# Exploring the Efficiency of Batch Active Learning for Human-in-the-Loop Relation Extraction

Ismini Lourentzou
University of Illinois at Urbana-Champaign
lourent2@illinois.edu

Daniel Gruhl
IBM Watson Research Lab, NY, US
dgruhl@us.ibm.com

Steve Welch
IBM Watson Research Lab, NY, US
welchs@us.ibm.com

## ABSTRACT

Domain-specific relation extraction requires training data for supervised learning models, and thus, significant labeling effort. Distant supervision is often leveraged for creating large annotated corpora however these methods require handling the inherent noise. On the other hand, active learning approaches can reduce the annotation cost by selecting the most beneficial examples to label in order to learn a good model. The choice of examples can be performed sequentially, i.e. select one example in each iteration, or in batches, i.e. select a set of examples in each iteration. The optimization of the batch size is a practical problem faced in every real-world application of active learning, however it is often treated as a parameter decided in advance. In this work, we study the trade-off between model performance, the number of requested labels in a batch and the time spent in each round for real-time, domain specific relation extraction. Our results show that the use of an appropriate batch size produces competitive performance, even compared to a fully sequential strategy, while reducing the training time dramatically.

## CCS CONCEPTS

• **Information systems → Information extraction**; • **Computing methodologies → Information extraction**; *Active learning settings*; *Neural networks*;

## KEYWORDS

relation extraction;deep learning;active learning;batch mode active learning;neural networks

**ACM Reference Format:**
Ismini Lourentzou, Daniel Gruhl, and Steve Welch. 2018. Exploring the Efficiency of Batch Active Learning for Human-in-the-Loop Relation Extraction. In *WWW '18 Companion: The 2018 Web Conference Companion, April 23–27, 2018, Lyon, France.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3184558.3191546

## 1 INTRODUCTION

Many important natural language processing tasks, such as knowledge graph completion and question answering require semantic relation classification, where the goal is to categorize relations between entities in unstructured text. Supervised methods for this task are either based on hand-engineered features or learned representations by deep neural networks. However both methods rely heavily on large quantities of high-quality annotated data. The requirement of large labeled corpora limits the application of neural models to many Information Extraction tasks, as it is often quite expensive and challenging to acquire large amounts of reliable gold standard validation data for training. To address this issue approaches such as active learning and distant supervision were proposed.

Distant supervision aims to classify sentences at a bag level, where a bag contains noisy sentences mentioning the same entity pair but possibly not describing the same relation. To reduce the noise multi-instance learning is used, however these methods cannot handle sentence-level prediction or bags where all sentences do not describe a relation. Moreover the coverage of annotations is largely dependent on the type of entities/relations: while popular relations will have good coverage, tail ones may not be well represented. Thus, incorporating human annotation is crucial, especially for domains where we have many tail relations or many sentences where the entities are mentioned but the relation does not hold (e.g.,finding adverse drug events in medical forums).

Active learning tries to find the most efficient way to query the unlabeled data and learn a classifier with the minimal amount of human supervision. In classical active learning setting a single instance at each iteration is chosen. However, the sequential active learning methods have many drawbacks when combined with expensive complex models, such as neural networks: training deep networks usually takes a long time, and therefore updating the model after each label is costly in terms of both the human annotation time waiting for the next datum to tag as well as computational resources. Moreover, due to the local optimization methods used for training neural networks it is highly unlikely for a single point to result in significant impact on the performance. Therefore in practical applications it is often useful to perform batch active learning, as the cost of acquiring a batch of labels for training might be significantly less than the cost of acquiring the same number of sequential individual label requests. This holds true when the time to update the model and select the next example is prohibitively large. But under labeling budget constraints there is an inherent trade-off between efficiency and performance, as large batches will result in less frequent model updates and increased prediction error.

Decisions regarding the parameters such as the batch size or the total budget constraints are usually taken as arguments in batch model active learning related work. However, these decisions are likely to be suboptimal as they do not rely on information acquired from the data distribution or the learned model. Thus, optimizing these parameters automatically is an important problem for many tasks. Ideally, we would want a methodology that can

*WWW '18 Companion, April 23–27, 2018, Lyon, France*

inform us about the batch size rather fast irrespective of the number of unlabeled examples, i.e. low complexity.

In this paper we focus on applying neural models for extracting an arbitrary user-defined relation from a potentially infinite pool of unlabeled Web and social stream data. Despite the advantages of batch active learning, previous work in relation extraction has not explored the trade-offs between performance and batch size, or annotation costs versus training delay [5]. To better understand the nature of annotation costs for relation extraction we present an empirical study of batch active learning in ten real-world relation extraction tasks involving human annotators. More specifically we try to optimize the batch size so as to keep the model performance at a satisfactory level but reduce the total training time.

The contribution of this work is a systematic analysis for optimizing the batch size in an end-to-end neural net framework for relation extraction method with Human-in-the-Loop on any domain and concept that the user is interested in extracting. We examine several popular strategies to select the next examples to present to the human annotator: uncertainty sampling [26], QUIRE [22] and a recently proposed method that computes uncertainty estimates by sampling from the same neural model [15]. We test our hypothesis on publicly available standard datasets for relation extraction and on a challenging task of extracting causal relations among drugs and adverse drug events from user generated text. Our experimental results show that increasing the batch size in active learning up to around five examples produces comparable results with a sequential active learning approach. Furthermore, we propose to always keep the human annotator busy, even during model updates by training and performing next batch selection on slightly out-of-date information. We show that this approach reduces the total training time by $\approx 50\%$ without hurting the overall performance.

