Accelerating the Annotation of Sparse Named Entities by Dynamic Sentence Selection

Yoshimasa Tsuruoka[1], **Jun'ichi Tsujii**[1,2,3] **and Sophia Ananiadou**[1,3]
[1] School of Computer Science, The University of Manchester, UK
[2] Department of Computer Science, The University of Tokyo, Japan
[3] National Centre for Text Mining (NaCTeM), Manchester, UK
yoshimasa.tsuruoka@manchester.ac.uk
tsujii@is.s.u-tokyo.ac.jp
sophia.ananiadou@manchester.ac.uk

Abstract

This paper presents an active learning-like framework for reducing the human effort for making named entity annotations in a corpus. In this framework, the annotation work is performed as an iterative and interactive process between the human annotator and a probabilistic named entity tagger. At each iteration, sentences that are most likely to contain named entities of the target category are selected by the probabilistic tagger and presented to the annotator. This iterative annotation process is repeated until the estimated coverage reaches the desired level. Unlike active learning approaches, our framework produces a named entity corpus that is free from the sampling bias introduced by the active strategy. We evaluated our framework by simulating the annotation process using two named entity corpora and show that our approach could drastically reduce the number of sentences to be annotated when applied to sparse named entities.

1 Introduction

Named entities play a central role in conveying important domain specific information in text, and good named entity recognizers are often required in building practical information extraction systems. Previous studies have shown that automatic named entity recognition can be performed with a reasonable level of accuracy by using various machine learning models such as support vector machines (SVMs) or conditional random fields (CRFs) (Tjong Kim Sang and De Meulder, 2003; Settles, 2004; Okanohara et al., 2006).

However, the lack of annotated corpora, which are indispensable for training machine learning models, makes it difficult to broaden the scope of text mining applications. In the biomedical domain, for example, several annotated corpora such as GENIA (Kim et al., 2003), PennBioIE (Kulick et al., 2004), and GENETAG (Tanabe et al., 2005) have been created and made publicly available, but the named entity categories annotated in these corpora are tailored to their specific needs and not always sufficient or suitable for text mining tasks that other researchers need to address.

Active learning is a framework which can be used for reducing the amount of human effort required to create a training corpus (Dagan and Engelson, 1995; Engelson and Dagan, 1996; Thompson et al., 1999; Shen et al., 2004). In active learning, samples that need to be annotated by the human annotator are picked up by a machine learning model in an iterative and interactive manner, considering the informativeness of the samples. Active learning has been shown to be effective in several natural language processing tasks including named entity recognition.

The problem with active learning is, however, that the resulting annotated data is highly dependent on the machine learning algorithm and the sampling strategy employed, because active learning annotates only a *subset* of the given corpus. This sampling bias is not a serious problem if one is to use the annotated corpus only for their own machine learning purpose and with the same machine learning algorithm. However, the existence of bias is not desirable if one also wants the corpus to be used by other applications or researchers. For the same reason, ac-

BioNLP 2008: Current Trends in Biomedical Natural Language Processing, pages 30–37,
Columbus, Ohio, USA, June 2008.

tive learning approaches cannot be used to enrich an existing linguistic corpus with a new named entity category.

In this paper, we present a framework that enables one to make named entity annotations for a given corpus with a reduced cost. Unlike active learning approaches, our framework aims to annotate *all* named entities of the target category contained in the corpus. Obviously, if we were to ensure 100% coverage of annotation, there is no way of reducing the annotation cost, i.e. the human annotator has to go through every sentence in the corpus. However, we show in this paper that it is possible to reduce the cost by slightly relaxing the requirement for the coverage, and the reduction can be drastic when the target named entities are sparse.

We should note here that the purpose of this paper is not to claim that our approach is superior to existing active learning approaches. The goals are different—while active learning aims at optimizing the performance of the resulting machine learning-based tagger, our framework aims to help develop an unbiased named entity-annotated corpus.

This paper is organized as follows. Section 2 describes the overall annotation flow in our framework. Section 3 presents how to select sentences using the output of a probabilistic tagger. Section 4 describes how to estimate the coverage during the course of annotation. Experimental results using two named entity corpora are presented in section 5. Section 6 describes related work and discussions. Concluding remarks are given in section 7.

