dataset for building annotation time models

From the previous user study reported in Chapter 3, we have collected the annotated data in week 1 and week 2, resulting in 240-minute annotated data per user, for training the individual annotation time models. Table 12 shows the distribution of features in training data for both users.

Table 12. Distributions of the training data for building annotation time models

User	Total annotated sentences	Annotation time (second) per annotated sentence		Words per annotated sentence		Words as entity per annotated sentence		Entities per annotated sentences	
		Mean	SD	Mean	SD	Mean	SD	Mean	SD
User 1	887	16.28	18.15	16.19	17.86	4.58	5.53	2.11	2.75
User 2	891	16.24	19.18	15.48	14.39	4.89	5.88	2.05	2.42

4.2.2.2 Dataset for simulation studies

In the simulation studies, we used the same dataset as the one used in Chapter 2 (the training corpus from the 2010 i2b2/VA NLP challenge). The dataset contains 20,423 unique sentences and was randomly split into five folds. We performed 5-fold cross validation so that four out of five folds were used as the pool of data to be queried and the remaining fold was the independent test set for evaluation.

4.2.2.3 Dataset for the user study

In the new user study, the test corpus of the 2010 i2b2/VA challenge, which has 477 clinical documents with 29,789 unique sentences, was used as the pool for querying sentences. Table 13 shows the distribution of words and different types of entities in that dataset. The entity density of this corpus is 0.25, which was calculated by the number of words in entities (or entity words) divided by the total number of words. The training corpus of the 2010 i2b2/VA challenge, which

contains 349 clinical documents with 20,423 unique sentences, was used to evaluate the NER models to generate the learning curves.

Table 13. Distribution of words and different types of entities in the pool of 29,789 unique sentences

	Overall count	Mean of count per sentence	StDev of count per sentence
Word	341,982	11.48	9.80
Entity	41,624	1.40	1.73
Problem entity	16,796	0.56	1.04
Treatment entity	12,740	0.43	0.93
Test entity	12,088	0.41	1.15

4.2.3 Evaluation

4.2.3.1 Evaluation of annotation time models

To evaluate linear regression models, we used R^2 to find how well the actual annotation time fits an annotation cost model. In this study, R^2 was calculated as the square of the *Pearson correlation coefficient* between the actual annotation time and the estimated annotation time. We simply assume that the model with higher R^2 is better at estimating annotation time.

4.2.3.2 Evaluation using the simulation study

Using a similar simulation study design as in Chapter 2 and 3, we evaluated fifteen querying algorithms including nine existing methods that do not consider annotation costs (from three categories: Uncertainty sampling, *CAUSE*, and *Random*) and six new methods that consider the

annotation estimation time models (from the CAUSE2 category). We describe these methods as following:

Uncertainty Methods:

(1) Least Confidence (*LC*): to take the uncertainty from the best possible sequence label based on the posterior probability output from CRF. The uncertainty of a sentence is equal to $1 - P(y^*|x)$, where y^* is the most likely sequence label.

(2) *N-best sequence entropy (nBest)*: to take the entropy of the probability distribution over N-best sequence labels predicted by the CRF model. The probabilities of the N-best sequence labels were normalized so that the sum of them is equal to 1. We used N=3 in our experiments.

(3) *Word entropy*: to take the summation of entropy of individual words given the probability distribution over all possible labels.

(4) *Entity entropy*: to take the summation of entropy of the beginning word of the estimated entities (e.g. B-entity; excluding the entropy from the inside "I" and outside "O" of the estimated entities).

***CAUSE* Methods:**

(1) *CAUSE_LC*: clustering and uncertainty sampling based on *least confidence* as the uncertainty measurement;

(2) *CAUSE_nBest*: clustering and uncertainty sampling based on *N-best sequence entropy* as the uncertainty measurement;

(3) *CAUSE_WordEntropy*: clustering and uncertainty sampling based on *word entropy* as the uncertainty measurement;

(4) *CAUSE_EntityEntropy*: clustering and uncertainty sampling based on *entity entropy* as the uncertainty measurement.

***CAUSE2* Methods**:

(1) *CAUSE_LCPerCost*: clustering and uncertainty per estimated annotation cost sampling based on $UPC(s) = LC(s) / AT(s)$, where $LC(s)$ is the *least confidence* of sentence s and $AT(s)$ is the estimated annotation time in second for sentence s;

(2) *CAUSE_nBestPerCost*: clustering and uncertainty per estimated annotation cost sampling based on $UPC(s) = nBest(s) / AT(s)$, where $nBest(s)$ is the *N-best sequence entropy* of sentence s and $AT(s)$ is the estimated annotation time in second for sentence s;

(3) *CAUSE_WordEntropyPerCost*: clustering and uncertainty per estimated annotation cost sampling based on $UPC(s) = WordEntropy(s) / AT(s)$, where $WordEntropy(s)$ is the *word entropy* of sentence s and $AT(s)$ is the estimated annotation time in second for sentence s;

(4) *CAUSE_EntityEntropyPerCost*: clustering and uncertainty per estimated annotation cost sampling based on $UPC(s) = EntityEntropy(s) / AT(s)$, where $EntityEntropy(s)$ is the *entity entropy* of sentence s and $AT(s)$ is the estimated annotation time in second for sentence s;

(5) *CAUSE_WordEntropyPerWord*: clustering and uncertainty per estimated annotation cost sampling based on $UPC(s) = WordEntropy(s) / X(s)$, where $WordEntropy(s)$ is the *word entropy* of sentence s and $X(s)$ is the number of words in sentence s;

(6) *CAUSE_EntityEntropyPerEntity*: clustering and uncertainty per estimated annotation cost sampling based on $UPC(s) = EntityEntropy(s) / Z(s)$, where $EntityEntropy(s)$ is the *entity entropy* of sentence s and $Z(s)$ is the number of estimated entities in sentence s;

As shown in the result section, our proposed annotation estimation model is a better estimate of time than the model based on the number of words in the sentences only. Therefore, we used the predicted annotation time based on the model trained from individual users to generate the learning curves that plot F-measures vs. estimated annotation time. For each method, five learning curves from the 5-fold cross validation were generated and averaged to a final learning curve, which simulates the 120-minute user study.

The ALC score was used to assess the global performance for each method. The derivation of ALC score was described in the evaluation section in Chapter 3. We also compared final F-measures at the end of 120 minutes. In addition, we also reported other characteristics of different methods based on the cross validation data from sentences queried at the end of 120 minutes, such as average sentence length (words per sentence), average sentence entity count (entities per sentence), and average entity density (entity words per word).

4.2.3.3 Evaluation by the user study

To further validate the utility of the new *CAUSE2* method in practice, we conducted a similar user study as that in Chapter 3 to compare the best-performing algorithm in *CAUSE2* with the random sampling method. The same two users participated in this user study, following a similar study design as the previous one. Table 14 shows the schedule of the new user study, which was slightly adjusted. Compared to the user study in Chapter 3, the workload per day is reduced to 2 hours per day, starting with the sentence-by-sentence annotation review session for 30 minutes as the warm up training, followed by two 30-minute annotation sessions with 10-15 minutes break in between. Two days were needed to evaluate one method now. We hope that the reduced

workload per day can ensure users to be more consistent during the entire annotation period for each method.

Table 14. Schedule of the new user study using new data

Week	**Day**	**Event**	**Task**	**Duration** (minutes)
Week 1	Day 1	Main study using new data for *Random* (part 1)	1. Sentence-by-sentence annotation and review	30
			2. Two 30-minute sessions (Session 1 and 2) of main study using new data for *Random*	75
	Day 2	Main study using new data for *Random* (part 2)	1. Sentence-by-sentence annotation and review	30
			2. Two 30-minute sessions (Session 3 and 4) of main study using new data for *Random*	75
Week 2	Day 1	Main study using new data for *CAUSE2* (part 1)	1. Sentence-by-sentence annotation and review	30
			2. Two 30-minute sessions (Session 1 and 2) of main study using new data for *CAUSE2*	75
	Day 2	Main study using new data for *CAUSE2* (part 2)	1. Sentence-by-sentence annotation and review	30
			2. Two 30-minute sessions (Session 3 and 4) of main study using new data for *CAUSE2*	75

We then generated the learning curves that plot F-measure vs. actual annotation time for each method. The global ALC scores and the final F-measures were calculated to compare two methods.

4.3 Results

4.3.1 Annotation cost models evaluation results

We trained both the baseline and the proposed annotation cost models for individual users using their own annotated data from previous studies (described in Dataset section). Figure 22 shows the baseline annotation cost models based on sentence length only for user 1 and user 2, which did not fit well to the data.

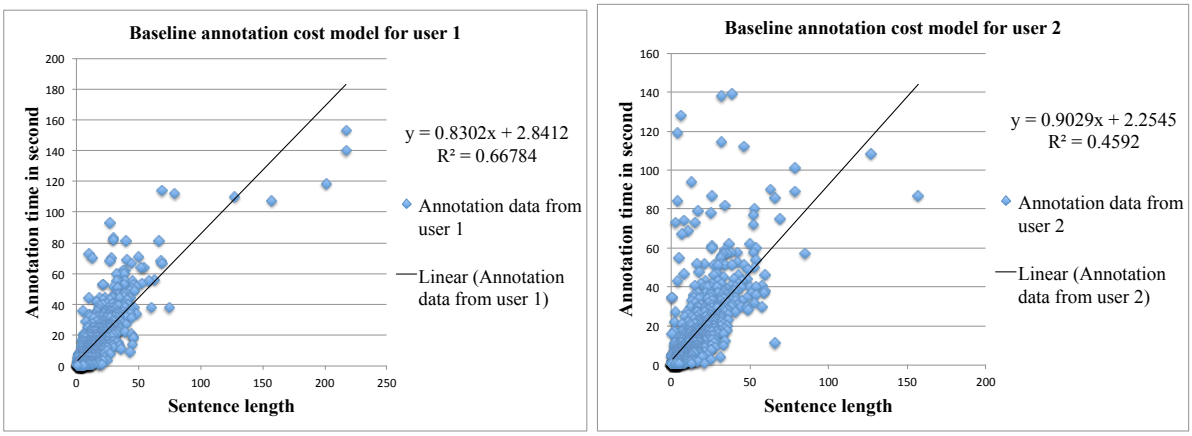

Figure 22. The baseline annotation cost models for user 1 and user 2

For the proposed annotation cost estimation models, we show the trained models for user 1 and user 2 below:

For user 1: $AT(s) = 2.13 + 0.24*X(s) + 1.57*Y(s) + 1.43*Z(s)$

For user 2: AT(s) = 2.69 + 0.32*X(s) + 1.08*Y(s) + 1.63*Z(s)

More statistical analysis for the annotation cost models for both users is summarized in Table 15 and 16. These results show that the weights of all three predictors and the intercept are significant in the annotation time models for both users.