The rest of the paper is organized as follows. We give an overview of related work in Section 2; we formally define the relation extraction problem and describe our experiments in Section 3; and we present our analysis on both on publicly available standard datasets, as well as in the medical domain for the extraction of adverse drug events (Section 4). Finally, in Section 5 we describe future directions of our work.

## 2 RELATED WORK

### 2.1 Relation Extraction

Early works in relation extraction include classical machine learning approaches with SVMs and kernel-based methods as the ones most commonly used [18, 39, 60], including specialized kernels designed for relation extraction [8, 34] and Tree Kernels [12, 24, 58, 60]. Their main drawback is that they rely on human-engineered features and linguistic knowledge in the form of various Natural Language Processing operations (POS tagging, morphology, dependency parsing) [7, 39, 49], which can make them difficult to extend to new entity-relation types, different prose styles, new domains and other languages.

Considerable attention has been given to deep learning models for relation classification. Convolutional Neural Networks (CNNs) have been extensively explored: with lexical features and synonym class embeddings [29]; with the addition of POS tagging and WordNet hypernyms and pre-trained word embeddings [59]; including dependency patterns and dependency trees [9, 31, 33, 57]; exploiting pre-training on large general corpus and then fine-tuning on the target corpus [27]; relying on word-level attention mechanism to detect cues and learn which parts of a sentence are relevant to a given relation type [45]; with the combination of word embeddings and clustering to improve the generalization of relation extractors across domains [36]. Some works also investigated replacing the common soft-max loss function with a ranking-based loss function [42] and add a novel attention mechanism to capture the relevance of words with respect to the target entities [53]. Ensemble of CNNs and Recurrent Neural Networks (RNNs) have also been explored with a novel mechanism for sentence splitting and a simple voting scheme [52] as well as hierarchical attention-based RNNs [30].

The main drawback of many related works is that models are built under the assumption that a (large) pool of manually annotated examples exists already and in many cases this assumption does not hold: the definition of a relation is highly dependent on the task at hand and on the view of the user, therefore having annotated data readily available for any specific case is unlikely. Several approaches have been proposed to reduce the annotation cost for relation classification. The most prominent methods exploit large knowledge bases to automatically label entities in text [4, 16, 23, 40] and circumvent the annotation problem. Such methods rely on distant supervision and assume that when two entities co-occur a certain relation is expressed in the sentence, and then try to handle the noise [3, 28, 41, 54]. For many ambiguous relations mere co-occurrence does not guarantee the existence of the relation and these architectures can fail on the prediction task. For example, while annotating data on Adverse Drug Events (see Section 4.1) we found that half of the sentences mentioning a drug and an Adverse Drug Event do not express causality between them[1].

A machine learning system build solely on large corpora is unlikely to capture the subtle nuances of constantly evolving social language including new terms, phrases and deviation from normal usage. Human knowledge is therefore crucial, but human supervision can be expensive. Active learning methods limit this cost by selecting the most useful examples for human annotation. We briefly discuss how active learning has been leveraged in relation extraction related work.

### 2.2 Active Learning for relation extraction

Angeli et al. [3] leverage active learning for providing partial supervision to a distantly supervised relation extractor using a small number of carefully selected examples. They show that, for the 2013 KBP English Slot Filling task[2], 10,000 labeled examples and a large corpus for distantly labeled data can yield notable improvements in performance over distantly labeled data alone. Sterckx et al. [48] perform noise reduction by using semantic clustering and word embeddings: they perform hierarchical clustering of the candidate training samples to select the most reliable ones. Fu and Grishman [14] propose to interleave self-training with co-testing to reduce the

[1] In many of such cases, the condition for which you are taking the drug is mentioned. E.g., "I took aspirin for my headache".
[2] http://surdeanu.info/kbp2013/

annotation cost. The co-testing (the sampling method) leverages local and global data views [50]: a global classifier that relies on similarity of relation phrases and a local classifier that uses a set of lexical and syntactic features. The effectiveness of instance-ranking criteria used in active learning, such as uncertainty [26], representativeness [22] or information gain [15], is highly dependent on the underlying data and the relation to extract and it is very difficult to identify strong connections between any of the criteria and the task [21]. Moreover, the methods leveraged in relation extraction related work assume a sequential active learning setting, where we query one example at a time. However single instance selection strategies are quite expensive when dealing with training neural models in terms of computational resources and waiting time for the human annotator, as they require tedious retraining with each instance labeled.

### 2.3 Batch Mode Active Learning

Many batch mode active learning methods have extended single instance selection strategies or propose other heuristics based on the informativeness or diversity of the selected batch [6, 13, 20, 51]. Proposed frameworks that try to incorporate information overlap between the instances [17] treat this as an integer programming problem and utilize second-order Taylor approximation methods. Wei et al. [56] design submodular functions for specific classifiers such as Nearest Neighbors and Naive Bayes. Recent work [43] applies core-set theory to CNNs and compares with empirical risk minimization [55] and clustering [13]. However, all of the aforementioned methods have two main drawbacks:

(1) second-order methods have high complexity and do not scale well with larger datasets.
(2) the number of instances per batch is not optimized but rather pre-selected to a specific constant number.