2 Annotating Named Entities by Dynamic Sentence Selection

Figure 1 shows the overall flow of our annotation framework. The framework is an iterative process between the human annotator and a named entity tagger based on CRFs. In each iteration, the CRF tagger is trained using all annotated sentences available and is applied to the unannotated sentences to select sentences that are likely to contain named entities of the target category. The selected sentences are then annotated by the human annotator and moved to the pool of annotated sentences.

This overall flow of annotation framework is very similar to that of active learning. In fact, the only differences are the criterion of sentence selection and the fact that our framework uses the estimated coverage as the stopping condition. In active learning, sentences are selected according to their informativeness to the machine learning algorithm. Our approach, in contrast, selects sentences that are most likely to contain named entities of the target category. Section 3 elaborates on how to select sentences using the output of the CRF-based tagger.

1. Select the first n sentences from the corpus and annotate the named entities of the target category.
2. Train a CRF tagger using all annotated sentences.
3. Apply the CRF tagger to the unannotated sentences in the corpus and select the top n sentences that are most likely to contain target named entities.
4. Annotate the selected sentences.
5. Go back to 2 (repeat until the estimated coverage reaches a satisfactory level).

Figure 1: Annotating named entities by dynamic sentence selection.

The other key in this annotation framework is when to stop the annotation work. If we repeat the process until all sentences are annotated, then obviously there is not merit of using this approach. We show in section 4 that we can quite accurately estimate how much of the entities in the corpus are already annotated and use this estimated coverage as the stopping condition.

3 Selecting Sentences using the CRF tagger

Our annotation framework takes advantage of the ability of CRFs to output multiple probabilistic hypotheses. This section describes how we obtain named entity candidates and their probabilities from CRFs in order to compute the expected number of named entities contained in a sentence [1].

[1]We could use other machine learning algorithms for this purpose as long as they can produce probabilistic output. For

3.1 The CRF tagger

CRFs (Lafferty et al., 2001) can be used for named entity recognition by representing the spans of named entities using the "BIO" tagging scheme, in which 'B' represents the beginning of a named entity, 'I' the inside, and 'O' the outside (See Table 2 for example). This representation converts the task of named entity recognition into a sequence tagging task.

A linear chain CRF defines a single log-linear probabilistic distribution over the possible tag sequences $\mathbf{y}$ for a sentence $\mathbf{x}$:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp \sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(t, y_t, y_{t-1}, \mathbf{x_t}),$$

where $f_k(t, y_t, y_{t-1}, \mathbf{x_t})$ is typically a binary function indicating the presence of feature k, λ_k is the weight of the feature, and $Z(X)$ is a normalization function:

$$Z(\mathbf{x}) = \sum_{y} \exp \sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(t, y_t, y_{t-1}, \mathbf{x_t}).$$

This modeling allows us to define features on states ("BIO" tags) and edges (pairs of adjacent "BIO" tags) combined with observations (e.g. words and part-of-speech (POS) tags).

The weights of the features are determined in such a way that they maximize the conditional log-likelihood of the training data[2] $\mathcal{L}(\theta) = \sum_{i=1}^{N} \log p_\theta(\mathbf{y}^{(i)}|\mathbf{x}^{(i)})$. We use the L-BFGS algorithm (Nocedal, 1980) to compute those parameters.

Table 1 lists the feature templates used in the CRF tagger. We used unigrams of words/POS tags, and prefixes and suffixes of the current word. The current word is also normalized by lowering capital letters and converting all numerals into '#', and used as a feature. We created a word shape feature from the current word by converting consecutive capital letters into 'A', small letters 'a', and numerals '#'.

example, maximum entropy Markov models are a possible alternative. We chose the CRF model because it has been proved to deliver state-of-the-art performance for named entity recognition tasks by previous studies.

[2]In the actual implementation, we used L2 norm penalty for regularization.

Word Unigram	w_i, w_{i-1}, w_{i+1}	& y_i
POS Unigram	p_i, p_{i-1}, p_{i+1}	& y_i
Prefix, Suffix	prefixes of w_i	& y_i
	suffixes of w_i	& y_i
	(up to length 3)	
Normalized Word	N(w_i)	& y_i
Word Shape	S(w_i)	& y_i
Tag Bi-gram	true	& $y_{i-1}y_i$

Table 1: Feature templates used in the CRF tagger.