Table 15. Statistical analysis for annotation cost model for user 1

Coefficients	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	2.13	0.38	5.63	2.43E-08
Number of Entities (Z)	1.43	0.23	6.22	7.68E-10
Number of Entity Words (Y)	1.57	0.11	14.41	<2.00E-16
Number of Words (X)	0.24	0.03	7.88	9.31E-15

Note: Residual standard error: 8.299 on 883 degrees of freedom; Multiple R-squared: 0.7916, Adjusted R-squared: 0.7909; F-statistic: 1118 on 3 and 883 DF, p-value: < 2.2e-16.

Table 16. Statistical analysis for annotation cost model for user 2

Coefficients	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	2.69	0.65	4.14	3.73E-05
Number of Entities (Z)	1.63	0.36	4.47	8.85E-06
Number of Entity Words (Y)	1.08	0.15	7.37	3.82E-13
Number of Words (X)	0.32	0.06	5.34	1.21E-07

Note: Residual standard error: 13.17 on 887 degrees of freedom; Multiple R-squared: 0.5298, Adjusted R-squared: 0.5282; F-statistic: 333.1 on 3 and 887 DF, p-value: < 2.2e-16

The models can be interpreted as the following: users spent 0.24 – 0.32 seconds in scanning words, 1.08 – 1.57 seconds in identifying the boundaries of entities, and 1.43 – 1.63 seconds in marking entities with "idle time" of 2.13 – 2.69 seconds to annotate a sentence. Table 17 shows R^2 of the baseline and the proposed annotation time models for both users. The annotation time model for user 2 was not as good as the model for user 1.

Table 17. Evaluation of different annotation cost models in R^2

User	Baseline annotation cost models	Proposed annotation cost models
User 1	0.67	0.79
User 2	0.46	0.53

4.3.2 Results of the simulation studies

Table 18 shows the results of ALC scores of both users for different methods. The *CAUSE2* methods that consider both informativeness and cost achieved superior performance, when compared to other methods that consider informativeness only. There are small differences between the two users for different *CAUSE* Methods.

Table 18. ALC scores of both users for different AL methods in the simulation study

Categories	Methods	ALC scores for user 1	ALC scores for user 2
Uncertainty based sampling methods (*Uncertainty*)	*LC*	0.783	0.785
	N-best sequence entropy (nBest)	0.824	0.826
	Word entropy	0.766	0.763
	Entity entropy	0.774	0.773
Clustering and uncertainty sampling methods (*CAUSE*)	*CAUSE_LC*	0.841	0.842
	CAUSE_nBest	0.847	0.848
	CAUSE_WordEntropy	0.833	0.833
	CAUSE_EntityEntropy	0.837	0.837
Clustering and uncertainty per estimated cost sampling methods (*CAUSE2*)	*CAUSE_LCPerCost*	0.870	0.882
	CAUSE_nBestPerCost	0.859	0.862
	CAUSE_WordEntropyPerCost	0.845	0.888
	CAUSE_EntityEntropyPerCost	0.884	0.859
	CAUSE_WordEntropyPerWord	0.853	0.854
	CAUSE_EntityEntropyPerEntity	0.861	0.865
Passive Learning (*Random*)	*Random*	0.840	0.840

Figure 23 shows the learning curves of four different methods from user 1, consisting of the best performing method from each category: *Random*, *N-best sequence entropy* (*Uncertainty*), *CAUSE_nBest* (*CAUSE*), and *CAUSE_EntityEntropyPerCost* (*CAUSE2*). *CAUSE2* seemed to outperform all other methods (learning curve was above others).

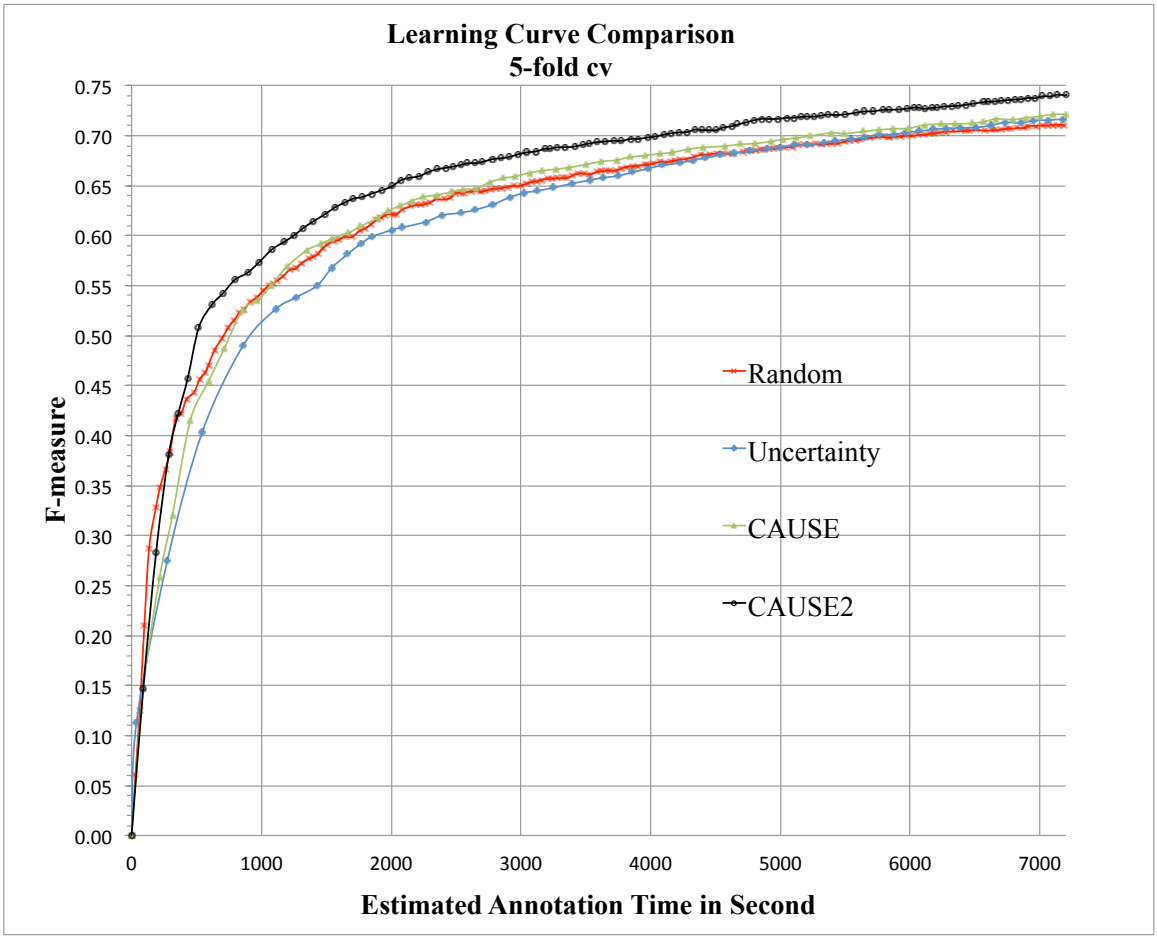

Figure 23. Learning curves of the best-performing method in each of four categories: *Random*, *Uncertainty (N-best sequence entropy)*, *CAUSE (CAUSE_nbest)*, and *CAUSE2 (CAUSE_EntityEntropyPerCost),* for user 1

Note: the learning curves are averaged based on the 5-fold cross validation results

Table 19 shows characteristics of the sentences queried by fifteen methods in 120 estimated minutes, including average sentence length, entities per sentence, and entity density. Sentences randomly selected have very similar characteristics (~11 words per sentence, 1.25 entities per sentence, and 0.24 in entity density) compared to the entire corpus. Sentences queried by *Uncertainty* are very long (25 – 45 words per sentence); sentences by *CAUSE* are shorter (22 – 28 words per sentence); and sentences by *CAUSE2* are even shorter (7 – 20 words per sentence).

Similar patterns were found in terms of entities per sentence. However, although the *CAUSE2* sentences are shorter, they have higher entity density. These findings suggest that *CAUSE2* selected sentences with reduced annotation difficulty, while retaining informativeness of these sentences.

Table 19. Characteristics in average sentence length, entities per sentence, and entity density for different AL methods

Categories	Methods	Average sentence length (words per sentence)	Entities per sentence	Entity density (entity words per word)
Uncertainty based sampling methods (Uncertainty)	*LC*	42.65	6.46	0.36
	N-best sequence entropy (nBest)	25.30	3.79	0.34
	Word entropy	44.81	6.22	0.37
	Entity entropy	41.17	7.02	0.37
Clustering and uncertainty sampling methods (CAUSE)	*CAUSE_LC*	27.59	3.91	0.35
	CAUSE_nBest	22.76	3.20	0.35
	CAUSE_WordEntropy	32.08	4.33	0.37
	CAUSE_EntityEntropy	28.61	4.49	0.36
Clustering and uncertainty per estimated cost sampling methods (CAUSE2)	*CAUSE_LCPerCost*	8.85	1.09	0.27
	CAUSE_nBestPerCost	7.31	0.87	0.24
	CAUSE_WordEntropy PerCost	14.31	1.64	0.30
	CAUSE_EntityEntropy PerCost	13.12	2.07	0.33
	CAUSE_WordEntropy PerWord	12.18	2.00	0.43
	CAUSE_EntityEntropy PerEntity	20.25	2.67	0.34
Passive Learning	*Random*	11.08	1.25	0.24

4.3.3 Results of the user study

CAUSE_EntityEntropyPerCost, the best *CAUSE2* method based on the simulation studies, was implemented in *Active LEANER* and compared with random sampling in the user study. The annotation time models described in Section 3.1 was used in the user study. Figure 24 and 25 display the learning curves of the 120-minute user studies for user 1 and user 2, respectively. Based on the learning curves, we further calculated ALC scores and F-measures of the NER systems at the end of 120-minute annotation. For each user, we also performed a statistical analysis to test whether *CAUSE2* is statistically significantly different from *Random* in terms of learning curves. Appendix A shows the details of the statistical analysis based on the Wilcoxon signed-rank test [86]. Table 20 shows the ALC scores, F-measures at the end of 120-minute annotation, and the statistical test P-value.

According to our results, *CAUSE2* significantly outperformed *Random* globally in terms of ALC scores for both users. However, the improvements of *CAUSE2* vs. *Random* are different between user 1 and user 2. For user 1, *CAUSE2* always performed better than *Random,* almost for the entire annotation process. For building an NER model with F-measure of 0.70, user 1 spent ~86 minutes in *CAUSE2* versus ~117 minutes in *Random,* indicating about ~31 minutes (26.5%) reduction in annotation time against *Random*. For user 2, the benefit of *CAUSE2* mostly showed in the early stage of annotation (i.e., the first 30 minutes). After that, both methods seemed to perform similarly.

