Proactive Learning for Named Entity Recognition

Maolin Li, Nhung T. H. Nguyen, Sophia Ananiadou
National Centre for Text Mining
School of Computer Science, The University of Manchester, United Kingdom
{maolin.li, nhung.nguyen, sophia.ananiadou}@manchester.ac.uk

Abstract

The goal of active learning is to minimise the cost of producing an annotated dataset, in which annotators are assumed to be perfect, i.e., they always choose the correct labels. However, in practice, annotators are not infallible, and they are likely to assign incorrect labels to some instances. Proactive learning is a generalisation of active learning that can model different kinds of annotators. Although proactive learning has been applied to certain labelling tasks, such as text classification, there is little work on its application to named entity (NE) tagging. In this paper, we propose a proactive learning method for producing NE annotated corpora, using two annotators with different levels of expertise, and who charge different amounts based on their levels of experience. To optimise both cost and annotation quality, we also propose a mechanism to present multiple sentences to annotators at each iteration. Experimental results for several corpora show that our method facilitates the construction of high-quality NE labelled datasets at minimal cost.

1 Introduction

Manually annotating a dataset with NEs is both time-consuming and costly. Active learning, a semi-supervised machine learning algorithm, aims to address such issues (Lewis, 1995; Settles, 2010). Instead of asking annotators to label the whole dataset, active learning methods present only representative and informative instances to annotators. Through iterative application of this process, a high-quality annotated corpus can be produced in less time and at lower cost than traditional annotation methods.

There are two strong assumptions in active learning: (1) instances are labelled by experts, who always produce correct annotations and are not affected by the tedious and repetitive nature of the task; (2) all annotators are paid equally, regardless of their annotation quality or level of expertise. However, in practice, it is highly unlikely that all annotators will assign accurate labels all of the time. For example, especially for complex annotation tasks, some labels are likely to be assigned incorrectly (Donmez and Carbonell, 2008, 2010; Settles, 2010). Furthermore, if annotation is carried out for long periods of time, tiredness and reduced concentration may ensue (Settles, 2010), which can lead to annotation errors. An additional issue is that different annotators may have varying levels of expertise, which could make them reluctant to annotate certain cases, and they may assign incorrect labels in other cases. It is also possible that an inexperienced annotator may assign random labels.

To address the above-mentioned assumptions, proactive learning has been proposed to model different types of experts (Donmez and Carbonell, 2008, 2010). Proactive learning assumes that (1) not all annotators are perfect, but that there is at least one "perfect" expert and one less experienced or "fallible" annotator; (2) as the perfect expert always provides correct answers, their time is more expensive than that of the fallible annotator. The annotation process in proactive learning is similar to traditional active learning. At each iteration, annotators will be asked to tag an unlabelled instance, the result of which will be added to the labelled dataset. However, the difference with proactive learning is that, in order to reduce annotation cost, an appropriate annotator is chosen to label each selected instance. For example, if

Proceedings of the BioNLP 2017 workshop, pages 117–125,
Vancouver, Canada, August 4, 2017.

there is a high probability that the fallible annotator will provide the correct label for an unlabelled instance, then proactive learning will send this instance to be annotated by fallible annotator. This aims to ensure a simultaneous saving of costs and maintenance of the quality of the data.

Proactive learning has been used for several annotation tasks, such as binary and multi-class text classification, and parsing (Donmez and Carbonell, 2008, 2010; Olsson, 2009). In contrast, this paper proposes a proactive learning method for NE tagging, i.e., a sequence labelling task.

Similarly to other efforts that have used proactive learning, our method models two annotators: a reliable one and a fallible one, who have different probabilities of providing correct labels. The reliable annotator is much more likely to produce correct annotations, but their time is expensive. In contrast, the fallible annotator is likely to assign incorrect annotations more often, but charges less for their services. It should be noted that the characteristics of our reliable expert are different from those proposed in previous work (Donmez and Carbonell, 2008, 2010). Specifically, in the conventional proactive learning, the reliable expert is assumed to be perfect, i.e., he/she always provides correct annotations. However, in practice, such an assumption is too strong, especially for NE annotation. Therefore, we assume that the reliable expert is not perfect, but that he/she has a higher expertise level in the target domain, and has a very low error rate. In order to determine an appropriate annotator for each sentence, we calculate the probability that an annotator will assign the correct sequence of labels in a selected unlabelled sentence. Furthermore, at each iteration, we use a batch sampling mechanism to select several sentences for annotators to label (instead of selecting only a single sentence), which optimises both cost and performance.