Both components, i.e. the number of instances to be queried from a given pool of unlabeled set of examples and the selection of the specific instances to be labeled are critical for a system that can generalize to many tasks and minimize human labeling effort. Most existing work requires the number of instances in each round as input argument. In real-world applications prior knowledge is crucial for these choices. But in the case of starting a system from scratch, there is typically no knowledge of the data stream with respect to its quality, the complexity of the samples or the confidence of current models that will help in designing a classifier with good generalization accuracy. Thus we cannot decide in advance the batch size.

The question then is how to optimize the whole process taking into account annotation and training time, as well as model performance. To the best of our knowledge, the only work that optimizes both the selection of batch size and instances transforms the problem into a single optimization function that maximizes diversity, uncertainty and redundancy as well as adding a penalty term that depends on the batch size [10]. They solve the optimization problem with gradient-based methods, however apart from the quadratic complexity with respect to the number of unlabeled data samples (and thus not scaling well for large datasets), their method tries to optimize a function that penalizes larger batch sizes, while in our case we try to find the largest possible batch size that keeps performance at a satisfactory level, subject to our total annotating budget. Nevertheless, we have experimented with their method but unfortunately, even without the expensive computation of the diversity scoring function, the time for one iteration to return a batch size and the corresponding instances was prohibitive for our real-world Human-in-the-Loop system.

The aim of this work is to investigate the influence of the batch size for different active learning strategies on different relation classification tasks and extract valuable knowledge in finding the optimal batch for Human-in-the-Loop systems while keeping a satisfactory level of performance. Additionally we propose an approach that will eliminate the waiting time for the human annotator without reducing the system performance. Our training time is reduced significantly while training with 200 examples, our accuracy is on average only 5% less that a model trained on full data, which achieves 90% accuracy.

## 3 RELATION CLASSIFICATION

In this work, we treat relation extraction as a binary classification task, where given user-generated text $s$ containing one or more target entities $e_i$, our goal is to identify if $s$ expresses a certain relation $r$ among the entities $e_i$. We treat relation extraction as a cold-start problem, where no labeled data exist and query a human annotator for labels. Thus active learning is the most appropriate framework to tackle this problem.

We consider a pool-based active learning scenario [44] in which there exists a small set of labeled data $L = (x_1, y_1), \ldots, (x_{n_l}, y_{n_l})$ and a large pool of unlabeled instances $U = x_1, \ldots, x_{n_u}$. The task for the learner is to draw examples to be labeled from $U$, so as to maximize the performance of the classifier while limiting the expected number of labels requested and thus the annotation cost. In our task an instance is a text snippet expressing the relation between the entities and annotation refers to manually assigning a “true/false" label to each instance, i.e. $y_i \in \{0, 1\}$, where $y_i$ is the annotation of instance $x_i$.

To acquire a large pool of unlabeled text data from any web source such as online news articles or social media streams (Twitter, blogs etc.), one can create dictionaries using any off-the-shelf tool (e.g. [2, 11]) and select sentences based on the co-occurrence of the entities of interest. There are several approaches available for identifying entities in unstructured text [23, 40, 48], thus we treat this first step as a black-box component. We then segment the learning process into $B$ training rounds of $k$ instances at a time and interactively annotate the data as we train the models. In each round we train a neural model using the instances we have labeled so far and use the model to select the next $k$ examples to annotate from $U$. Thus our training procedure resembles recent advances in Deep Learning showing that increasing the batch size during training produces comparable results with methods that decay the learning rate but often leads to shorter training times [46].

We experiment with several active learning strategies to determine the next batch of examples, specifically:

- **us**: Uncertainty sampling [26], which ranks the samples according to the model’s belief it will mislabel them
- **quire**: QUIRE measures each instance informativeness and representativeness by its prediction uncertainty [22]

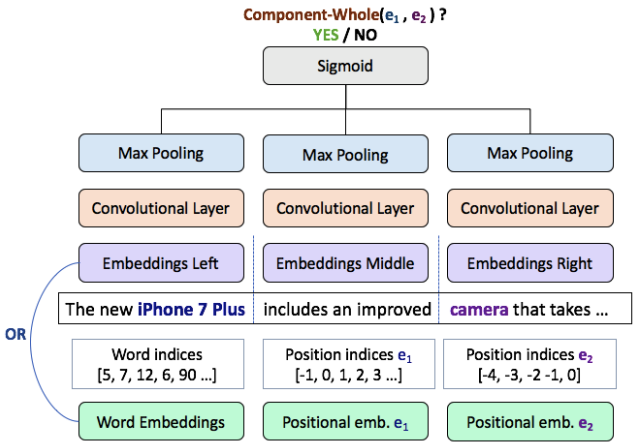


**Figure 1: CNNs for relation extraction**

- **bald**: a recently proposed combination of Monte Carlo and Dropout to obtain uncertainty measures and Bayesian Active Learning by Disagreement as an acquisition function to select examples that are expected to maximize the information gained about the model parameters [15].

Our goal is not to specifically improve a particular learning model per-se, but rather minimize the use of computational resources as well as the human annotation effort and waiting time by choosing an optimal batch size $k$.

We chose Convolutional Neural Networks (CNNs) as our classification models, as they are highly expressive leading to very low training error and faster in training than recurrent architectures. More importantly CNNs are known to perform well in the relation classification task [37, 59]. To keep our classifier lightweight and robust our input representations rely solely on distributional semantics and not on lexical features or any other language-dependent prior knowledge, as shown in Fig. 1:

- **CNNpos**: Positional features [59] along with word sequences, i.e. we generate three embedding matrices, one initialized with pre-trained word embeddings and two randomly initialized for the positional features
- **CNNcontext**: context-wise splits of sentences [1], i.e. using pre-trained word embeddings and the two entities in the text as split points to generate three matrices - left, middle and right context.