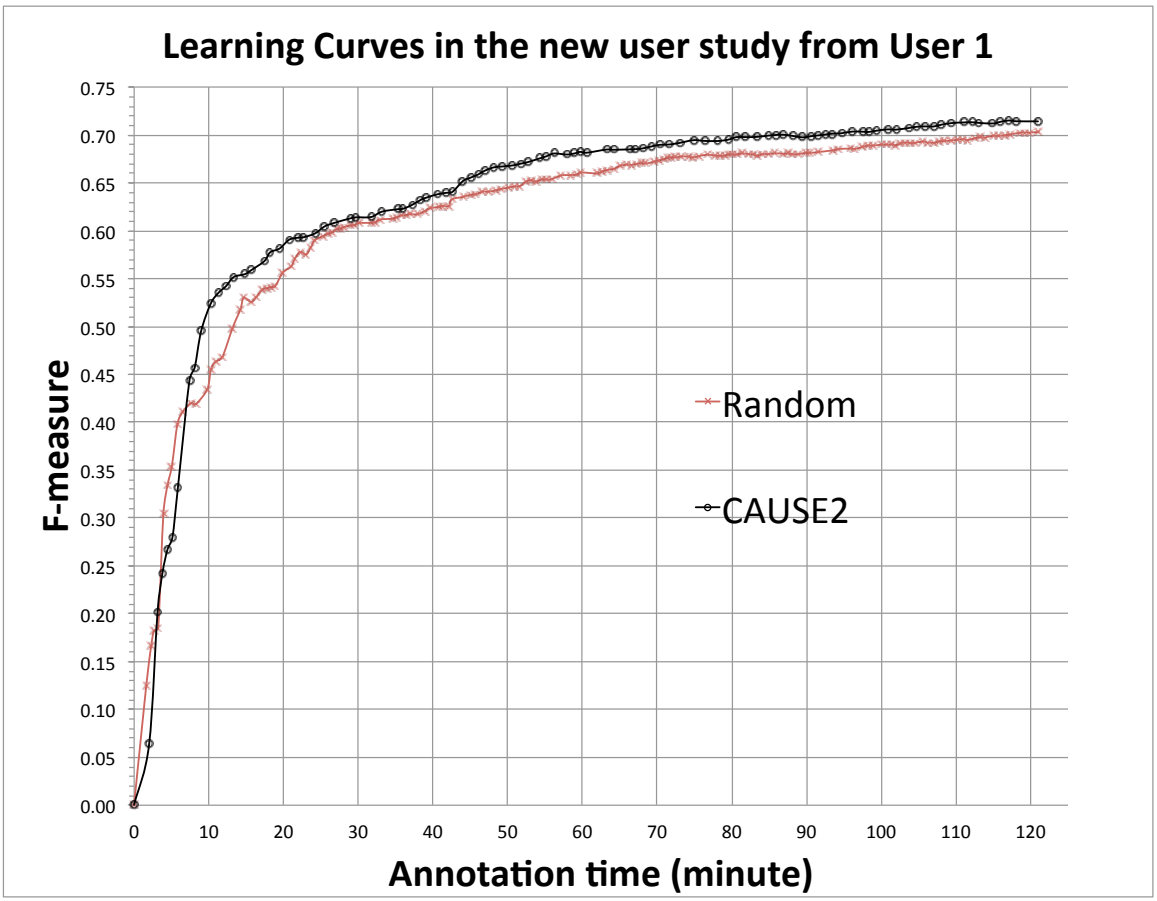

Figure 24. Learning curves by *Random* and *CAUSE2* from user 1 in the new user study

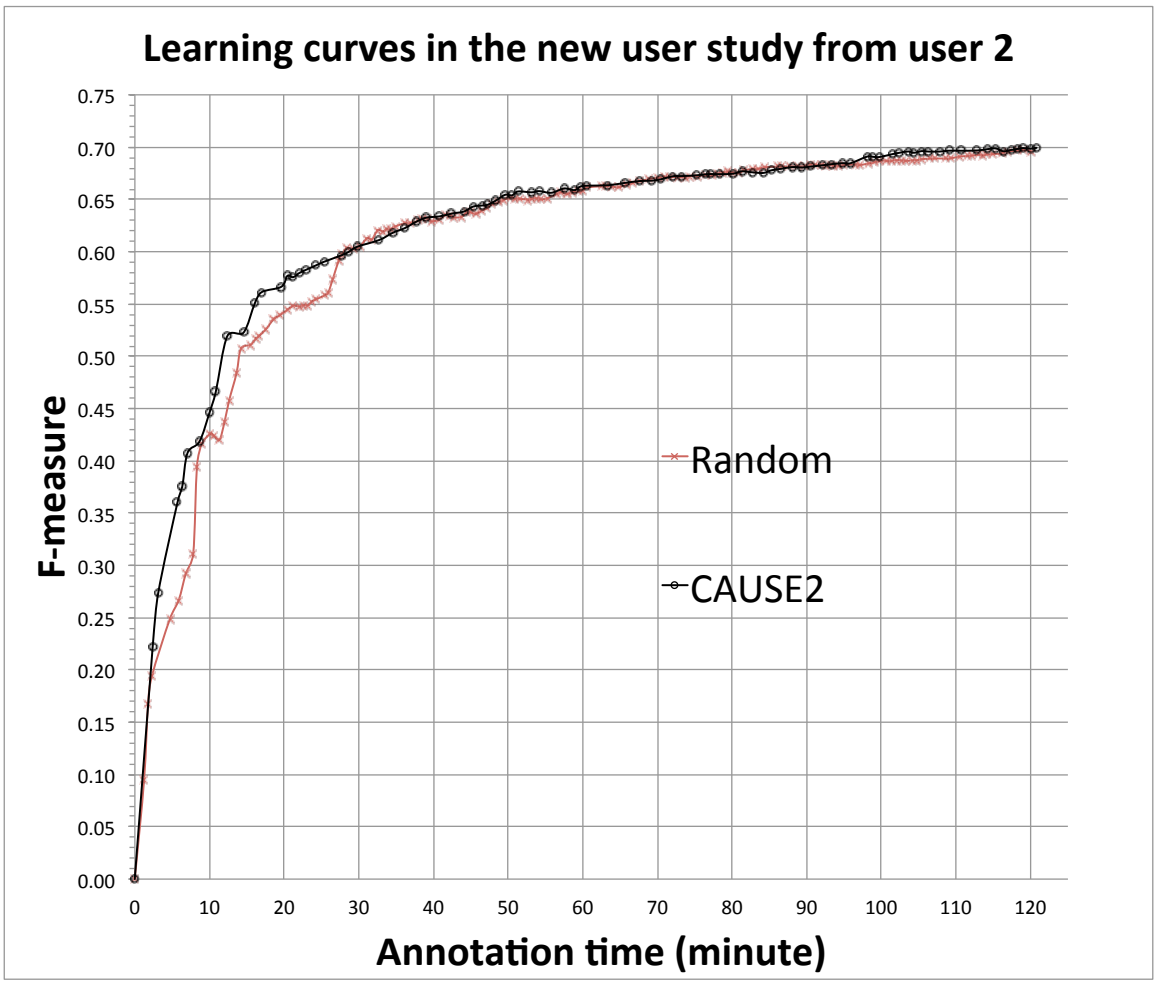

Figure 25. Learning curves by *Random* and *CAUSE2* from user 2 in the new user study

Table 20. ALC scores, F-measures at the end of 120-minute annotation, and the statistical test P-values of *CAUSE2* vs. *Random*

Users	Methods	ALC scores	F-measures at 120 minutes	P-values based on Wilcoxon signed-rank test
User 1	*Random*	0.833	0.703	6.4×10^{-5}
	CAUSE2	0.858	0.714	
User 2	*Random*	0.820	0.696	6.5×10^{-4}
	CAUSE2	0.841	0.700	

Meanwhile, we also simulated the new user study following the design in the simulation study (Section 2.3.2). Figure 26 and 27 show the simulated learning curves of *Random* and *CAUSE2* based on the cost models for user 1 and user 2, respectively. It seemed that the learning curves in Figure 26 (simulation) are similar to the learning curves in Figure 24 (user study), indicating that the simulation study is a feasible way to mimic the user study.

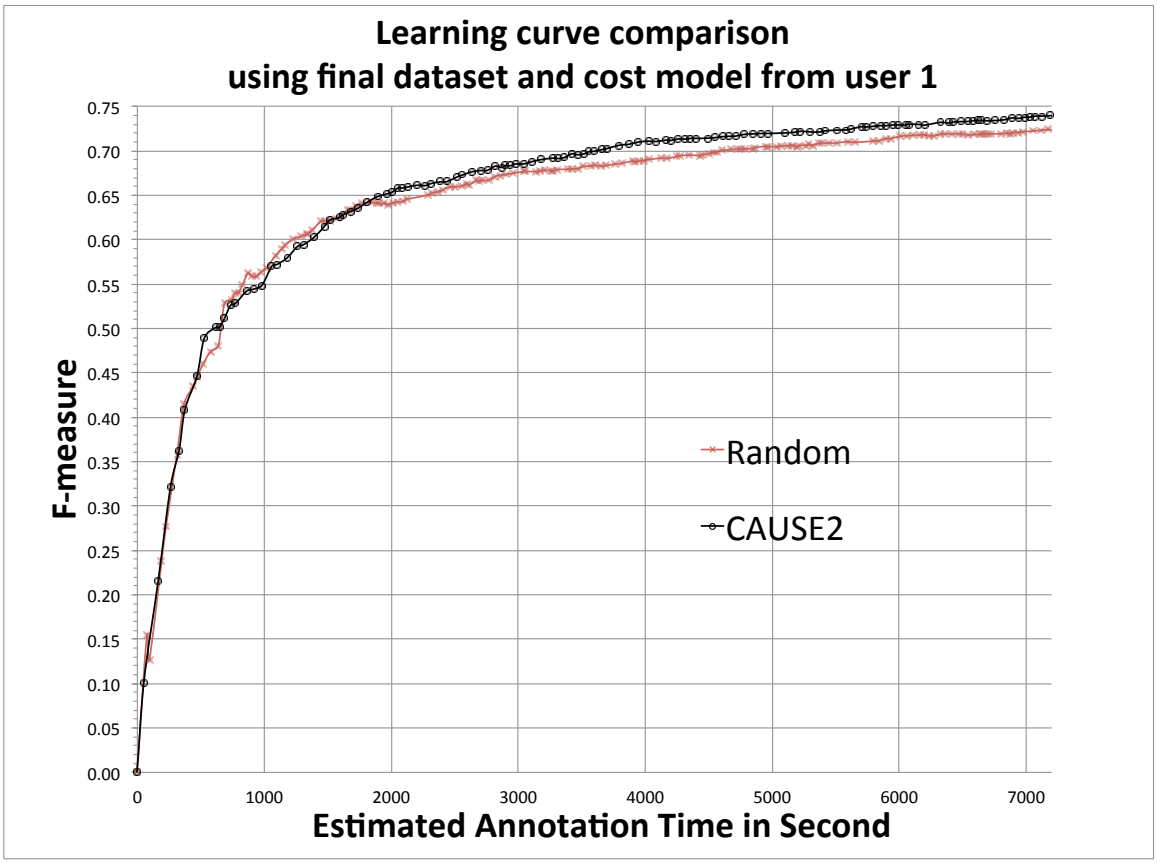

Figure 26. Simulated learning curves of *Random* and *CAUSE2* based on the cost models from user 1