For evaluation purposes, we simulate the two annotators by using two machine-learning based NER methods, namely LSTM-CRF (Lample et al., 2016) as the reliable expert, and CRF (Lafferty et al., 2001) as the fallible expert. We then apply our method to three corpora from different domains: ACE2005 (Walker et al., 2006) for general language entities, COPIOUS—an in-house corpus of biodiversity entities[1], and GENIA (Kim et al., 2003)—a corpus of biomedical entities. Our experimental results demonstrate that by using the proposed method, we can obtain a high-quality labelled corpus at a lower cost than current baseline methods.

The contributions of our work are as follows. Firstly, we have modified the conventional proactive learning method to ensure its suitability for a sequence labelling task. Secondly, in contrast to previous work, which selects a single instance for each annotator at each iteration (Donmez and Carbonell, 2008, 2010; Moon and Carbonell, 2014), our method selects multiple sentences for presentation to annotators. Thirdly, by applying our method to a number of different corpora, we demonstrate that our method is generalisable to different domains.

2 Methodology

The proposed proactive learning for NE tagging is outlined in Algorithm 1. As an initial step, the performance of each expert is estimated based on a benchmark dataset (see Section 2.1). Subsequently, at each iteration, all sentences in the unlabelled dataset are sorted according to an active learning criterion. The top-N most informative sentences are then used as input to the batch sampling step. In this step, a batch of sentences is divided into two sets to be distributed to the reliable and fallible experts, respectively. Sentences distributed to the fallible experts are not only informative, but there is also a high probability that the expert will provide correct labels for them. Meanwhile, only those sentences that are estimated to be too difficult for the fallible expert to annotate will be sent to the reliable expert. By applying this process, annotation cost can be reduced. Further details about the batch sampling algorithm are presented in Section 2.2.

In Algorithm 1, UL_r is the set of selected unlabelled sentences assigned to the reliable expert and UL_f is the set assigned to the fallible expert. L_r, L_f are the annotated results of UL_r, UL_f.

2.1 Expert performance estimation

As mentioned above, our method assumes that there are two types of experts. One is reliable, who has a higher probability of assigning the correct sequence of labels for a sentence, and has a high cost for their time. The other expert is fallible, meaning that they may assign a higher proportion of incorrect labels for a sequence, but

[1]The corpus is available upon request.

Algorithm 1: Proactive Learning for NER

Input: a labelled dataset L, an unlabelled dataset UL, a test dataset T, a budget B, a reliable expert e_r with cost C_r for each sentence, a fallible expert e_f with cost C_f, the current cost C
Output: a labelled dataset L

1 Estimate the performance of each expert as described in Section 2.1;
2 **while** $C < B$ **do**
3 Train a named entity recognition model M on L;
4 Sort all sentences in the unlabelled dataset according to an active learning criterion;
5 Select the top N sentences;
6 $UL_r, UL_f = BatchSampling(M, top\ N \text{ sentences})$;
7 $L_r, L_f \leftarrow e_r$ and e_f annotate UL_r and UL_f respectively;
8 $L = L \cup L_r \cup L_f$;
9 $UL = UL - UL_r - UL_f$;
10 $C = C + C_r * |L_r| + C_f * |L_f|$;
11 **end**

charges less for their time. The likely annotation quality of each expert is estimated based on two different probabilities: the class probability, $p(label|expert, c)$ and the sentence probability $p(CorrectLabels|expert, \boldsymbol{x})$.

2.1.1 Class probability

The class probability, $p(label|expert, c)$, is the probability that an $expert$ provides a correct label when annotating a named entity of class c. This probability is obtained by asking both the reliable and fallible experts to annotate a benchmark dataset and calculating F_1 scores for each of them against the gold standard annotations.

2.1.2 Sentence probability

The sentence probability is the probability that an $expert$ provides a sequence of correct labels for a sentence $\boldsymbol{x}$.