3.2 Computing the expected number of named entities

To select sentences that are most likely to contain named entities of the target category, we need to obtain the *expected number* of named entities contained in each sentence. CRFs are well-suited for this task as the output is fully probabilistic.

Suppose, for example, that the sentence is "Transcription factor GATA-1 and the estrogen receptor". Table 2 shows an example of the 5-best sequences output by the CRF tagger. The sequences are represented by the aforementioned "BIO" representation. For example, the first sequence indicates that there is one named entity 'Transcription factor' in the sequence. By summing up these probabilistic sequences, we can compute the probabilities for possible named entities in a sentence. From the five sequences in Table 2, we obtain the following three named entities and their corresponding probabilities.

'Transcription factor' (0.677 + 0.242 = 0.916)
'estrogen receptor' (0.242 + 0.009 = 0.251)
'Transcription factor GATA-1' (0.012 + 0.009 = 0.021)

The expected number of named entities in this sentence can then be calculated as 0.916 + 0.251 + 0.021 = 1.188.

In this example, we used 5-best sequences as an approximation of all possible sequences output by the tagger, which are needed to compute the exact expected number of entities. One possible way to achieve a good approximation is to use a large N for N-best sequences, but there is a simpler and more efficient way [3], which directly produces the exact

[3]We thank an anonymous reviewer for pointing this out.

Probability	Transcription	factor	GATA-1	and	the	estrogen	receptor
0.677	B	I	O	O	O	O	O
0.242	B	I	O	O	O	B	I
0.035	O	O	O	O	O	O	O
0.012	B	I	I	O	O	O	O
0.009	B	I	I	O	O	B	I
:	:	:	:	:	:	:	:

Table 2: N-best sequences output by the CRF tagger.

expected number of entities. Recall that named entities are represented with the "BIO" tags. Since one entity always contains one "B" tag, we can compute the number of expected entities by simply summing up the marginal probabilities for the "B" tag on each token in the sentence[4].

Once we compute the expected number of entities for every unannotated sentence in the corpus, we sort the sentences in descending order of the expected number of entities and choose the top n sentences to be presented to the human annotator.

4 Coverage Estimation

To ensure the quality of the resulting annotated corpus, it is crucial to be able to know the current coverage of annotation at each iteration in the annotation process. To compute the coverage, however, one needs to know the total number of target named entities in the corpus. The problem is that it is not known until all sentences are annotated.

In this paper, we solve this dilemma by using an estimated value for the total number of entities. Then, the estimated coverage can be computed as follows:

$$(estimated\_coverage) = \frac{m}{m + \sum_{i \in U} E_i} \quad (1)$$

where m is the number of entities actually annotated so far and E_i is the expected number of entities in sentence i, and U is the set of unannotated sentences in the corpus. At any iteration, m is always known and E_i is obtained from the output of the CRF tagger as explained in the previous section.

[4]The marginal probabilities on each token can be computed by the forward-backward algorithm, which is much more efficient than computing N-best sequences for a large N.

	# Entities	Sentences (%)
CoNLL: LOC	7,140	5,127 (36.5%)
CoNLL: MISC	3,438	2,698 (19.2%)
CoNLL: ORG	6,321	4,587 (32.7%)
CoNLL: PER	6,600	4,373 (31.1%)
GENIA: DNA	2,017	5,251 (28.3%)
GENIA: RNA	225	810 (4.4%)
GENIA: cell_line	835	2,880 (15.5%)
GENIA: cell_type	1,104	5,212 (28.1%)
GENIA: protein	5,272	13,040 (70.3%)

Table 3: Statistics of named entities.

5 Experiments

We carried out experiments to see how our method can improve the efficiency of annotation process for sparse named entities. We evaluate our method by simulating the annotation process using existing named entity corpora. In other words, we use the gold-standard annotations in the corpus as the annotations that would be made by the human annotator during the annotation process.