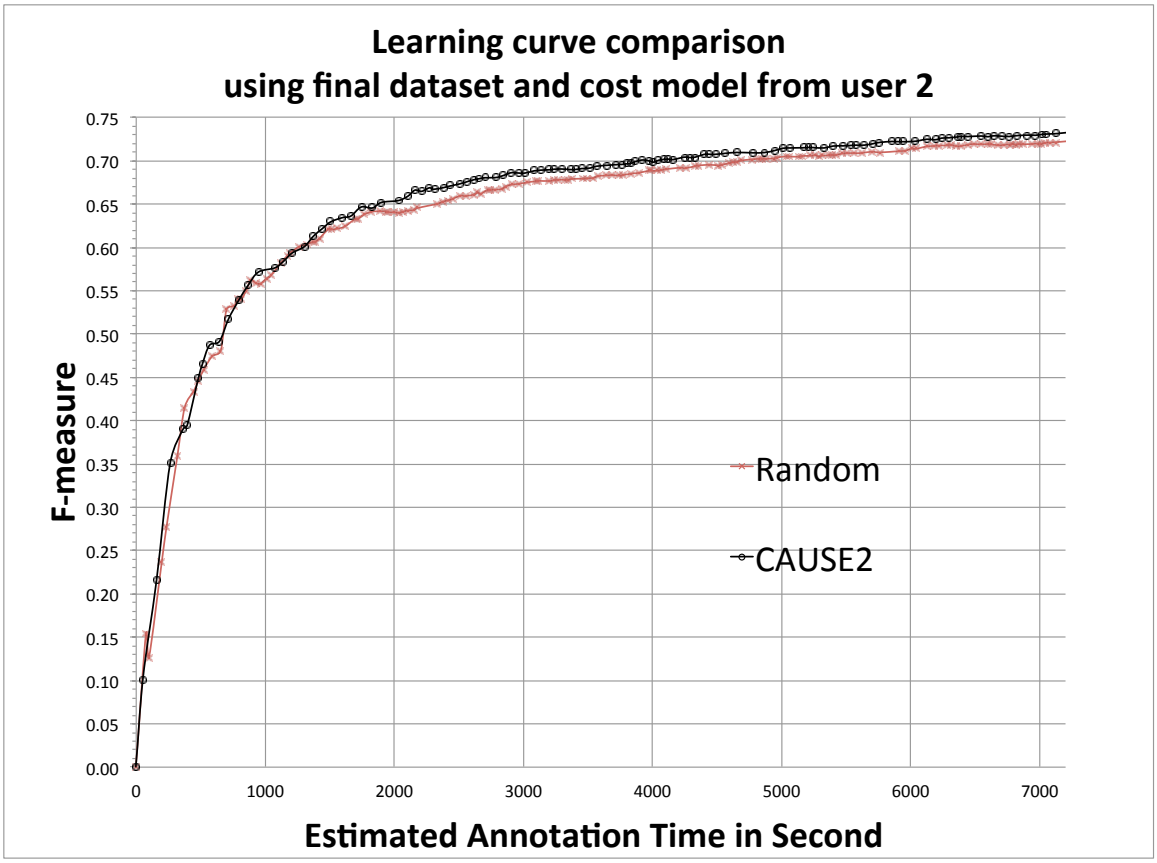

Figure 27. Simulated learning curves of *Random* and *CAUSE2* based on the cost models from user 2

Table 21 shows additional measurements of the annotation processes, such as annotation speed and quality. Both users performed faster (> 9 entities per minute) compared to the user study in Chapter 3. With respect to annotation quality, both users maintained at least the acceptable level of 0.80 in F-measure.

Table 21. Annotation quantity, speed, and quality comparison in the 120-minute main study for *Random* and *CAUSE2* from two users in the new user study

User	Method	Annotated Entity count	Annotation speed (Entities per min)	Annotation quality (F-measure)
User 1	*Random*	1,104	9.20	0.85
	CAUSE2	1,090	9.08	0.83
User 2	*Random*	1,087	9.06	0.84
	CAUSE2	1,183	9.86	0.80

We further analyzed users' annotation quality across sessions. Figure 28 shows the annotation quality at four 30-minute sessions for both users. Both users' annotation quality was relatively consistent for *Random* (user 1: 0.85 – 0.87; user 2: 0.83 – 0.85). However, in the user study for *CAUSE2*, both users had reduced annotation quality in the later sessions (User 1: 0.88 – 0.78; user 2: 0.88 – 0.73). A survey, presented in Appendix B, revealed that the reduced *annotation quality* could be due to the increased difficulty of samples in later sessions of *CAUSE2*. We also suspect that the decreased annotation quality also affects the benefit of *CAUSE2*. In particular, *CAUSE2* showed better performance only for the first 30 minutes for user 2, which could be related to the dropped annotation quality of user 2 in the later three sessions (e.g., below 0.8). This may also explain why the simulated results (Figure 27) are better than the user study results, as the simulation experiments were based on the gold standard annotation with annotation quality of 1.0 in F-measure.

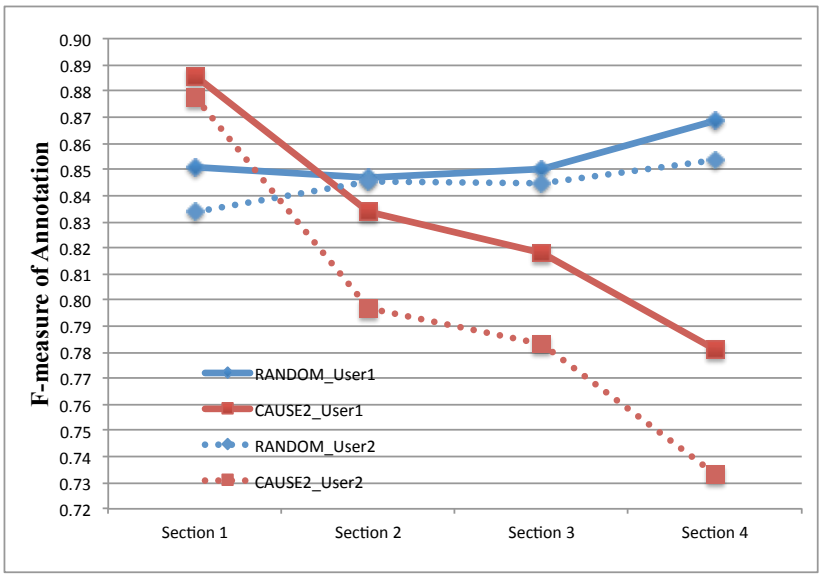

Figure 28. Annotation quality across different sessions for both *Random* and CAUSE and for both users

More characteristics about user annotation processes are reported in Table 22. These results of *CAUSE2* from both users are very close to the simulated values (13.12 words per sentence, 2.07 entities per sentence, 0.33 in entity density).

Table 22. Additional characteristics of annotation processes for both users for *Random* and *CAUSE* in each 120-minute annotation

User	Method	Sentences per min	Words per sentence	Words per min	Entities Per Sentence	Entity Density
User 1	*Random*	6.91	11.44	79.03	1.33	0.24
	CAUSE2	4.26	13.06	55.61	2.13	0.35
User 2	*Random*	6.61	11.52	76.14	1.37	0.25
	CAUSE2	3.85	13.40	51.60	2.56	0.37

4.4 Discussion

In this study, we integrated the annotation cost estimation models into the previously developed AL algorithms and demonstrated the utility of this approach using both simulation and user studies. To the best of our knowledge, *CAUSE2* is the first kind of AL algorithm that combines uncertainty, representativeness, and cost models to efficiently build NER systems for clinical text. Despite the success of the *CAUSE2* algorithm in this study, it is just a start. Many aspects of the user annotation process need to be further explored. Based on our current experiments, we conducted some additional analysis, hoping to provide some insights to important issues.

Differences between users: Our user study showed that the benefit of *CAUSE* was different between two users. In the results section, we have identified that annotation quality could be related to this finding. Another possible reason could be related to the quality of the annotation time models. As shown in section 3.1, our proposed annotation cost estimation model worked better on user 1 than that on user 2. In other words, it is more difficult to predict the annotation time by user 2 using the current model. Ideally, we hope to develop models that work effectively for most of the users, so that we do not need to develop different models for individual users. However, the results from only two users are insufficient to draw any conclusions. In the future, we plan to recruit more users for the user study.

Annotation time models: Another interesting research direction is to develop more sophisticated annotation time models, instead of linear regression models. The simple annotation time model proposed in this chapter, however, could be improved by capturing the following features also associated with the user's annotation time: (1) Proficiency in recognizing a medical concept: users could know one concept better than another, which results in different annotation time for

identifying different concepts. (2) Relation between medical concepts or non-concept phrases. More complicated relations between concepts could increase the time for users to make accurate decisions. (3) Context out of the modeling scope: there were cases where users would need to read the context outside a given sentence (i.e. the neighbors of the sentence and section headers) to better identify the concepts in the sentence.

Pros and cons of simulation studies: With an accurate annotation time model, we could evaluate a large number of methods and obtain results close to reality. Moreover, a simulation study is much more economical than a user study, which could be very costly or time consuming (sometimes even unfeasible) to evaluate a large number of methods. However, we need to be cautious that the simulated results based on estimated annotation cost do overestimate the benefit of AL. Two reasons may cause overestimation of benefit: (1) simulation is often based on gold standard; while a user study relies on annotations generated by users in real time. The annotation quality in the simulation is 100% versus 80-85% in the user study and (2) The simulated update process (e.g. querying -> annotation -> training -> querying…) is ideal, while the actual update process in the user study may not be optimal for reasons such as supporting the no-waiting annotation workflow.