We firstly compute the probability for each token in the sentence by combining the class probability and the likelihood that an $expert$ provides a correct label for the token $\boldsymbol{x}_i$, as shown in Equation 1. The equation is inspired by Moon and Carbonell (2014), who used it for a classification task.

$$p(CorrectLabel|expert, \boldsymbol{x}_i) = \sum_{c}^{|C|} p(c|\boldsymbol{x}_i) * p(label|expert, c) \quad (1)$$

C is the set of all entity labels and the label O. $p(c|\boldsymbol{x}_i)$ is the probability that a token $\boldsymbol{x}_i$ is an entity of class c, which is predicted by an NER model.

Algorithm 2: Batch Sampling

Input: a named entity recognition model M, top-N sentences selected according to an active learning criterion
Output: UL_r, UL_f

1 $UL_r = \emptyset$;
2 $UL_f = \emptyset$;
3 **while** *Batch Size* **do**
  // Stage 1
4 **foreach** *sentence* $\boldsymbol{x}$ **do**
5 **if** $p(CorrectLabels|fallible, \boldsymbol{x}) > \alpha$ **then**
6 $UL_f = UL_f \cup \{\boldsymbol{x}\}$;
7 BatchSize = BatchSize - 1
8 **end**
9 **end**
  // Stage 2
10 **if** *Batch Size* $\neq 0$ **then**
11 Sort the remaining sentences according to a re-ranking criterion;
12 Calculate threshold β;
13 **foreach** *sentence* $\boldsymbol{x}$ **do**
14 **if** *Batch Size* $\neq 0$ **then**
15 **if** $diff(reliable, fallible, \boldsymbol{x}) < \beta$ **then**
16 $UL_f = UL_f \cup \{\boldsymbol{x}\}$;
17 **else**
18 $UL_r = UL_r \cup \{\boldsymbol{x}\}$;
19 **end**
20 BatchSize = BatchSize - 1;
21 **end**
22 **end**
23 **end**
24 **end**

Given the probabilities that an expert will provide correct labels for each tokens in a sentence, the sentence probability is calculated by averaging all of these probabilities, as presented in Equation 2.

$$p(CorrectLabels|expert, \boldsymbol{x}) = \frac{\sum_{i}^{|\boldsymbol{x}|} p(CorrectLabel|expert, \boldsymbol{x}_i)}{|\boldsymbol{x}|} \quad (2)$$

$|\boldsymbol{x}|$ is the length of the sentence $\boldsymbol{x}$.

2.2 Batch sampling

Instead of asking annotators to label only one sentence at each iteration, it is more efficient to ask them to annotate several sentences. To facilitate this, we propose a batch sampling algorithm that can select a set of sentences and assign them to appropriate annotators (see Algorithm 2).

The input of the algorithm is a set of sentences in the unlabelled dataset that are considered to be the most informative ones, based on an active learning criterion (as described in line 5 of Algorithm 1).

This batch sampling process is divided into two stages. In the first stage, unlabelled sentences for which the sentence probability for the fallible expert is higher than a threshold α, will be assigned to the fallible expert. Otherwise, the sentence will be passed to the second stage. In the second stage, we firstly reorder sentences according to a re-ranking criterion, as shown in Equation 3. The intuition behind this re-ranking step is that in order to save on annotation costs, we set a high priority for sentences to be assigned to the fallible expert in certain cases. Specifically, for sentences that are informative and for which there is a small difference between the sentence probabilities for the reliable and fallible experts, we favour the selection of the fallible one.

$$ReRankingCriterion = \frac{ActiveLearningCriterion(\boldsymbol{x})}{diff(reliable, fallible, \boldsymbol{x})} \quad (3)$$

For an unlabelled sentence $\boldsymbol{x}$, the difference between the sentence probabilities for the two experts is calculated as shown in Equation 4.

$$\begin{aligned} &diff(reliable, fallible, \boldsymbol{x}) \\ &\quad = |p(CorrectLabels|reliable, \boldsymbol{x}) \\ &\qquad - p(CorrectLabels|fallible, \boldsymbol{x})| \end{aligned} \quad (4)$$

If the above difference is not significant, i.e., it is less than a threshold β, $\boldsymbol{x}$ will be distributed to the fallible expert. Otherwise, $\boldsymbol{x}$ will be assigned to the reliable expert.