5.1 Corpus

We used two named entity corpora for the experiments. One is the training data provided for the CoNLL-2003 shared task (Tjong Kim Sang and De Meulder, 2003), which consists of 14,041 sentences and includes four named entity categories (LOC, MISC, ORG, and PER) for the general domain. The other is the training data provided for the NLPBA shared task (Kim et al., 2004), which consists of 18,546 sentences and five named entity categories (DNA, RNA, cell_line, cell_type, and protein) for the biomedical domain. This corpus is created from the GENIA corpus (Kim et al., 2003) by merging the original fine-grained named entity categories.

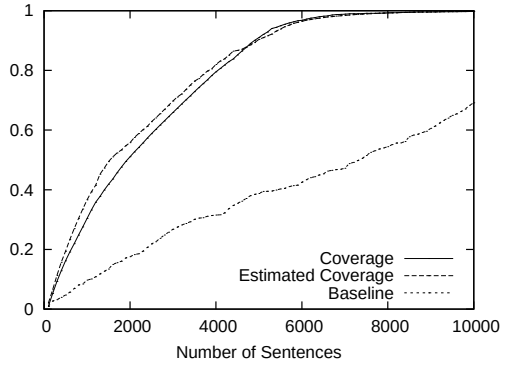

Figure 2: Annotation of LOC in the CoNLL corpus.

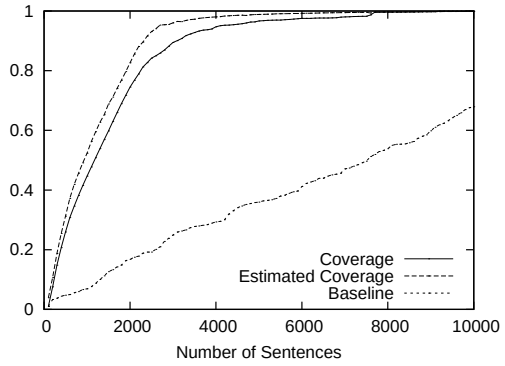

Figure 3: Annotation of MISC in the CoNLL corpus.

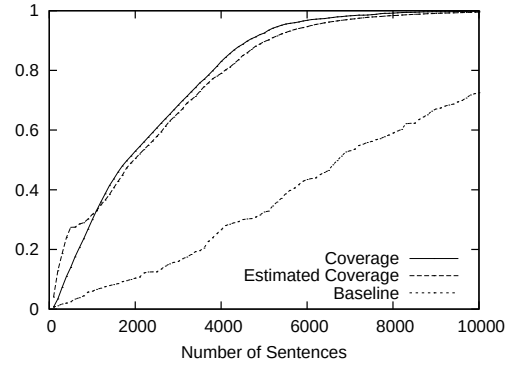

Figure 4: Annotation of ORG in the CoNLL corpus.

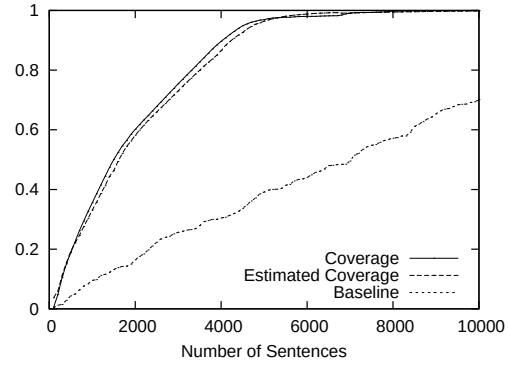

Figure 5: Annotation of PER in the CoNLL corpus.

Table 3 shows statistics of the named entities included in the corpora. The first column shows the number of named entities for each category. The second column shows the number of the sentences that contain the named entities of each category. We can see that some of the named entity categories are very sparse. For example, named entities of "RNA" appear only in 4.4% of the sentences in the corpus. In contrast, named entities of "protein" appear in more than 70% of the sentences in the corpus.

In the experiments reported in the following sections, we do not use the "protein" category because there is no merit of using our framework when most sentences are relevant to the target category.

5.2 Results

We carried out eight sets of experiments, each of which corresponds to one of those named entity categories shown in Table 3 (excluding the "protein" category). The number of sentences selected in each iteration (the value of n in Figure 1) was set to 100 throughout all experiments.

Figures 2 to 5 show the results obtained on the CoNLL data. The figures show how the coverage increases as the annotation process proceeds. The x-axis shows the number of annotated sentences.