Our models are using 100-dimensional pre-trained Glove word embeddings [38], 100-dimensional positional embeddings, and contain 300 convolutional filters, kernels of width 3, and ReLU nonlinearities [35]. Training is performed with cross-entropy as cost function that is optimized with Adam [25] with 0.001 initial learning rate. Dropout is set to 0.25.

## 4 EXPERIMENTS

As noted, the relation extraction task is a challenging one. Especially in the case of developing early prototype systems little can be done with a traditional neural network in the absence of a significant quantity of hand labeled data. While a task specific labeling system

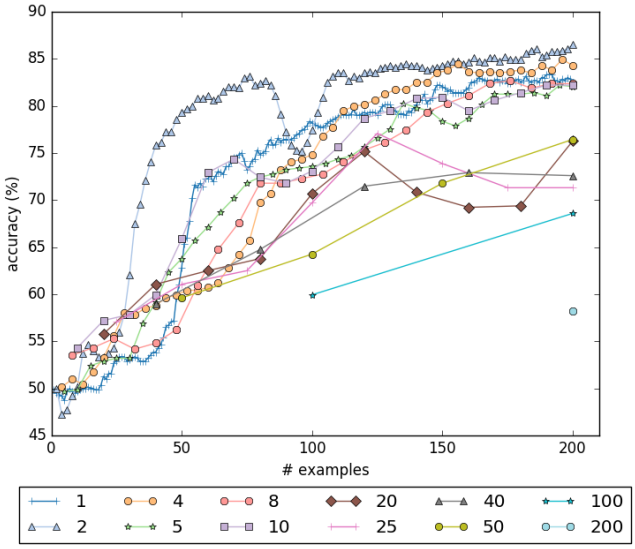


**Figure 2: A look at the impact of batch size on training rate for one active learning strategy, one neural structure on one task. Note that the best strategy in this case is two at a time.**

can help [47], it makes sense to consider the "best order" to ask the user for input in the hopes of achieving a sufficiently performant system with minimal human effort.

Our goal in this work is to limit the human and computational resources without significantly impacting the performance of the models by optimizing the active learning batch size for an arbitrary relation extraction task. We simulate the Human-in-the-Loop by using existing benchmark datasets on relation extraction. More specifically, we treat all examples as unlabeled and "request" the annotations in small batches from the existing labels, as if they were annotated in real-time by a user. This setting allows us to run in parallel multiple experiments varying the batch size for all active learning strategies and all tasks. We also continue our analysis on our real case scenario of extracting Adverse Drug Reaction (details on the data in Section 4.1).

Our experiments showcase a methodology that can be used to decide on the optimal batch size based on the *average* performance on datasets that solve the same task for disjoint domains, for example relation extraction where the relation is different across datasets. We present a set of directly useful recommendations that can guide the development of domain-specific relation extraction systems.

### 4.1 Datasets

For our analysis to produce robust results that generalize across relation classification tasks and models we utilize two different datasets containing 10 relations in total:

(1) We perform our analysis on a real case experiment by extracting Adverse Drug Events (ADE) relations from a Web forum[3]. Our Human-in-the-Loop is a medical doctor using our system to annotate the data. In this dataset posts are tagged based on mentions of certain drugs, adverse drug

[3] http://www.askapatient.com/

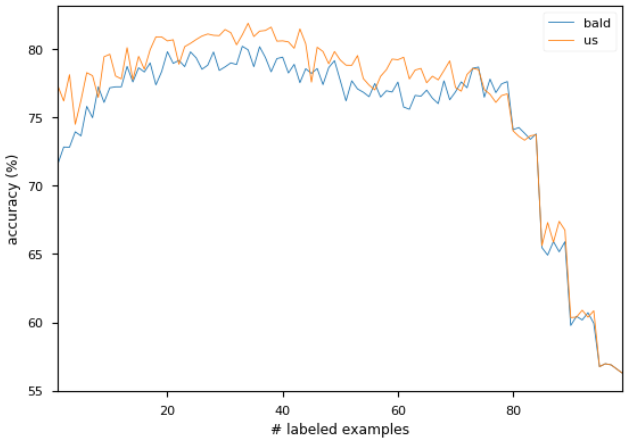


**Figure 3: An exploration of the impact of initial batch size. For our datasets an an initial batch of 30 seems like a good place to start. It gives enough examples to begin to span the space. This plot is the average of 10 datasets with CNNcontext as our classification model.**

reactions, symptoms, findings etc. However, the mere co-occurrence of a drug and an ADE in a sentence does not necessarily imply a causal event/drug relation among the two. We name this dataset *causalADEs* [32].

(2) We also leverage existing corpora: the *Semeval 2010 - Task 8* dataset [19], which consists of 8,000 training and 2,717 test examples covering nine relation types: Cause-Effect, Component-Whole, Content-Container, Entity-Destination, Message-Topic, Entity-Origin, Instrument-Agency, Member-Collection, Product-Producer. Additionally, some sentences are labeled as "Other", indicating that none of those relations are expressed.

We run a series of experiments to quantify best practices with respect to batch size and HumL systems. Ultimately, we will examine 10 tasks and apply 3 active learning strategies to them. Initially though, we will begin looking at a single plot, that of using uncertainty sampling to train a CNNcontext model on the SemEval Component task (see Figure 2). In this experiment, we train the model with batches of various sizes without any pre-training. This allows us to observe how the model is affected by varying the batch size in cold-start scenarios when no annotated data are available and we wish to start the human annotation process as quickly as possible.