Equations (5) - (7) describe the estimation of the threshold β, in which $\boldsymbol{x}^i$ is the i^{th} sentence in the top-N sentences selected by an active learning criterion. γ is a parameter that controls the value of the threshold β. γ ranges from 0 to 1. If $\gamma = 0$, no sentences will be given to the fallible expert to annotate. If $\gamma = 1$, the fallible expert will label all the $BatchSize$ sentences. It should be noted that β is a dynamic threshold, which is recalculated based on the difference between $diff_{max}$ and $diff_{min}$ at each iteration.

$$diff_{min} = min_i^N(diff(reliable, fallible, \boldsymbol{x}^i)) \quad (5)$$

$$diff_{max} = max_i^N(diff(reliable, fallible, \boldsymbol{x}^i)) \quad (6)$$

$$\beta = diff_{min} + \gamma(diff_{max} - diff_{min}) \quad (7)$$

3 Experiments

3.1 Dataset

We have applied our method to three different corpora: (1) ACE2005 (Walker et al., 2006) which includes named entities for the general domain, e.g., person, location, and organisation; (2) COPIOUS that includes five categories of biodiversity entities, such as taxon, habitat, and geographical location; (3) GENIA (Kim et al., 2003), a biomedical named entity corpus.

Table 1 shows the entity classes and the number of entities of each class that are annotated in the three corpora. As shown in the table, for the GENIA corpus, we combined the DNA and RNA entities into a single named entity class. Meanwhile, for ACE2005, although top-level entity classes are divided into a number of different subtypes, we only considered the top-level classes, as shown in the table.

For active and proactive learning experiments, 1% and 20% of sentences of each corpus were used as the initial labelled set and the test set, respectively. The remaining 79% of sentences were regarded as unlabelled data.

3.2 Expert simulation

We simulated the reliable and fallible experts by using two machine learning models: LSTM-CRF (Lample et al., 2016)—a neural network NER and CRF (Lafferty et al., 2001). To evaluate the performance of the two models, we conducted preliminary experiments, by firstly trained the two models on 80% of the labelled corpora and subsequently testing them on the remaining 20% of the data.

Word embeddings As the three corpora belong to three different domains, we used three corresponding pre-trained word embeddings as input to the LSTM-CRF model.

- ACE2005: GoogleNews vectors[2], which include approximately 100 billion words.
- COPIOUS: we applied word2vec to the English subset of the Biodiversity Heritage Library[3] to learn vectors for biodiversity entities. The set has approximately 26 million pages with more than 8 billion words.

[2]http://code.google.com/archive/p/word2vec/
[3]http://www.biodiversitylibrary.org/

Corpus	Entity	Labelled	Unlabelled	Test	Total
ACE2005	Person (PER)	291	22853	5179	28323
	Organization (ORG)	36	4554	690	5280
	Geo-Political Entity (GPE)	21	5813	1360	7194
	Location (LOC)	7	760	168	935
	Facility (FAC)	5	1136	227	1368
	Weapon (WEA)	8	609	178	795
	Vehicle (VEH)	7	640	123	770
COPIOUS	Habitat	23	619	366	1008
	Taxon	116	4485	1728	6329
	Person	24	768	258	1050
	Geographical Location (GeoLoc)	42	4373	1942	6357
	Temporal Expression (TempExp)	20	904	358	1282
GENIA	DNA&RNA	88	6592	1757	8437
	Cell	133	9623	2437	12193
	Protein	316	24940	6402	31658

Table 1: Statistic information of the three corpora

- GENIA: word vectors trained on a combination of PubMed, PMC and English Wikipedia texts (Pyysalo et al., 2013).

CRF features To train the CRF model, we used CRF++[4] and employed following features: word base, lemma, part-of-speech tag and chunk tag of a token. We also used unigram and bigram features that combine the features of the previous, current and following token.

As illustrated in Table 2, the LSTM-CRF model is mostly more precise and achieves wider coverage than CRF. We therefore selected LSTM-CRF to simulate the reliable expert and CRF to simulate the fallible expert.

Corpus	CRF			LSTM-CRF		
	Pre.	Re.	F1	Pre.	Re.	F1
ACE2005	73.89	65.07	69.20	75.69	74.11	74.89
COPIOUS	81.01	48.58	60.74	77.18	74.77	75.96
GENIA	73.90	64.52	68.89	75.41	73.91	74.66

Table 2: Performance of CRF and LSTM-CRF on the three corpora

The reliable expert (the LSTM-CRF model) was trained on 80% of the labelled data, while the fallible one (the CRF model) was trained on 60%. The F_1 scores of the reliable and fallible experts when applied to the test dataset are presented in Table 3.