Each figure contains three lines. The normal line represents the coverage actually achieved, which is computed as follows:

$$(coverage) = \frac{entities\_annotated}{total\_number\_of\_entities}. \quad (2)$$

The dashed line represents the coverage estimated by using equation 1. For the purpose of comparison, the dotted line shows the coverage achieved by the baseline annotation strategy in which sentences are selected sequentially from the beginning to the end in the corpus.

The figures clearly show that our method can drastically accelerate the annotation process in comparison to the baseline annotation strategy. The improvement is most evident in Figure 3, in which

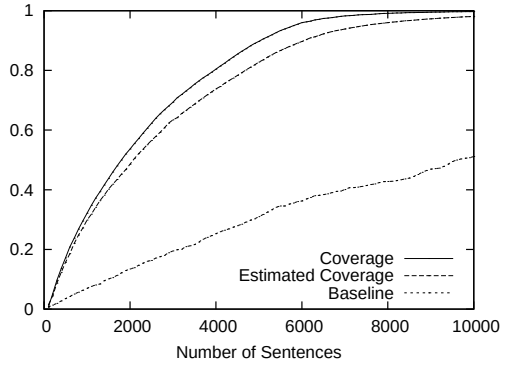

Figure 6: Annotation of DNA in the GENIA corpus.

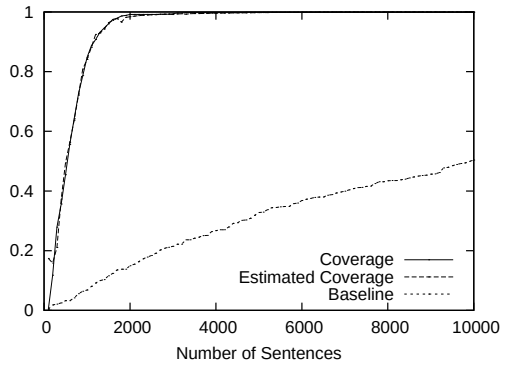

Figure 7: Annotation of RNA in the GENIA corpus.

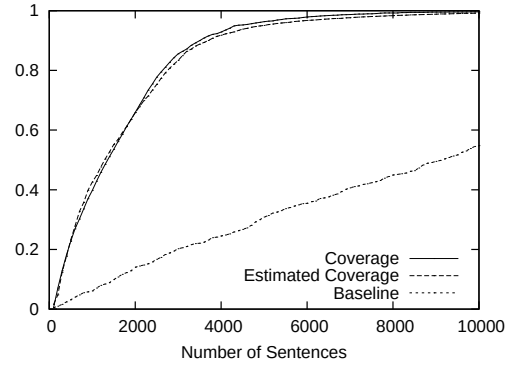

Figure 8: Annotation of cell_line in the GENIA corpus.

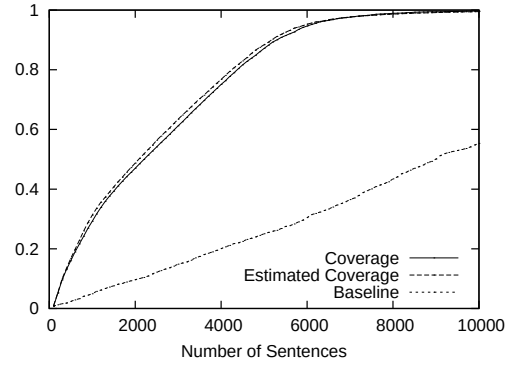

Figure 9: Annotation of cell_type in the GENIA corpus.

named entities of the category "MISC" are annotated.

We should also note that coverage estimation was surprisingly accurate. In all experiments, the difference between the estimated coverage and the real coverage was very small. This means that we can safely use the estimated coverage as the stopping condition for the annotation work.

Figures 6 to 9 show the experimental results on the GENIA data. The figures show the same characteristics observed in the CoNLL data. The acceleration by our framework was most evident for the "RNA" category.

Table 4 shows how much we can save the annotation cost if we stop the annotation process when the estimated coverage reaches 99%. The first column shows the coverage actually achieved and the second column shows the number and ratio of the sentences annotated in the corpus. This table shows that, on average, we can achieve a coverage of 99.0% by annotating 52.4% of the sentences in the corpus. In other words, we could roughly halve the annotation cost by accepting the missing rate of 1.0%.