AL for clinical NER in the long term: In this study, we limited the annotation time to 120 minutes, which is not long enough to show the long-term effect of AL methods. To evaluate the long-term performance of AL, we simulated both *CAUSE2* and *Random* sampling for up to 20 hours (10 times as long as the user study) for user 1. Figure 29 shows the simulated learning curves of *Random* and *CAUSE2* for 20 hours.

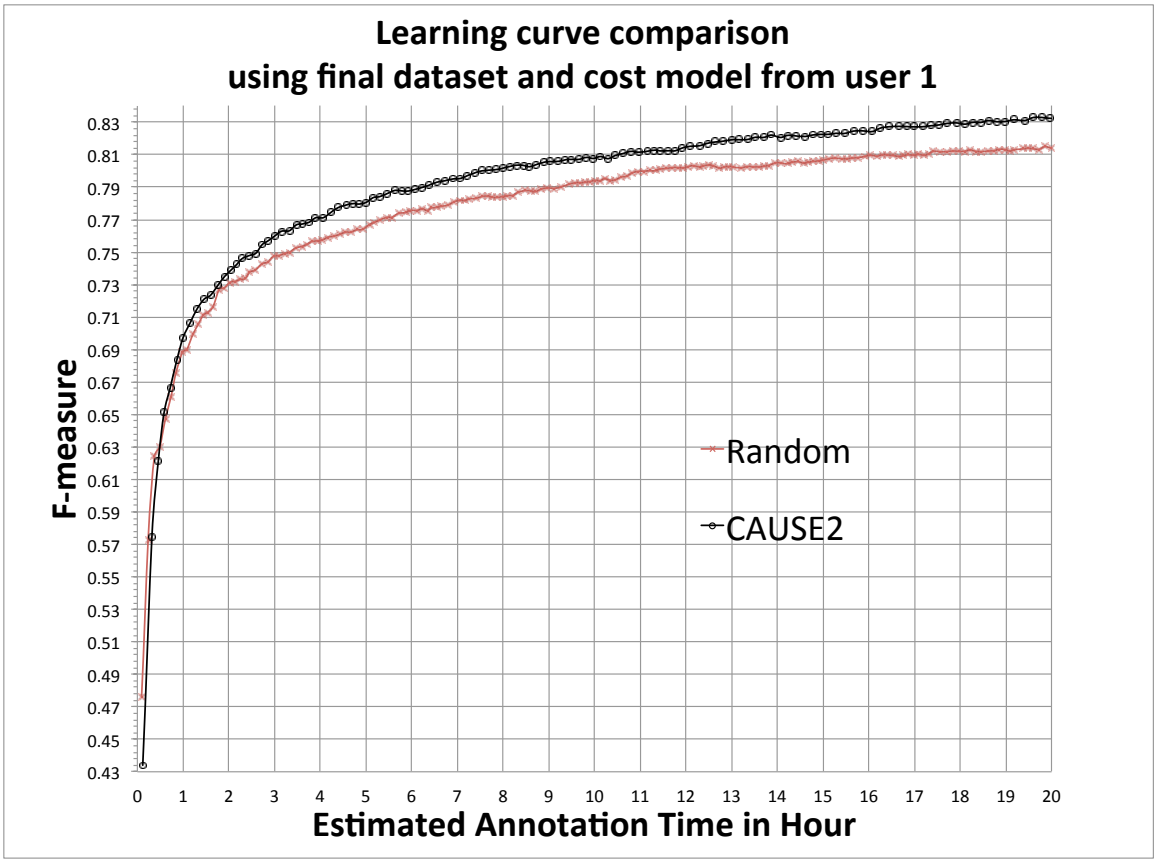

Figure 29. Simulated learning curves for *Random* and *CAUSE2* for 20 estimated hours of annotation time based on an annotation cost model for user 1

Based on the above figure, we further calculated the estimated annotation time for both methods at different F-measure levels (0.70 - 0.81) and reported the percentages of saving using *CAUSE2,* when compared to *Random* (see Table 23). Simulated results show that AL would achieve higher percentages of saving when we extend the annotation time, which is very promising. We plan to extend the user study to evaluate the long-term effect of AL for building clinical NER systems.

Table 23. Estimated annotation cost savings by *CAUSE2* at different F-measures

F-measure	**Estimated annotation time in hours**		**Annotation time reduction percentage**
	CAUSE2	*Random*	
0.81	10.62	17.10	37.89%
0.80	7.54	11.21	32.74%
0.79	6.38	9.26	31.10%
0.78	5.00	6.93	27.85%
0.77	3.90	5.46	28.57%
0.76	3.16	4.42	28.51%
0.75	2.59	3.33	22.22%
0.74	2.17	2.73	20.51%
0.73	1.76	2.03	13.30%
0.72	1.46	1.60	8.75%
0.71	1.31	1.44	9.03%
0.70	1.15	1.22	5.74%

CHAPTER 5

Conclusion

5.1 Summary of key findings

In this dissertation research, we systematically studied AL for a clinical NER task. We summarize the key findings from each chapter in the following paragraphs.

In Chapter 1, we introduced AL as a possible solution to develop clinical NLP systems in a more effective way. A survey was conducted to show the impact of AL in biomedical text processing. Most of the studies provided evidence supporting the promises of AL in annotation cost reduction for different tasks. However, we have concerns about their assumption of equal annotation cost per sample, which is not true for most of the annotation tasks. We also studied the literature on AL in a practical setting, which suggested that AL without the appropriate consideration of annotation cost could be no different from random sampling.

In Chapter 2, we conducted a preliminary study to examine multiple existing and novel AL algorithms for a clinical NER task. The preliminary results showed that uncertainty sampling based algorithms outperformed diversity based sampling methods and random sampling under two evaluation assumptions: (1) same cost per sentence, and (2) same cost per word. However, we found that under the second evaluation assumption that is intuitively closer to the real world, the advantage of uncertainty sampling against random sampling was smaller, as compared to the first evaluation assumption. Therefore, we postulated that AL needs to be further evaluated in the real-world setting.

In Chapter 3, we developed an AL enabled annotation system for NER (*Active LEARNER*), which allows us to assess the practicality of AL in a user study. Meanwhile, we proposed a clustering and uncertainty sampling engine (*CAUSE*), which considers both the representativeness and the uncertainty of sentences. The results in the user study from the two nurses showed a mixed outcome. For one user, AL helped save annotation time compared to random sampling. For another, however, random sampling achieved better performance than AL. Moreover, we discovered that human factors, such as annotation speed and annotation quality, also affected the user study results.

In Chapter 4, we proposed an annotation cost modeling formula, which was integrated in the *CAUSE* model to query the sentences that are most informative per estimated cost. This cost-conscious model, *CAUSE2*, was compared with *CAUSE*, uncertainty sampling, and random sampling in the simulation using estimated annotation time as cost. Furthermore, we conducted a new user study to evaluate *CAUSE2* and random sampling. The results from two users showed that *CAUSE2* globally outperformed random sampling. To achieve an NER model with 0.70 in F-measure, *CAUSE2* reduced the annotation time of 26.5% compared to random sampling for one user. However, the advantage of AL was not obvious for another user. By further analyzing the results with the users' annotation data, we found that annotation quality and annotation cost model mainly caused the difference in benefit of AL for two users. We suggested that a larger number of users are needed to test whether AL could work effectively for most users in the clinical NER task.

In this chapter, we summarized our contributions from the dissertation research, further analyzed some limitations in the study, and describe potential future directions for the work.

5.2 Innovations and contributions

5.2.1 Innovations

We propose a novel AL algorithm (*CAUSE*) that combines uncertainty sampling and clustering in an innovative way. The conventional hybrid method combines uncertainty and diversity sampling by assigning sentences overall scores, which are a function of their uncertainty and diversity (e.g. summation of the weighted scores of both uncertainty and diversity) [66, 67]. The *CAUSE*, different from the conventional method, implements a novel two-layer sampling strategy, which performs cluster sampling to rank clusters followed by representative sampling to find the most representative sample for the top ranked cluster.

We further improve the *CAUSE* model to *CAUSE2,* which queries the most informative and least costly sentences based on their three properties: uncertainty, representativeness, and annotation time. AL with real annotation cost has been empirically studied [60], but our annotation cost model is new for the clinical NER task with an interpretable formula for the annotation process.

To the best of our knowledge, this is the first study on building practical AL systems for clinical NER. This is also the first study on evaluating the AL methods for the real-time clinical NER in a user study. Previous studies of AL with actual annotation cost for NER were not actually based on the real-time setting [60, 61, 87, 88]. They collected the annotation time for a set of randomly selected samples and then performed simulation on these annotated samples. However, they ignored the human factors during the annotation process. In our study, we truly evaluated AL in a user study with consideration of all factors in the annotation. We discovered that human factors created the variance in the AL results and AL with better modeling on human factors could improve the actual performance.

5.2.2 Contributions

Our studies have contributed in the areas of biomedical informatics, biomedical NLP, computer sciences, and healthcare. With respect to biomedical informatics, we conducted empirical studies of AL in clinical NER. Over twenty unique querying methods were systematically evaluated using both theoretical and practical measurements. Moreover, our study demonstrated an informatics tool that truly interacts with medical domain experts, and is one of the newest applications that cross biomedical research and information technology. From the user study, we obtained valuable experience to enhance the practical usage of biomedical NLP technology. Our novel methods, based on state-of-the-art computer science techniques, are generalizable for text processing tasks in open domains. They contributed to the development of interactive machine learning and NLP technologies in computer science. With reference to healthcare, our system could enhance the efficiency in building clinical NER systems, thus facilitating clinical research that uses EHR data. Our novel AL paradigm and system could be one of the big data analytic solutions in healthcare.

5.3 Limitations and future work

The user study results are only based on two users, which is not sufficient to draw conclusions about the effectiveness of AL in practice. From the perspective of evaluation, two querying methods may not be comparable when users performed very differently in annotation speed and annotation quality. Although the annotation training was designed to improve the consistency of

users' annotation across different methods, the method evaluated later in the user study very likely gained advantage from the prior experiments since the users' annotation performance generally increased over time.

The annotation cost models we proposed do not consider the differences in annotation difficulty across different sentences, which could be an important variable to improve the estimation of annotation time. Moreover, the user-specific modeling process may require a significant effort in annotation data collection. The personalized annotation cost models may not be consistent over time. In addition, the performance of our methods also relied on the quality of clustering results. We did not apply a systematic parameter tuning strategy to find an optimal parameter setting (e.g. optimal number of semantic topics and clusters). The clustering results were not quantitatively evaluated and optimized in our study.