Corpus	Fallible	Reliable
ACE2005	61.19	74.89
COPIOUS	50.92	75.96
GENIA	57.67	74.66

Table 3: F_1 scores of each expert on the three corpora

[4]https://taku910.github.io/crfpp/

The class probability of each expert is precalculated based on the the F_1 score of each class that an expert can achieve on the 1% initial labelled set. Meanwhile, the sentence probability of each expert is estimated at each iteration.

3.3 Active learning criteria

Various active learning criteria were investigated using the three corpora. We firstly estimated the performance (F_1 score) of a supervised NER model by using CRF++ and the above-mentioned features. We then compared the performance of each active learning criterion with that of the supervised model. If the performance of one criterion approximates that of the supervised with the least number of iterations, we consider the criterion as the best one for proactive learning experiments.

We experimented with the following criteria: least confidence (Culotta and McCallum, 2005), normalized entropy (Kim et al., 2006), MMR (Maximal Marginal Relevance) (Kim et al., 2006), density (Settles and Craven, 2008) when using feature vectors and word embeddings, and the combination of least confidence and density criterion. Equation 8 describes the combination criterion used in our experiments. In this equation, UL is the current unlabelled dataset, $\boldsymbol{x}^u$ is the u^{th} unlabelled sentence in UL, the parameter $\lambda = 0.8$, and the similarity score (Settles and Craven, 2008) were calculated by using feature vectors.

$$\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}}(\lambda * Least\_Confidence(\boldsymbol{x}) + (1-\lambda)\frac{1}{|UL|}\sum_{u=1}^{|UL|} similarity(\boldsymbol{x}, \boldsymbol{x}^u)) \quad (8)$$

Corpus	Entity Class	Best Criterion
ACE2005	PER	Density (w2v)
	ORG	Density (f2v)
	GPE	Entropy
	LOC	Least Confidence
	FAC	Longest
	WEA	MMR
	VEH	Longest
		(Overall) Entropy
COPIOUS	Habitat	Density (f2v)
	Taxon	Entropy
	Person	Density (f2v)
	GeoLoc	Entropy
	TempExp	Least Confidence
		(Overall) Entropy
GENIA	Protein	Entropy
	Cell	LC+Density (f2v)
	DNA&RNA	Entropy
		(Overall) Entropy

Table 4: The best active learning criteria on the three corpora

We also implemented two baseline criteria. The first one is random selection, in which a batch of sentences is selected randomly at each iteration. The second one, namely $longest$, is a criterion that selects the longest sentences to be labelled.

Among these criteria, we selected the best criterion for further experiments. The best criterion is the one that produced competitive or better performance (F-score) than that of a supervised learning method with the least number of training instances. We report these criteria for each entity class as well as for the overall corpus in Table 4. In this table, Density (f2v) and Density (w2v) represent the density criteria when using feature and word vectors, respectively. Entropy is the normalized entropy. LC+Density is the combined criterion, described in Equation 8. As shown in the table, the best criteria at the level of individual classes are diverse. However, overall, normalized entropy is the best criterion for all three corpora. We therefore selected this criterion in our proactive learning experiments.

3.4 Proactive learning results

Our method was evaluated on the test datasets of the three corpora mentioned in Section 3.1. For all experiments with proactive learning, we used the following settings: $\alpha = 0.975$, $\gamma = 0.05$, $N = 200$, and the annotation costs are 3 and 1 per sentence for the reliable and fallible experts, respectively.

3.4.1 $BatchSize$

We investigated different values of $BatchSize$ including 20, 10, 5, and 1. The results when $BatchSize$ is 1 was not shown in Figure 1 as our method always selects the fallible expert at every iteration, which results in a performance that is inferior to the baselines. For the GENIA corpus, the F-scores are comparable, regardless of the $BatchSize$ used. Meanwhile, for the ACE2005 corpus, the F-scores are the highest when the batch size is 20. In contrast, for the COPIOUS corpus, the best scores are obtained with a batch size of 10.

3.4.2 Comparison with baselines

Figure 2 compares the experimental results of the two baseline methods ($Reliable$ and $Fallible$) and the best performance of the proposed proactive learning method (PA) with batch sizes of 20, 10, and 5, respectively, on the three corpora. $Reliable$ refers to a baseline in which we only select the reliable expert at each iteration. Similarly, only the fallible expert was selected in the $Fallible$ experiments.