As expected, the cost reduction was most drastic when "RNA", which is the most sparse named entity category (see Table 3), was targeted. The cost reduction was more than seven-fold. These experimental results confirm that our annotation framework is particularly useful when applied to sparse named entities.

Table 4 also shows the timing information on the experiments [5]. One of the potential problems with this kind of active learning-like framework is the computation time required to retrain the tagger at each iteration. Since the human annotator has to wait while the tagger is being retrained, the computation time required for retraining the tagger should not be very long. In our experiments, the worst case (i.e. DNA) required 443 seconds for retraining the tagger at the last iteration, but in most cases

[5] We used AMD Opteron 2.2GHz servers for the experiments and our CRF tagger is implemented in C++.

	Coverage	Sentences Annotated (%)	Cumulative Time (second)	Last Interval (second)
CoNLL: LOC	99.1%	7,600 (54.1%)	3,362	92
CoNLL: MISC	96.9%	5,400 (38.5%)	1,818	61
CoNLL: ORG	99.7%	8,900 (63.4%)	5,201	104
CoNLL: PER	98.0%	6,200 (44.2%)	2,300	75
GENIA: DNA	99.8%	11,900 (64.2%)	33,464	443
GENIA: RNA	99.2%	2,500 (13.5%)	822	56
GENIA: cell_line	99.6%	9,400 (50.7%)	15,870	284
GENIA: cell_type	99.3%	8,600 (46.4%)	13,487	295
Average	99.0%	- (52.4%)	-	-

Table 4: Coverage achieved when the estimated coverage reached 99%.

the training time for each iteration was kept under several minutes.

In this work, we used the BFGS algorithm for training the CRF model, but it is probably possible to further reduce the training time by using more recent parameter estimation algorithms such as exponentiated gradient algorithms (Globerson et al., 2007).

6 Discussion and Related Work

Our annotation framework is, by definition, not something that can ensure a coverage of 100%. The seriousness of a missing rate of, for example, 1% is not entirely clear—it depends on the application and the purpose of annotation. In general, however, it is hard to achieve a coverage of 100% in real annotation work even if the human annotator scans through all sentences, because there is often ambiguity in deciding whether a particular named entity should be annotated or not. Previous studies report that inter-annotator agreement rates with regards to gene/protein name annotation are f-scores around 90% (Morgan et al., 2004; Vlachos and Gasperin, 2006). We believe that the missing rate of 1% can be an acceptable level of sacrifice, given the cost reduction achieved and the unavoidable discrepancy made by the human annotator.

At the same time, we should also note that our framework could be used in conjunction with existing methods for semi-supervised learning to improve the performance of the CRF tagger, which in turn will improve the coverage. It is also possible to improve the performance of the tagger by using external dictionaries or using more sophisticated probabilistic models such as semi-Markov CRFs (Sarawagi and Cohen, 2004). These enhancements should further improve the coverage, keeping the same degree of cost reduction.

The idea of improving the efficiency of annotation work by using automatic taggers is certainly not new. Tanabe et al. (2005) applied a gene/protein name tagger to the target sentences and modified the results manually. Culotta and McCallum (2005) proposed to have the human annotator select the correct annotation from multiple choices produced by a CRF tagger for each sentence. Tomanek et al. (2007) discuss the reusability of named entity-annotated corpora created by an active learning approach and show that it is possible to build a corpus that is useful to different machine learning algorithms to a certain degree.

The limitation of our framework is that it is useful only when the target named entities are sparse because the upper bound of cost saving is limited by the proportion of the relevant sentences in the corpus. Our framework may therefore not be suitable for a situation where one wants to make annotations for named entities of many categories simultaneously (e.g. creating a corpus like GENIA from scratch). In contrast, our framework should be useful in a situation where one needs to modify or enrich named entity annotations in an existing corpus, because the target named entities are almost always sparse in such cases. We should also note that named entities in full papers, which recently started to attract much attention, tend to be more sparse than those in abstracts.

7 Conclusion

We have presented a simple but powerful framework for reducing the human effort for making name entity annotations in a corpus. The proposed framework allows us to annotate *almost* all named entities

of the target category in the given corpus without having to scan through all the sentences. The framework also allows us to know when to stop the annotation process by consulting the estimated coverage of annotation.