There are a few things to notice here. The first is that training with 100 or 200 examples per batch is substantially less efficacious than the smaller batches. Additionally note that by the time you've scored 200 examples, batches of 5 or 10 do nearly as well as anything else. Lastly note that until there are around 20 examples scored the system does not really take off. The intuition here is that you need enough examples to "span the space" or you end up over fitting what little data you have. It's this last point we will examine first.

## 4.2 Initial Batch

As we noticed above, despite having the best performance in the end, active learning with just one or two examples in not appropriate when initializing the model due to high variance in small corpora, thus the model tends to "overfit" these first few concepts. An alternative is to order the data based on unsupervised text based criteria and select the highest ranked ones as initial training examples. Our experiments with several criteria, including random, showed that maximizing linguistic dissimilarity between sentences (by utilizing Glove embeddings) works well [32].

The first question is how large this initial batch should be for good results. We explore this by fixing the learning batch size at 5 and vary the size of the initial batch ($B^0$) generated via linguistic dissimilarity to prime the run. We continue the process until we hit a fixed training size of 200 (our budget constraint) and plot the accuracy at 200. As you can see in Figure 3 [4], starting with a batch of about 20-40 examples results in better results. The intuition here is that less than 20 and the system overfits the initial training data, more than 40 and the active learning is unable to take over and focus on the regions of confusion.

**Recommendation:** Use an initial batch ($B^0$) derived through linguistic dissimilarity of about 30 labeled examples to train the system before engaging active learning for more efficient human annotation.

## 4.3 Subsequent Batch Size

After obtaining an initial linguistically diverse batch of 30 examples as a good starting point, we need to decide on a proper subsequent batch size. Since computing the next "batch" and loading it into the UI for the subject matter expert to score takes some time, there is a preference for larger batches. However, as Figure 4 shows, there is a negative impact of these larger batch sizes (here we compare the accuracy across different batch sizes, after 100 training examples). The best performance is when using batch size of 1, but the real drop seems to be after 5 (which only loses 5% compared to the batch size of 1). Thus, if your system has a finite cost associated with generating batches this may be good place to stop.

**Recommendation:** A default batch size of 5 examples seems to be a good compromise between efficiency of example generation and speed of learning in the active system.

## 4.4 Interleaving

While the prior section points out the advantage of smaller batches, these advantages do not come for free. Generating a batch of examples and loading it into the scoring framework for the user to look at takes time. We have observed that generating a training example for a single sentence (e.g., is there a causal relation between A and B) with a good UI and well defined task takes between three and ten seconds on average. Five seconds is a fairly good median. If it takes 25 seconds to compute a batch and load it into the UI, then the work flow for a single item batch will be:

(1) User spends 5 seconds scoring a single example.
(2) System spends 25 seconds getting the next example ready.
(3) Repeat.

Or, in other words, over 80% of the time the user is sitting around waiting. Even with the recommended batch size of 5 the user will

[4]Due to being computationally expensive to perform such an experiment with QUIRE, we do not include it in this experiment. However, we performed an experiment in which we train QUIRE with {5, 10, 20, 40, 50} examples per batch and the trend looks similar to our results for uncertainty and bald, with a performance lower than bald.

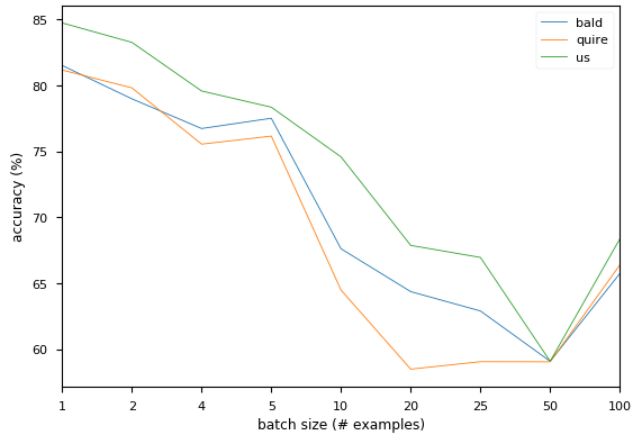


**Figure 4: A view of performance of the CNNcontext model trained under different active learning methods. This is a look at the performance after 100 examples have been scored. As can be seen, compared to the fully sequential approach of one example at a time, there is approximately only 5% decrease in the performance of using a slightly larger batch size of 5 examples.**

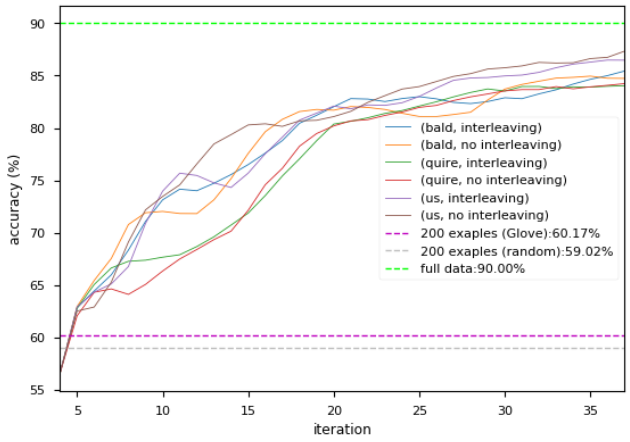


**Figure 5: Comparison of interleaving and classic training sessions in terms of accuracy.**

be spending half their time waiting. The single largest cost in a Human-in-the-Loop system is the human annotation time. In an ideal world they would be scoring constantly.