It can be seen that the performance of the three models is comparable between ACE2005 and the COPIOUS corpus. For these two corpora, PA outperformed the two baselines. In most cases, by using PA, better F-scores are obtained at the same cost as the two baselines. Both PA and $Reliable$ performance is increased when the total cost is increased. Meanwhile, for the $Fallible$ model, the performance stabilises at a lower level than the other methods when cost rises above a certain level.

Regarding the GENIA corpus, PA acheived a higher performance than $Reliable$, but a lower performance than $Fallible$ in the range of costs from 0 to approximately 3,500. This can be partly explained by the fact that there are only three NE classes in this corpus. Hence, the annotation task is simpler than for the the other corpora, even for the fallible expert. However, when the cost is greater than 3,500, the performance of $Fallible$ becomes stable, while the performance of PA continues to increase.

We also investigated the number of times that each expert was selected during the iterative process of PA. The results are shown in Figure 3. PA ($Reliable$) and PA ($Fallible$) correspond to number of times that the reliable and fallible expert respectively, were selected in PA, while

Figure 1: Pro-active learning results on the three corpora when using different $BatchSize$

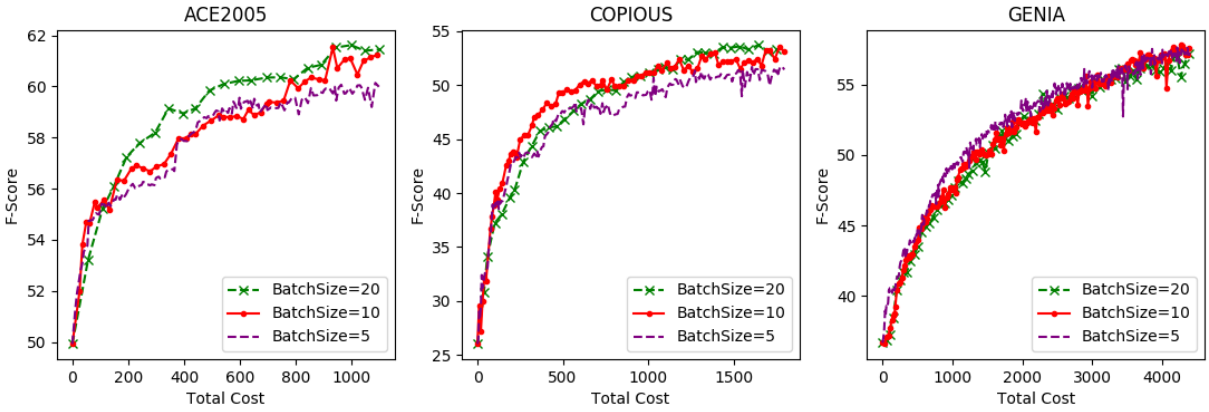
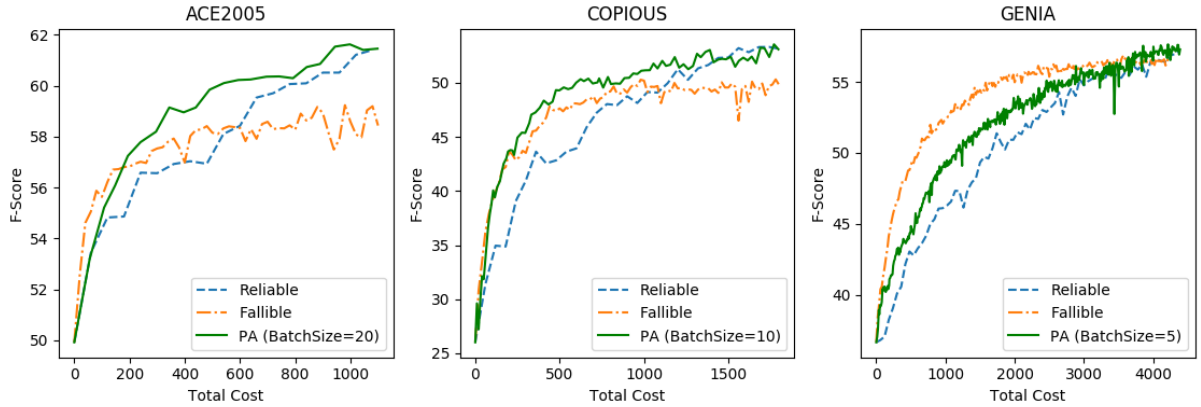


Figure 2: The best pro-active learning results on the three corpora in comparison to the baselines



Reliable corresponds to the number of times that the reliable expert was selected in *Reliable* baseline experiment. The figure illustrates that the number of times that the fallible expert is selected grows continually as the number of iterations increases. This shows that our method can effectively distribute appropriate unlabelled sentences to the fallible expert, in order to save on annotation costs.