In the future, we plan to extend the user study involving a larger number of expert users involved. A standardized user study is needed for a larger number of users. To reduce the influence of human factors, we plan to develop a novel integrated system that merges all querying methods and evaluates them at the same time in the user study. In the annotation cost model, we will add more predictive variables, such as annotation time and annotation difficulty based on the document frequency of concepts and the complexity of syntactic structures of sentences. A more general annotation model is also needed to compare to the personalized model used in this study. To investigate the human factor influence on active learning results in the economical simulation study, we would design experiments by adding noise on both estimated annotation time and annotation quality. Moreover, we also plan to improve the querying methods by designing better heuristics to optimize both the informativeness and the annotation cost of sentences. In terms of interface improvement, we will enhance user's annotation experience by

adding interactive components (e.g. annotation recommendations) to increase user's annotation speed and annotation quality.

5.4 Conclusion

In this dissertation research, we systematically studied AL in a clinical NER task. We built a practical AL algorithm to query the most informative and least costly sentences. The results showed AL has the potential to save annotation time and improve model quality for building ML-based NER systems. To the best of our knowledge, this is the first study on building practical AL systems for clinical NER, providing a new direction of AL development in practice for clinical research.

Appendix A. Wilcoxon Signed-rank Test to Evaluate Difference Learning Curves

First, we generated the smooth learning curves, which plot the F-measures at every 5-minute time stamp. The F-measure point at a 5-minute time stamp (x) is the average F-measure of the actual F-measure points between the current time stamp (x) and the previous time stamp (x – 1). There are 24 points in each smoothed learning curve presenting the F-measures from 0 to 120 minutes of annotation. Figure 30 and 31 demonstrate the smoothed learning curves from user 1 and user 2, respectively, in the new user study.

As every method has an F-measure point at the same time stamp, the learning curves from different methods are comparable. Then we performed a statistical analysis based on Wilcoxon signed-rank test for the hypothesis that two methods have statistically significant difference in learning curves. Table 24 shows the Wilcoxon signed-rank test results.

Table 24. Wilcoxon signed-rank test based on the smooth learning curves by *Random* and *CAUSE2* from user 1 and user 2

Users	Methods	P-values based on Wilcoxon signed-rank test	Signed rank
User 1	*Random*	6.4×10^{-5}	22
	CAUSE2		
User 2	*Random*	6.5×10^{-4}	37
	CAUSE2		

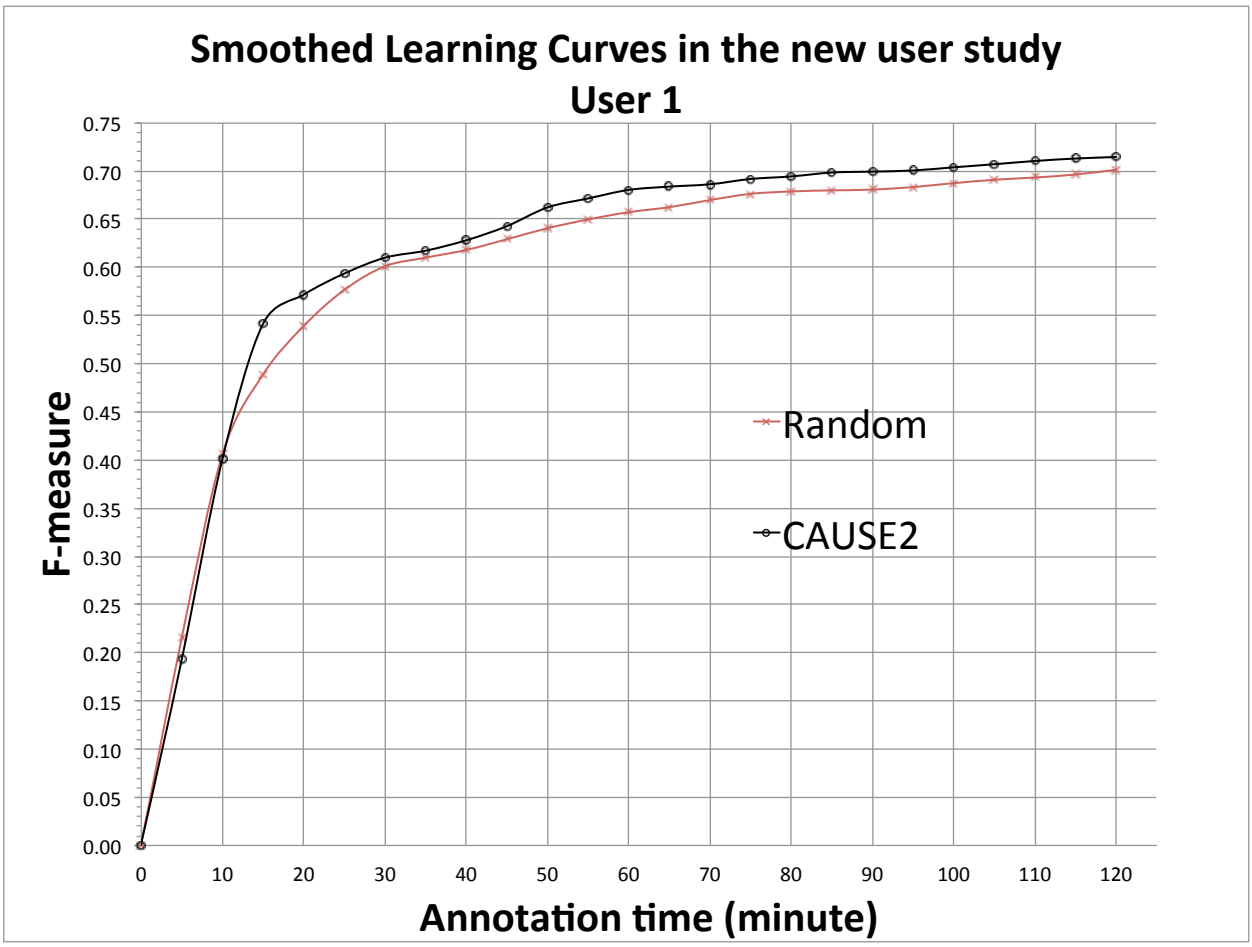

Figure 30. Smoothed learning curves of *Random* and *CAUSE2* from user 1 in the new user study

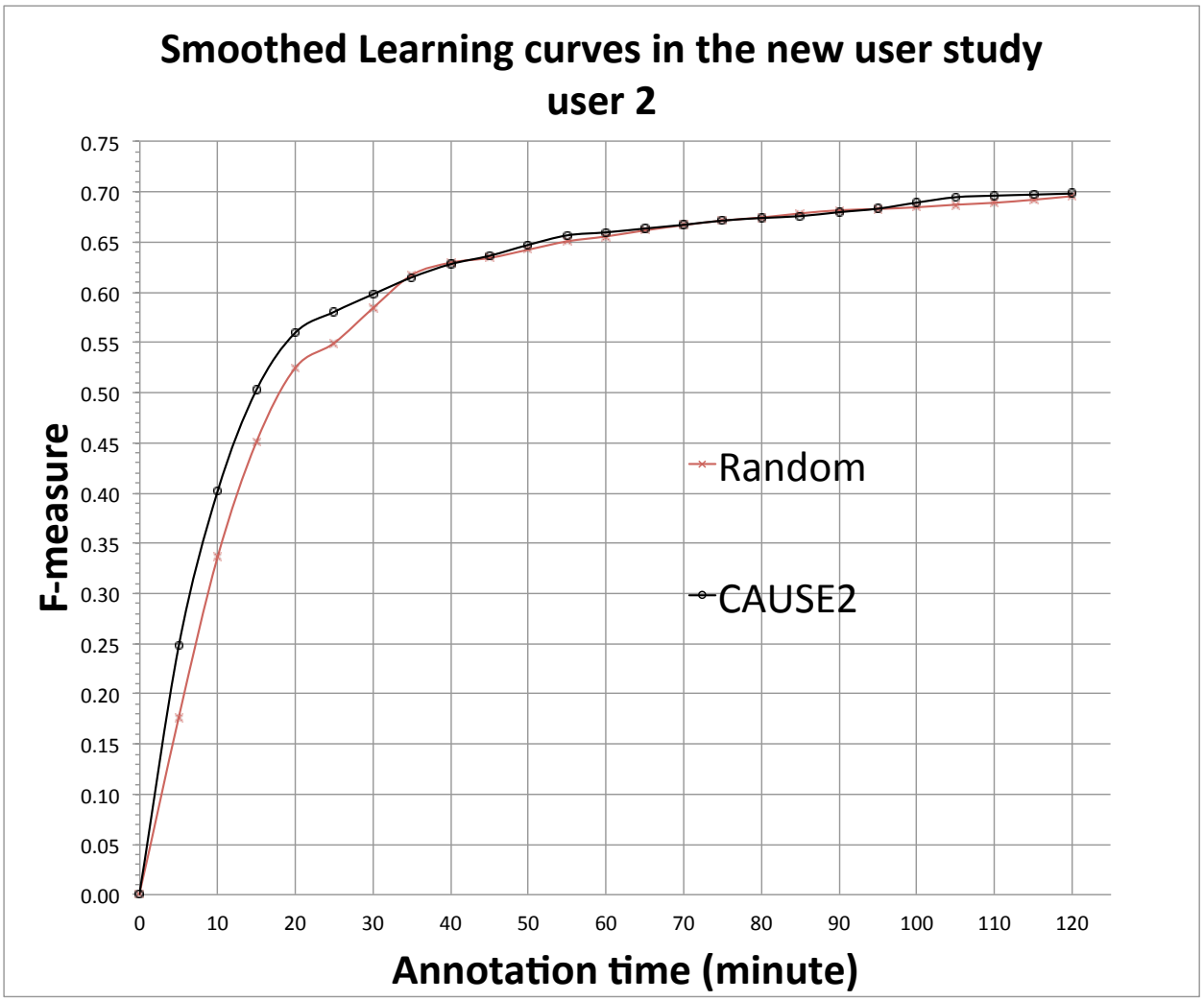

Figure 31. Smoothed learning curves of *Random* and *CAUSE2* from user 2 in the new user study

Appendix B. Survey from Two Users

User 1:

Categories	Description	Score		Comments
		Week 6	Week 7	
Average Length of sentence	Please score your impression on the average length of the set of sentences given in each week. Note: Higher score represents your impression that the sentences were longer. We preset score of 3 for Week 6 as the reference. You may score 3 for the sentences in week 7 if their average length was very similar to the one in week 6; or score 1 or 2 if it was lower than week 6; or score 4 or 5 if it was higher than week 6.	**1 2 3 4 5**	**1 2 3 4 5**	**No change**
Medical concept density per sentence	Please score your impression on the average density of medical concepts from the sentences given in each week. Note: Higher score represents your impression that medical concepts in a sentence were denser (or the ratio of the words that you tagged as part of the medical concept in a sentence was higher). We preset score of 3 for Week 6 as the reference. You may score 3 for the concept density in week 7 if it was very similar to week 6; or score 1 or 2 if it was lower than week 6; or score 4 or 5 if it was higher than week 6.	**1 2 3 4 5**	**1 2 3 4 5**	**No change**
Annotation difficulty	Please score your impression on the annotation difficulty in annotating the sentences given in each week. Note: Higher score represents your impression that the sentences were more difficult to annotate. We preset score of 3 for week 6 as the reference. You may score 3 for the annotation difficulty in week 7 if it was very similar to week 6; or score 1 or 2 if it was less difficult than week 6; or score 4 or 5 if it was more difficult than week 6.	**1 2 3 4 5**	**1 2 3 4 5**	**Harder**