Experimental results demonstrated that the framework can reduce the number of sentences to be annotated almost by half, achieving a coverage of 99.0%. Our framework was particularly effective when the target named entities were very sparse.

Unlike active learning, this work enables us to create a named entity corpus that is free from the sampling bias introduced by the active learning strategy. This work will therefore be especially useful when one needs to enrich an existing linguistic corpus (e.g. WSJ, GENIA, or PennBioIE) with named entity annotations for a new semantic category.

Acknowledgment

This work is partially supported by BBSRC grant BB/E004431/1. The UK National Centre for Text Mining is sponsored by the JISC/BBSRC/EPSRC.

References

Aron Culotta and Andrew McCallum. 2005. Reducing labeling effort for structured prediction tasks. In *Proceedings of AAAI-05*, pages 746–751.

Ido Dagan and Sean P. Engelson. 1995. Committee-based sampling for training probabilistic classifiers. In *Proceedings of ICML*, pages 150–157.

Sean Engelson and Ido Dagan. 1996. Minimizing manual annotation cost in supervised training from corpora. In *Proceedings of ACL*, pages 319–326.

A. Globerson, T. Koo, X. Carreras, and M. Collins. 2007. Exponentiated gradient algorithms for log-linear structured prediction. In *Proceedings of ICML*, pages 305–312.

J.-D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. 2003. GENIA corpus–a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19 (Suppl. 1):180–182.

J.-D. Kim, T. Ohta, Y. Tsuruoka, Y. Tateisi, and N. Collier. 2004. Introduction to the bio-entity recognition task at JNLPBA. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA)*, pages 70–75.

Seth Kulick, Ann Bies, Mark Libeman, Mark Mandel, Ryan McDonald, Martha Palmer, Andrew Schein, and Lyle Ungar. 2004. Integrated annotation for biomedical information extraction. In *Proceedings of HLT-NAACL 2004 Workshop: Biolink 2004*, pages 61–68.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML*, pages 282–289.

Alexander A. Morgan, Lynette Hirschman, Marc Colosimo, Alexander S. Yeh, and Jeff B. Colombe. 2004. Gene name identification and normalization using a model organism database. *Journal of Biomedical Informatics*, 37:396–410.

Jorge Nocedal. 1980. Updating quasi-newton matrices with limited storage. *Mathematics of Computation*, 35(151):773–782.

Daisuke Okanohara, Yusuke Miyao, Yoshimasa Tsuruoka, and Jun'ichi Tsujii. 2006. Improving the scalability of semi-markov conditional random fields for named entity recognition. In *Proceedings of COLING/ACL*, pages 465–472.

Sunita Sarawagi and William W. Cohen. 2004. Semi-markov conditional random fields for information extraction. In *Proceedings of NIPS*.

Burr Settles. 2004. Biomedical named entity recognition using conditional random fields and rich feature sets. In *COLING 2004 International Joint workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP) 2004*, pages 107–110.

Dan Shen, Jie Zhang, Jian Su, Guodong Zhou, and Chew-Lim Tan. 2004. Multi-criteria-based active learning for named entity recognition. In *Proceedings of ACL*, pages 589–596, Barcelona, Spain.

Lorraine Tanabe, Natalie Xie, Lynne H. Thom, Wayne Matten, and W. John Wilbur. 2005. GENETAG: a tagged corpus for gene/protein named entity recognition. *BMC Bioinformatics*, 6(Suppl 1):S3.

Cynthia A. Thompson, Mary Elaine Califf, and Raymond J. Mooney. 1999. Active learning for natural language parsing and information extraction. In *Proceedings of ICML*, pages 406–414.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2003*, pages 142–147.

Katrin Tomanek, Joachim Wermter, and Udo Hahn. 2007. An approach to text corpus construction which cuts annotation costs and maintains reusability of annotated data. In *Proceedings of EMNLP-CoNLL*, pages 486–495.

Andreas Vlachos and Caroline Gasperin. 2006. Bootstrapping and evaluating named entity recognition in the biomedical domain. In *Proceedings of the HLT-NAACL BioNLP Workshop on Linking Natural Language and Biology*, pages 138–145.