Interleaving provides us a potential for achieving this. Without interleaving the system trains batch $B^n$ using all the information from batches $B^0 \dots B^{n-1}$. With interleaving it uses only $B^0 \dots B^{n-2}$. This means that the user can be scoring a batch $B^{n-1}$ while the computation and loading of the next batch $B^n$ is occurring.

Obviously, with less training data the accuracy is likely to suffer; the question is by how much. We perform this experiment by comparing the two approaches, i.e. with or without interleaving, using a $B^0$ of 30 and a batch size of 5 (see Figure 5). As can be seen, these two are quite close in terms of accuracy. If we additionally plot the total time required for all iterations (Figure 6) the result is even more striking; we see that interleaving produces comparable performance in $\approx$ 50% less training time, irrespective of the active learning method chosen. Moreover, we showcase the inefficiency of training for one round with 200 examples (horizontal lines).

**Recommendation:** Use interleaving with a batch size that is as small as possible while still allowing continuous human work.

## 4.5 Active Learning comparison

To conclude, we also present a comparison of active learning methods. As expected, Uncertainty is much faster than the rest of the active learning strategies, and QUIRE is slower than all (Fig. 6). Since bald requires sampling from the model during testing time, it requires slightly more time than uncertainty to compute the final ranking of the samples, but also suffers from noise due to the monte carlo estimation of the ranking score. Uncertainty seems to be the winner in all dimensions, as it produces the best results faster (shown in Figures 3-5). Despite the fact that using only uncertainty does not incorporate other information, such as representativeness or diversity, the method is extremely robust and appropriate for Human-in-the-Loop applications that require efficient switching between model updates and human querying.

**Recommendation:** Start with active learning methods that are based on fast, less complex metrics. Compare with additional methods when sufficient data are gathered.

## 4.6 Overall impact

In figure 5 we can also see the overall impact of leveraging active learning methods. The dotted line represents scoring 200 examples selected with linguistic dissimilarity; it indicates 61% accuracy, with random being slightly lower. For a fixed amount of work (200 examples) we see our prescription results in a 40% increase in performance (to an accuracy of 86%). For a fixed performance point we see an even more impressive result, as 25 scored examples achieve the same performance as the 200, an 72.5% reduction in human time.

Finally, we also plot the average accuracy of training with all data available as labeled. We see that the difference in terms of accuracy with our best performing model is only 4%. On average each relation task has a pool with more than 1,000 examples. Thus our system is trained on only 20% of the data, a result that proves the importance of incorporating human knowledge on relation extraction systems.

# 5 CONCLUSIONS AND FUTURE WORK

Relation extraction for any arbitrary domain of user interest is a challenging task. To leverage state-of-the-art neural network approaches in settings where large pre-annotated corpora are not available human annotation is necessary. In this work, we aim to reduce to the computational and annotation costs incurred from training a relation classifier under streamed annotations, while sustaining a reasonable level of performance. We provide an analysis of active learning methods that can be adapted to the setting in which labels are requested in equal-sized batches of $k$ examples. We show that as $k$ increases, the model performance is lower than that of the analogous results for fully-sequential active learning. Our experimental results show that we can achieve competitive performance for extracting relations with very little annotated data. Finally, we propose a method that trains on slightly outdated information but

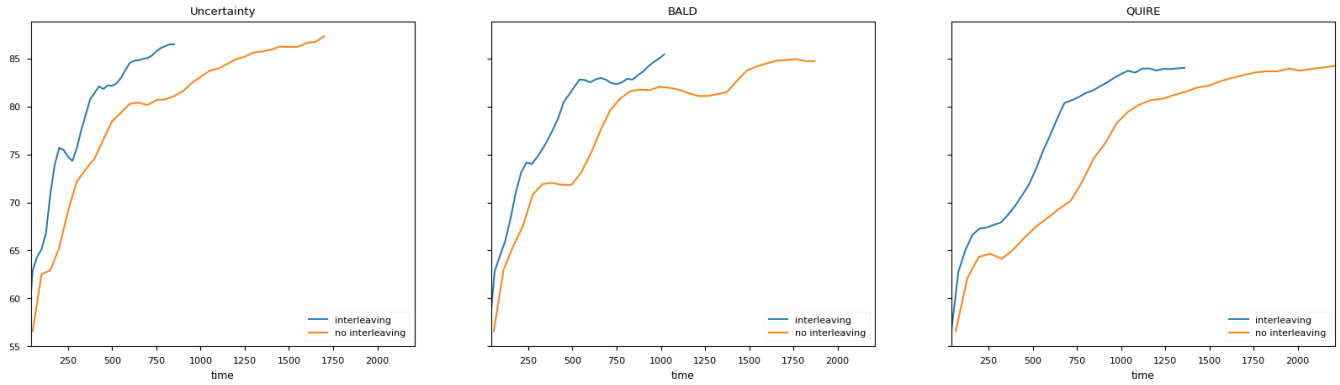


**Figure 6: Comparison of interleaving and classic training sessions in terms of total training and labeling time.**

keeps the human annotator busy, and show that this results in a $\approx$ 50% decrease in total time without significantly impacting the accuracy of the resulting model. Our Human-in-the-Loop system can easily learn new arbitrary relations efficiently, fully leveraging the human annotator throughout the process.