Annotation difficulty change over four 30-minute sections	Please score your impression on the change of annotation difficulty over section 1, 2, 3, and 4 for the sentences given in each week. Note: Score 5 represents your impression that the annotation was more and more difficult in the later sections; score 3 represents your impression that the annotation difficulty was very similar in each section; score 1 represents your impression that the annotation was less and less difficult in the later sections; score 4 represents your impression that the annotation seems more difficult over sections but the change was not very obvious; score 2 represents your impression that the annotation seems less difficult over sections but the change was not very obvious.	**1 2 3 4 5**	**1 2 3 4 5**	Please also comment on what probably was changing over sections others than annotation difficulty if there is any.
Clinical relevancy of sentences	Please score your impression on the clinical relevance of the sentences given in each week. Note: Higher score represents your impression that the sentences contained more clinically relevant information. We preset score of 3 for week 6 as the reference. You may score 3 for the clinical relevancy in week 7 if it was very similar to week 6; or score 1 or 2 if it was less than week 6; or score 4 or 5 if it was more than week 6.	**1 2 3 4 5**	**1 2 3 4 5**	**More relevant sentences**
Diversity of sentences	Please score your impression on the diversity of sentences given in each week. Note: Higher score represents your impression that the sentences contained more diverse or less duplicate content topics. We present score of 3 for week 6 as the reference. You may score 3 for the sentence diversity in week 7 if it was very similar to week 6; or score 1 or 2 if it was less diverse than week 6; or score 4 or 5 if it was more diverse than week 6.	**1 2 3 4 5**	**1 2 3 4 5**	**Same**

Please also comment on your impression on the difference of user study between week 6 and 7 others than the listed categories if there are any.

Comment: There is no significant difference in length, clinical relevance, and diversity. However, week 7 seems to be a little more difficult in terms of annotation difficulty. The phrases were more ambiguous and the gold standard was inconsistent. Also, the clinical notes in this corpus seemed to have more abbreviations and were written in a "lazier" way than the previous weeks.

User 2:

Categories	Description	Score		Comments
		Week 6	Week 7	
Average Length of sentence	Please score your impression on the average length of the set of sentences given in each week. Note: Higher score represents your impression that the sentences were longer. We preset score of 3 for Week 6 as the reference. You may score 3 for the sentences in week 7 if their average length was very similar to the one in week 6; or score 1 or 2 if it was lower than week 6; or score 4 or 5 if it was higher than week 6.	**1 2 3 4 5**	**1 2 3 4 5**	
Medical concept density per sentence	Please score your impression on the average density of medical concepts from the sentences given in each week. Note: Higher score represents your impression that medical concepts in a sentence were denser (or the ratio of the words that you tagged as part of the medical concept in a sentence was higher). We preset score of 3 for Week 6 as the reference. You may score 3 for the concept density in week 7 if it was very similar to week 6; or score 1 or 2 if it was lower than week 6; or score 4 or 5 if it was higher than week 6.	**1 2 3 4 5**	**1 2 3 4 5**	
Annotation difficulty	Please score your impression on the annotation difficulty in annotating the sentences given in each week. Note: Higher score represents your impression that the sentences were more difficult to annotate. We preset score of 3 for week 6 as the reference. You may score 3 for the annotation difficulty in week 7 if it was very similar to week 6; or score 1 or 2 if it was less difficult than week 6; or score 4 or 5 if it was more difficult than week 6.	**1 2 3 4 5**	**1 2 3 4 5**	

Annotation difficulty change over four 30-minute sections	Please score your impression on the change of annotation difficulty over section 1, 2, 3, and 4 for the sentences given in each week. Note: Score 5 represents your impression that the annotation was more and more difficult in the later sections; score 3 represents your impression that the annotation difficulty was very similar in each section; score 1 represents your impression that the annotation was less and less difficult in the later sections; score 4 represents your impression that the annotation seems more difficult over sections but the change was not very obvious; score 2 represents your impression that the annotation seems less difficult over sections but the change was not very obvious.	**1 2 3 4 5**	**1 2 3 4 5**	Please also comment on what probably was changing over sections others than annotation difficulty if there is any.
Clinical relevancy of sentences	Please score your impression on the clinical relevance of the sentences given in each week. Note: Higher score represents your impression that the sentences contained more clinically relevant information. We preset score of 3 for week 6 as the reference. You may score 3 for the clinical relevancy in week 7 if it was very similar to week 6; or score 1 or 2 if it was less than week 6; or score 4 or 5 if it was more than week 6.	**1 2 3 4 5**	**1 2 3 4 5**	
Diversity of sentences	Please score your impression on the diversity of sentences given in each week. Note: Higher score represents your impression that the sentences contained more diverse or less duplicate content topics. We present score of 3 for week 6 as the reference. You may score 3 for the sentence diversity in week 7 if it was very similar to week 6; or score 1 or 2 if it was less diverse than week 6; or score 4 or 5 if it was more diverse than week 6.	**1 2 3 4 5**	**1 2 3 4 5**	

Please also comment on your impression on the difference of user study between week 6 and 7 others than the listed categories if there are any.

REFERENCES

[1] Chen Y, Mani S, Xu H. Applying active learning to assertion classification of concepts in clinical text. Journal of biomedical informatics. 2012;45:265-72.

[2] Chen Y, Cao H, Mei Q, Zheng K, Xu H. Applying active learning to supervised word sense disambiguation in MEDLINE. J Am Med Inform Assoc. 2013.

[3] Chen Y, Carroll RJ, Hinz ER, Shah A, Eyler AE, Denny JC, et al. Applying active learning to high-throughput phenotyping algorithms for electronic health records data. J Am Med Inform Assoc. 2013;20:e253-9.

[4] Meystre Sm SGKK-SKCHJF. Extracting information from textual documents in the electronic health record: a review of recent research

Yearb Med Inform. Yearb Med Inform. 2008.

[5] Demner-Fushman D, Chapman WW, McDonald CJ. What can natural language processing do for clinical decision support? Journal of biomedical informatics. 2009;42:760-72.

[6] Kho AN, Pacheco JA, Peissig PL, Rasmussen L, Newton KM, Weston N, et al. Electronic medical records for genetic research: results of the eMERGE consortium. Science translational medicine. 2011;3:79re1.

[7] Friedman C. Towards a comprehensive medical language processing system: methods and issues. Proceedings : a conference of the American Medical Informatics Association / AMIA Annual Fall Symposium AMIA Fall Symposium. 1997:595-9.

[8] Ferrucci DA. IBM's Watson/DeepQA. SIGARCH Comput Archit News. 2011;39.

[9] Meystre SM, Savova GK, Kipper-Schuler KC, Hurdle JF. Extracting information from textual documents in the electronic health record: a review of recent research. Yearb Med Inform. 2008:128-44.

[10] Gottesman O, Kuivaniemi H, Tromp G, Faucett WA, Li R, Manolio TA, et al. The Electronic Medical Records and Genomics (eMERGE) Network: past, present, and future. Genet Med. 2013;15:761-71.

[11] Xu H, Fu Z, Shah A, Chen Y, Peterson NB, Chen Q, et al. Extracting and integrating data from entire electronic health records for detecting colorectal cancer cases. AMIA Annu Symp Proc. 2011;2011:1564-72.

[12] Xu H, Aldrich MC, Chen Q, Liu H, Peterson NB, Dai Q, et al. Validating drug repurposing signals using electronic health records: a case study of metformin associated with reduced cancer mortality. J Am Med Inform Assoc. 2015;22:179-91.

[13] Uzuner O, Solti I, Cadag E. Extracting medication information from clinical text. J Am Med Inform Assoc. 2010;17:514-8.

[14] Uzuner O, South BR, Shen S, DuVall SL. 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. J Am Med Inform Assoc. 2011;18:552-6.

[15] Sun W, Rumshisky A, Uzuner O. Evaluating temporal relations in clinical text: 2012 i2b2 Challenge. J Am Med Inform Assoc. 2013;20:806-13.

[16] Hripcsak G, Friedman C, Alderson PO, DuMouchel W, Johnson SB, Clayton PD. Unlocking clinical data from narrative reports: a study of natural language processing. Ann Intern Med. 1995;122:681-8.

[17] Friedman C, Shagina L, Lussier Y, Hripcsak G. Automated encoding of clinical documents based on natural language processing. J Am Med Inform Assoc. 2004;11:392-402.

[18] Aronson AR, Lang FM. An overview of MetaMap: historical perspective and recent advances. J Am Med Inform Assoc. 2010;17:229-36.

[19] Savova GK, Masanz JJ, Ogren PV, Zheng J, Sohn S, Kipper-Schuler KC, et al. Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. J Am Med Inform Assoc. 2010;17:507-13.

[20] Denny JC, Miller RA, Johnson KB, Spickard A, 3rd. Development and evaluation of a clinical note section header terminology. AMIA Annu Symp Proc. 2008:156-60.

[21] NIH. Unified Medical Language System (UMLS), http://www.nlm.nih.gov/research/umls/.

[22] Jain NL KC, Friedman C, Hripcsak G. Identification of suspected tuberculosis patients based on natural language processing of chest radiograph reports. Proc AMIA Annu Falls Symp. 1996.

[23] Jain NL FC. Identification of findings suspicious for breast cancer based on natural language processing of mammogram reports. Proc AMIA Annu Falls Symp. 1997.

[24] Melton GB HG. Automated detection of adverse events using natural language processing of discharge summaries. J Am Med Inform Assoc. 2005.

[25] Friedman C SL, Lussier Y, Hripcsak G. Automated encoding of clinical documents based on natural language processing. J Am Med Inform Assoc. 2004.

[26] Xu H AK, Grann VR, Friedman C. Facilitating cancer research using natural language processing of pathology reports. Medinfo 2004.

[27] Denny JC SJ, Miller RA . Spickard III A. Understanding medical school curriculum content using KnowledgeMap. J Am Med Inform Assoc. 2003.

[28] Denny JC SA, Miller RA, et al. Identifying UMLS concepts from ECG impressions using KnowledgeMap. AMIA Annu Symp Proc. 2005.