4 Related work

4.1 Active learning for NER

Active learning aims to decrease annotation cost, whilst maintaining acceptable quality of annotated data. To achieve this, the method iteratively selects the most informative sentences to be annotated from an unlabelled data set.

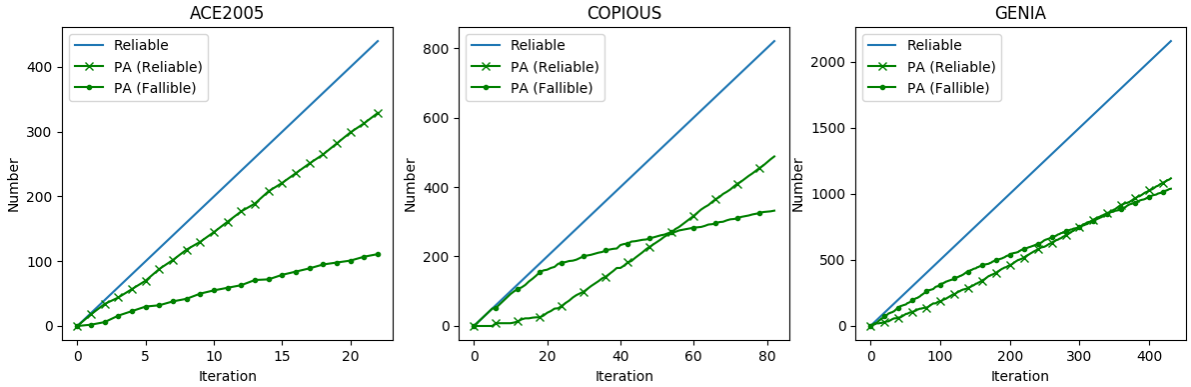
One of the most common selection criteria used in applying active learning to the task of NE labelling is the uncertainty-based criterion. This criterion assumes that the most uncertain sentence is the most useful instance for learning an NER model. There are several ways to implement this, such as least confidence (Culotta and McCallum, 2005)–the lower the probability of a sequence of labels, the less confidence the model, and entropy (Kim et al., 2006) that can measure the uncertainty of a probability distribution. Some other criteria are a diversity measurement (Kim et al., 2006) and a density criterion (Settles and Craven, 2008).

4.2 Cost-sensitive active learning

Cost-sensitive active learning is a type of active learning method that considers the annotation cost, e.g., budget, time or effort required to complete the annotation process (Olsson, 2009). Since proactive learning also models the reliability or expertise of each annotator in addition to the annotation cost, it can be considered as another case of cost-sensitive active learning.

Donmez and Carbonell (2008, 2010) investigated proactive learning for binary classification.

Figure 3: Number of times that each expert is selected in PA and $Reliable$ models



They predicted the probability that a reluctant oracle refuses to annotate an instance and the probability that a fallible oracle assigns a random label to an instance. Each oracle charges a different amount for their efforts. They also proposed a model that assigns different costs to unlabelled instances according to their annotation difficulty. For the multi-class classification task, Moon and Carbonell (2014) used the same approach but they had multiple experts, each of whom is specialised for each class. Kapoor et al. (2007) proposed a decision-theoretic method for the task of voice mail classification. They defined a criterion named "expected value-of-information" that combines the misclassification risk with the labelling cost.

Cost-sensitive active learning was also applied to part-of-speech (POS) tagging (Haertel et al., 2008). In this work, an hourly cost measurement was determined and a linear regression model was trained to predict the annotation cost. Hwa (2000) aimed to reduce the manual effort for a parsing task by using tree entropy cost. Meanwhile, Baldridge and Osborne (2004) measured the total annotation cost to create a treebank by using unit cost and discriminant cost.