Our work is directly applicable to Human-in-the-Loop relation extraction. However we have experimented only with RE systems, and therefore our work is still tentative for general applications. Although intuitively we believe that the recommendations should generalize well across many tasks, our results could be potentially sensitive to the data distribution. We leave the analysis of this sensitivity to future work.

Active learning might be widely explored but several components remain as open problems. We conclude with a description of potential future directions that we hope to explore:

- Adaptive batch size active learning methods, where the batch is changed dynamically between iterations, depending on additional features of specific instances.
- Our work assumes perfect ground truth labels. However, in reality we often deal with non-perfect labelers and this introduces challenges in real-world applications of active learning. It would be useful to explore how the optimal batch size varies with respect to the labeling noise.
- Blending semi-supervised with batch active learning, as this will help us explore distributional semantics for pre-training our models and potentially decrease the number of labels needed to reach a good performance.
- Framing the relation extraction problem as a resource-bounded multi-objective optimization problem and try to reduce the complexity of batch mode active learning methods.
- Meta-learning approaches, i.e. learning the best active learning strategy instead of relying on heuristics such as uncertainty, diversity etc. Current meta-learning approaches are limited to stream-based active learning or static one-step selection of a batch for labeling. Extending to pool-based adaptive scenarios can potentially leverage the representational similarity of unlabeled data points and lower the total number of examples that the system asks humans to annotate.

## REFERENCES


[1] Heike Adel, Benjamin Roth, and Hinrich Schütze. Comparing convolutional neural networks to traditional models for slot filling. In *NAACL-HLT*, 2016.

[2] Alfredo Alba, Anni Coden, Anna Lisa Gentile, Daniel Gruhl, Petar Ristoski, and Steve Welch. Language agnostic dictionary extraction. In *ISWC (ISWC-PD-Industry)*, number 1963 in CEUR Workshop Proceedings, 2017.

[3] Gabor Angeli, Julie Tibshirani, Jean Wu, and Christopher D Manning. Combining distant and partial supervision for relation extraction. In *EMNLP*, pages 1556–1567, 2014.

[4] Isabelle Augenstein, Diana Maynard, and Fabio Ciravegna. Distantly supervised web relation extraction for knowledge base population. *Semantic Web*, 7(4):335–349, 2016.

[5] Nguyen Bach and Sameer Badaskar. A review of relation extraction. *Literature review for Language and Statistics II*, 2, 2007.

[6] Klaus Brinker. Incorporating diversity in active learning with support vector machines. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 59–66, 2003.

[7] Razvan Bunescu and Raymond Mooney. Learning to extract relations from the web using minimal supervision. In *ACL*, 2007.

[8] Razvan C Bunescu and Raymond J Mooney. A shortest path dependency kernel for relation extraction. In *HLT/EMNLP*, pages 724–731. ACL, 2005.

[9] Rui Cai, Xiaodong Zhang, and Houfeng Wang. Bidirectional recurrent convolutional neural network for relation classification. In *ACL*, 2016.

[10] Shayok Chakraborty, Vineeth Balasubramanian, and Sethuraman Panchanathan. Adaptive batch mode active learning. *IEEE transactions on neural networks and learning systems*, 26(8):1747–1760, 2015.

[11] Anni Coden, Daniel Gruhl, Neal Lewis, Michael Tanenblatt, and Joe Terdiman. SPOT the drug! An unsupervised pattern matching method to extract drug names from very large clinical corpora. *HISB'12*, pages 33–39, 2012.

[12] Aron Culotta and Jeffrey Sorensen. Dependency tree kernels for relation extraction. In *ACL*, 2004.

[13] Begüm Demir, Claudio Persello, and Lorenzo Bruzzone. Batch-mode active-learning methods for the interactive classification of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 49(3):1014–1031, 2011.

[14] Lisheng Fu and Ralph Grishman. An efficient active learning framework for new relation types. In *IJCNLP*, 2013.

[15] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep Bayesian Active Learning with Image Data. In *ICML*, 2017.

[16] Anna Lisa Gentile, Ziqi Zhang, Isabelle Augenstein, and Fabio Ciravegna. Unsupervised wrapper induction using linked data. In *K-CAP*, pages 41–48. ACM, 2013.

[17] Yuhong Guo and Dale Schuurmans. Discriminative batch mode active learning. In *NIPS*, 2008.

[18] Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. Exploring various knowledge in relation extraction. In *ACL*, 2005.

[19] Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *DEW Workshop*, pages 94–99. ACL, 2009.

[20] Steven CH Hoi, Rong Jin, and Michael R Lyu. Large-scale text categorization by batch mode active learning. In *Proceedings of the 15th international conference on World Wide Web*, pages 633–642. ACM, 2006.

[21] Wei-Ning Hsu and Hsuan-Tien Lin. Active learning by learning. In *AAAI*, 2015.

[22] Sheng-Jun Huang, Rong Jin, and Zhi-Hua Zhou. Active learning by querying informative and representative examples. In *NIPS*, pages 892–900, 2010.

[23] Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. Distant supervision for relation extraction with sentence-level attention and entity descriptions. In *AAAI*, pages 3060–3066, 2017.

[24] Nanda Kambhatla. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, page 22. ACL, 2004.

[25] Diederik P. Kingma and Jimmy Ba. Adam:a method for stochastic optimization. In *ICLR*, 2015.

[26] David D Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In *ICML*, pages 148–156, 1994.