[29] Denny JC, Spickard A, 3rd, Johnson KB, Peterson NB, Peterson JF, Miller RA. Evaluation of a method to identify and categorize section headers in clinical documents. J Am Med Inform Assoc. 2009;16:806-15.

[30] Mitchell KJ, Becich MJ, Berman JJ, Chapman WW, Gilbertson J, Gupta D, et al. Implementation and evaluation of a negation tagger in a pipeline-based system for information extract from pathology reports. Studies in health technology and informatics. 2004;107:663-7.

[31] Harkema H, Dowling JN, Thornblade T, Chapman WW. ConText: an algorithm for determining negation, experiencer, and temporal status from clinical reports. Journal of biomedical informatics. 2009;42:839-51.

[32] Xu H, Stenner SP, Doan S, Johnson KB, Waitman LR, Denny JC. MedEx: a medication information extraction system for clinical narratives. J Am Med Inform Assoc. 2010;17:19-24.

[33] Li Q, Zhai H, Deleger L, Lingren T, Kaiser M, Stoutenborough L, et al. A sequence labeling approach to link medications and their attributes in clinical notes and clinical trial announcements for information extraction. J Am Med Inform Assoc. 2013;20:915-21.

[34] Tang B, Wu Y, Jiang M, Chen Y, Denny JC, Xu H. A hybrid system for temporal information extraction from clinical text. J Am Med Inform Assoc. 2013;20:828-35.

[35] Wu ST, Juhn YJ, Sohn S, Liu H. Patient-level temporal aggregation for text-based asthma status ascertainment. J Am Med Inform Assoc. 2014.

[36] Patrick J, Li M. High accuracy information extraction of medication information from clinical notes: 2009 i2b2 medication extraction challenge. J Am Med Inform Assoc. 2010;17:524-7.

[37] de Bruijn B, Cherry C, Kiritchenko S, Martin J, 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;18:557-62.

[38] Jiang M, Chen Y, Liu M, Rosenbloom ST, Mani S, Denny JC, et al. A study of machine-learning-based approaches to extract clinical entities and their assertions from discharge summaries. J Am Med Inform Assoc. 2011;18:601-6.

[39] Lafferty J, McCallum A, Pereira F. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proc of the 18th International Conf on Machine Learning, Williamstown, MA, Morgan Kaufmann. 2001:282-9.

[40] B. Taskar CG, Koller D. Max-margin Markov networks. 2003.

[41] I. Tsochantaridis TJTH, Altun Y. Large margin methods for structured and interdependent output variables. Journal of Machine Learning Research. 2005.

[42] B. Tang HCYWMJ, Xu H. Clinical entity recognition using structural support vector machines with rich features. in Proceedings of the ACM sixth international workshop on Data and text mining in biomedical informatics. 2012.

[43] B. Tang HCYWMJ, Xu H. Recognizing clinical entities in hospital discharge summaries using Structural Support Vector Machines with word representation features. BMC Med Inform Decis Mak. 2013.

[44] M. Jiang YCMLSTRSMJCD, Xu H. A study of machine-learning-based approaches to extract clinical entities and their assertions from discharge summaries. J Am Med Inform Assoc. 2011.

[45] Lewis DD, Gale WA. A sequential algorithm for training text classifiers. Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval. 1994:3-12.

[46] Zhu J, Hovy E. Active Learning for Word Sense Disambiguation with Methods for Addressing the Class Imbalance Problem. Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. 2007:783-90.

[47] Tong S, Koller D. Support vector machine active learning with applications to text classification. Journal of Machine Learning Research. 2002;2:45-66.

[48] Settles B, Craven M. An analysis of active learning strategies for sequence labeling tasks. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP). 2008:1069-78.

[49] Figueroa RL, Zeng-Treitler Q, Ngo LH, Goryachev S, Wiechmann EP. Active learning for clinical text classification: is it better than random sampling? J Am Med Inform Assoc. 2012;19:809-16.

[50] D. Lewis JC. Heterogeneous uncertainty sampling for supervised learning. In: Kaufmann M, editor. Proceedings of the Eleventh International Conference on Machine Learning1994. p. pages 148–56.

[51] H.S. Seung MO, and H. Sompolinsky. Query by committee. Proceedings of the ACM Workshop on Computational Learning Theory1992. p. pages 287–94.

[52] B. Settles MC, and S. Ray. Multiple-instance active learning. Advances in Neural Information Processing Systems (NIPS): MIT Press; 2008. p. pages 1289–96.

[53] Chaloner K, Verdinelli I. Bayesian experimental design: A review. Stat Sci. 1995;10:273-304.

[54] McCallum NRaA. Toward optimal active learning through sampling estimation of error reduction. Proceedings of the International Conference on Machine Learning (ICML): Morgan Kaufmann; 2001. p. pages 441–8.

[55] Dagan I, Engelson PS. Committee-based sampling for training probabilistic classifiers. In Proceedings of the International Conference on Machine Learning (ICML): Morgan Kaufmann; 1995. p. pages 150–7.

[56] Nigam AMaK. Employing EM in pool-based active learning for text classification. Proceedings of the International Conference on Machine Learning (ICML): Morgan Kaufmann; 1998. p. pages 359–67.

[57] Settles B. Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin-Madison. 2009.

[58] Z. H. A theory of language and information: a mathematical

approach. Oxford: Clarendon Press;. 1991.

[59] Settles B. Closing the loop: fast, interactive semi-supervised annotation with queries on features and instances. Proceedings of the Conference on Empirical Methods in Natural Language Processing. Edinburgh, United Kingdom: Association for Computational Linguistics; 2011. p. 1467-78.

[60] Settles B, Craven M, Friedland L. Active learning with real annotation costs. In Proceedings of the NIPS Workshop on Cost-Sensitive Learning2008. p. pages 1-10.

[61] Haertel RA, Seppi KD, Ringger EK, Carroll JL. Return on investment for active learning. In Proceedings of the Neural Information Processing Systems Workshop on Cost Sensitive Learning2008.

[62] Baldridge J, Palmer A. How well does active learning actually work?: Time-based evaluation of cost-reduction strategies for language documentation. Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1. Singapore: Association for Computational Linguistics; 2009. p. 296-305.

[63] Figueroa RL, Zeng-Treitler Q, Ngo LH, Goryachev S, Wiechmann EP. Active learning for clinical text classification: is it better than random sampling? J Am Med Inform Assoc. 2012.

[64] Miller TA, Dligach D, Savova GK. Active learning for coreference resolution. Proceedings of the 2012 Workshop on Biomedical Natural Language Processing. Montreal, Canada: Association for Computational Linguistics; 2012. p. 73-81.

[65] Wallace BC, Small K, Brodley CE, Trikalinos TA. Active learning for biomedical citation screening. Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. Washington, DC, USA: ACM; 2010. p. 173-82.

[66] Kim S, Song Y, Kim K, Cha J-W, Lee GG. MMR-based active machine learning for bio named entity recognition. Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers. New York, New York: Association for Computational Linguistics; 2006. p. 69-72.

[67] Settles B, Craven M. An analysis of active learning strategies for sequence labeling tasks. Proceedings of the Conference on Empirical Methods in Natural Language Processing. Honolulu, Hawaii: Association for Computational Linguistics; 2008. p. 1070-9.

[68] Kapoor A, Horvitz E, Basu S. Selective supervision: guiding supervised learning with decision-theoretic active learning. Proceedings of the 20th international joint conference on Artifical intelligence. Hyderabad, India: Morgan Kaufmann Publishers Inc.; 2007. p. 877-82.

[69] Arora S, Nyberg E, Ros\ CP, \#233. Estimating annotation cost for active learning in a multi-annotator environment. Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing. Boulder, Colorado: Association for Computational Linguistics; 2009. p. 18-26.

[70] Haertel R, Ringger E, Seppi K, Carroll J, McClanahan P. Assessing the costs of sampling methods in active learning for annotation. Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers. Columbus, Ohio: Association for Computational Linguistics; 2008. p. 65-8.

[71] Active Learning Challenge, http://www.causality.inf.ethz.ch/activelearning.php. 2010.

[72] Socher R, Bauer J, Manning CD, Ng AY. Parsing With Compositional Vector Grammars. ACL. 2013.

[73] Li Y, McLean D, Bandar ZA, O'Shea JD, Crockett K. Sentence Similarity Based on Semantic Nets and Corpus Statistics. IEEE Trans on Knowl and Data Eng. 2006;18:1138-50.

[74] Denny JC, Smithers JD, Miller RA, Spickard A, 3rd. "Understanding" medical school curriculum content using KnowledgeMap. J Am Med Inform Assoc. 2003;10:351-62.

[75] McInnes BT, Pedersen T, Pakhomov SV. UMLS-Interface and UMLS-Similarity : open source software for measuring paths and semantic similarity. AMIA Annu Symp Proc. 2009;2009:431-5.

[76] Leacock C, Miller GA, Chodorow M. Using corpus statistics and WordNet relations for sense identification. Comput Linguist. 1998;24:147-65.

[77] Wu Z, Palmer M. Verbs semantics and lexical selection. Proceedings of the 32nd annual meeting on Association for Computational Linguistics. Las Cruces, New Mexico: Association for Computational Linguistics; 1994. p. 133-8.

[78] NIH. SNOMED Clinical Terms (SNOMED CT), http://www.nlm.nih.gov/research/umls/Snomed/snomed_main.html.

[79] NIH. MeSH, http://www.nlm.nih.gov/mesh/meshhome.html.

[80] Frey BJ, Dueck D. Clustering by passing messages between data points. Science. 2007;315:972-6.

[81] Blei DM, Ng AY, Jordan MI. Latent dirichlet allocation. J Mach Learn Res. 2003;3:993-1022.

[82] Stenetorp P, Pyysalo S, Topi G, #263, Ohta T, Ananiadou S, et al. BRAT: a web-based tool for NLP-assisted text annotation. Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics. Avignon, France: Association for Computational Linguistics; 2012. p. 102-7.

[83] http://crfpp.googlecode.com/svn/trunk/doc/index.html.

[84] http://scikit-learn.org/stable/modules/generated/sklearn.cluster.AffinityPropagation.html.

[85] https://www.i2b2.org/NLP/Relations/assets/Concept%20Annotation%20Guideline.pdf.

[86] Wilcoxin F. Probability tables for individual comparisons by ranking methods. Biometrics. 1947;3:119-22.

[87] Donmez P, Carbonell JG. Proactive learning: cost-sensitive active learning with multiple imperfect oracles. Proceedings of the 17th ACM conference on Information and knowledge management. Napa Valley, California, USA: ACM; 2008. p. 619-28.

[88] Tomanek K, Hahn U. A comparison of models for cost-sensitive active learning. Proceedings of the 23rd International Conference on Computational Linguistics: Posters. Beijing, China: Association for Computational Linguistics; 2010. p. 1247-55.