5 Conclusion and future work

Our work constitutes the first attempt to use proactive learning method for named entity labelling. We simulated the behaviour of reliable and fallible experts having different levels of expertise and different costs. To save annotation costs and to ensure acceptable quality of the resulting annotated data, the method favours the selection of the fallible expert. In order to increase efficiency, we also proposed a batch sampling algorithm to select more than one sentence in each iteration. Experimental results for three corpora belonging to different domains demonstrate that the employment of non-perfect experts can help to build gold standard dataset at reasonable cost. Moreover, our method performed well across the three different corpora, demonstrating the generality of our approach.

A potential limitation of our approach is that the initial step is reliant on the availability of a gold standard corpus to estimate the experts' performance. However, for some domains, it may be difficult to obtain such a dataset. Therefore, as future work, we will explore how we can assess experts' performance without the need for gold-standard labelled data.

As a further extension to our work, we will explore the deployment of our method on crowd sourcing platforms, such as CrowdFlower[5] and Amazon Mechanical Turk[6]. These platforms allow annotations to be obtained from non-expert annotators in a rapid and cost-effective manner (Snow et al., 2008). These non-experts can be treated as non-perfect annotators in our proposed proactive learning method.

Acknowledgement

We would like to thank Paul Thompson for his valuable comments. This work is partially funded by the British Council (COPIOUS 172722806) and the Biotechnology and Biological Sciences Research Council, UK (EMPATHY, Grant No. BB/M006891/1).

[5]https://www.crowdflower.com/
[6]https://www.mturk.com/mturk/welcome

References

Jason Baldridge and Miles Osborne. 2004. Active Learning and the Total Cost of Annotation. In *EMNLP*. pages 9–16.

Aron Culotta and Andrew McCallum. 2005. Reducing labeling effort for structured prediction tasks. In *AAAI*. volume 5, pages 746–51.

Pinar Donmez and Jaime G Carbonell. 2008. Proactive learning: cost-sensitive active learning with multiple imperfect oracles. In *Proceedings of the 17th ACM conference on Information and knowledge management*. ACM, pages 619–628.

Pinar Donmez and Jaime G Carbonell. 2010. From active to proactive learning methods. In *Advances in Machine Learning I*, Springer, pages 97–120.

Robbie Haertel, Eric Ringger, Kevin Seppi, James Carroll, and Peter McClanahan. 2008. Assessing the costs of sampling methods in active learning for annotation. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*. Association for Computational Linguistics, pages 65–68.

Rebecca Hwa. 2000. Sample selection for statistical grammar induction. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13*. Association for Computational Linguistics, pages 45–52.

Ashish Kapoor, Eric Horvitz, and Sumit Basu. 2007. Selective Supervision: Guiding Supervised Learning with Decision-Theoretic Active Learning. In *IJCAI*. volume 7, pages 877–882.

J-D Kim, Tomoko Ohta, Yuka Tateisi, and Junichi Tsujii. 2003. GENIA corpus–a semantically annotated corpus for bio-textmining. *Bioinformatics* 19(suppl 1):i180–i182.

Seokhwan Kim, Yu Song, Kyungduk Kim, Jeong-Won Cha, and Gary Geunbae Lee. 2006. Mmr-based active machine learning for bio named entity recognition. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*. Association for Computational Linguistics, pages 69–72.

John Lafferty, Andrew McCallum, Fernando Pereira, et al. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML*. volume 1, pages 282–289.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural Architectures for Named Entity Recognition. In Kevin Knight, Ani Nenkova, and Owen Rambow, editors, *HLT-NAACL*. The Association for Computational Linguistics, pages 260–270.

David D Lewis. 1995. Evaluating and optimizing autonomous text classification systems. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pages 246–254.

Seungwhan Moon and Jaime G Carbonell. 2014. Proactive learning with multiple class-sensitive labelers. In *Data Science and Advanced Analytics (DSAA), 2014 International Conference on*. IEEE, pages 32–38.

Fredrik Olsson. 2009. A literature survey of active machine learning in the context of natural language processing. Swedish Institute of Computer Science.

Sampo Pyysalo, Filip Ginter, Hans Moen, Tapio Salakoski, and Sophia Ananiadou. 2013. Distributional semantics resources for biomedical text processing. In *Proceedings of LBM 2013*. LBM, pages 39–44.

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

Burr Settles and Mark Craven. 2008. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics, pages 1070–1079.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y Ng. 2008. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics, pages 254–263.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. ACE 2005 multilingual training corpus. *Linguistic Data Consortium, Philadelphia* 57.