[27] Zhuang Li, Lizhen Qu, Qiongkai Xu, and Mark Johnson. Unsupervised pre-training with sequence reconstruction loss for deep relation extraction models. In *Australasian Language Technology Association Workshop 2016*.

[28] Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. Neural relation extraction with selective attention over instances. In *ACL*, 2016.

[29] ChunYang Liu, WenBo Sun, WenHan Chao, and Wanxiang Che. Convolution neural network for relation extraction. In *Part II of the Proceedings of the 9th International Conference on Advanced Data Mining and Applications-Volume 8347*, 2013.

[30] Minguang Xiao Cong Liu. Semantic relation classification via hierarchical recurrent neural network with attention. In *COLING*, 2016.

[31] Yang Liu, Furu Wei, Sujian Li, Heng Ji, Ming Zhou, and Houfeng Wang. A dependency-based neural network for relation classification. In *arXiv preprint arXiv:1507.04646*, 2015.

[32] Ismini Lourentzou, Alfredo Alba, Anni Coden, Anna Lisa Gentile, Daniel Gruhl, and Steve Welch. Mining relations from unstructured content. In *Advances in Knowledge Discovery and Data Mining - 22nd Pacific-Asia Conference, PAKDD 2018, Melbourne, Australia, June 2018*, page to appear, 2018.

[33] Makoto Miwa and Mohit Bansal. End-to-end relation extraction using lstms on sequences and tree structures. In *arXiv preprint arXiv:1601.00770*, 2016.

[34] Raymond J Mooney and Razvan C Bunescu. Subsequence kernels for relation extraction. In *NIPS*, pages 171–178, 2006.

[35] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, pages 807–814, 2010.

[36] Thien Huu Nguyen and Ralph Grishman. Employing word representations and regularization for domain adaptation of relation extraction. In *ACL*, 2014.

[37] Thien Huu Nguyen and Ralph Grishman. Relation extraction: Perspective from convolutional neural networks. In *VS@ HLT-NAACL*, 2015.

[38] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.

[39] Longhua Qian, Guodong Zhou, Fang Kong, Qiaoming Zhu, and Peide Qian. Exploiting constituent dependencies for tree kernel-based semantic relation extraction. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 697–704. ACL, 2008.

[40] Alexander J. Ratner, Christopher De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. Data programming: Creating large training sets, quickly. In *NIPS*, pages 3567–3575, 2016.

[41] Benjamin Roth, Tassilo Barth, Michael Wiegand, and Dietrich Klakow. A survey of noise reduction methods for distant supervision. In *AKBC*, pages 73–78. ACM, 2013.

[42] Cicero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. Classifying relations by ranking with convolutional neural networks. In *arXiv preprint arXiv:1504.06580*, 2015.

[43] Ozan Sener and Silvio Savarese. A geometric approach to active learning for convolutional neural networks. *arXiv preprint arXiv:1708.00489*, 2017.

[44] Burr Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11, 2010.

[45] Yatian Shen and Xuanjing Huang. Attention-based convolutional neural network for semantic relation extraction. In *COLING*, 2016.

[46] Samuel L Smith, Pieter-Jan Kindermans, and Quoc V Le. Don't decay the learning rate, increase the batch size. *arXiv preprint arXiv:1711.00489*, 2017.

[47] Gabriel Stanovsky, Daniel Gruhl, and Pablo Mendes. Recognizing mentions of adverse drug reaction in social media using knowledge-infused recurrent models. In *EACL*, pages 142–151. ACL, 2017.

[48] Lucas Sterckx, Thomas Demeester, Johannes Deleu, and Chris Develder. Using active learning and semantic clustering for noise reduction in distant supervision. In *AKBC at NIPS*, pages 1–6, 2014.

[49] Fabian M Suchanek, Georgiana Ifrim, and Gerhard Weikum. Combining linguistic and statistical analysis to extract relations from web documents. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 712–717. ACM, 2006.

[50] Ang Sun and Ralph Grishman. Active learning for relation type extension with local and global data views. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1105–1112. ACM, 2012.

[51] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66, 2001.

[52] Ngoc Thang Vu, Heike Adel, Pankaj Gupta, et al. Combining recurrent and convolutional neural networks for relation classification. In *NAACL-HLT*, pages 534–539, 2016.

[53] Linlin Wang, Zhu Cao, Gerard de Melo, and Zhiyuan Liu. Relation classification via multi-level attention cnns. In *ACL*, 2016.

[54] Xiaobin Wang, Yu Hong, Jianmin Yao, Qiaoming Zhu, and Guodong Zhou. A novel approach for relation extraction with few labeled data. pages 73–84, 2016.

[55] Zheng Wang and Jieping Ye. Querying discriminative and representative samples for batch mode active learning. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 9(3):17, 2015.

[56] Kai Wei, Rishabh Iyer, and Jeff Bilmes. Submodularity in data subset selection and active learning. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1954–1963, 2015.

[57] Kun Xu, Yansong Feng, Songfang Huang, and Dongyan Zhao. Semantic relation classification via convolutional neural networks with simple negative sampling. *arXiv preprint arXiv:1506.07650*, 2015.

[58] Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. Kernel methods for relation extraction. *Journal of machine learning research*, 3:1083–1106, 2003.

[59] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, Jun Zhao, et al. Relation classification via convolutional deep neural network. In *COLING*, pages 2335–2344, 2014.

[60] Shubin Zhao and Ralph Grishman. Extracting relations with integrated information using kernel methods. In *ACL*, pages 419–426. ACL, 2005.