Active Entity Recognition in Low Resource Settings

Ning Gao
Microsoft
nigao@microsoft.com

Nikos Karampatziakis
Microsoft
nikosk@microsoft.com

Rahul Potharaju
Microsoft
rapoth@microsoft.com

Silviu Cucerzan
Microsoft Research
silviu@microsoft.com

ABSTRACT

The task of Named Entity Recognition (NER) has been well studied under high-resource conditions (e.g., extracting named mentions of Person, Organization and Location from news articles). However, there are very few studies of the NER task for open-domain collections and in low-resource settings. We focus on NER for low-resource collections, in which any entity types of practical interest to the users of the system must be supported. We try to achieve this with a low cost of annotation of data from the target domain/collection. We propose an entity recognition framework that combines active learning and conditional random fields (CRF), and which provides the flexibility to define new entity types as needed by the users. Our experiments on a help & support corpus show that the system can achieve F_1 measure of 0.77 by relying on only 100 manually-annotated sentences.

CCS CONCEPTS

• **Information systems** → **Entity resolution**; • **Computing methodologies** → **Active learning settings**.

KEYWORDS

Named Entity Recognition, Active Learning, Open Domain, Low Resource Settings

ACM Reference Format:
Ning Gao, Nikos Karampatziakis, Rahul Potharaju, and Silviu Cucerzan. 2019. Active Entity Recognition in Low Resource Settings. In *The 28th ACM International Conference on Information and Knowledge Management (CIKM '19), November 3–7, 2019, Beijing, China.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3357384.3358109

1 INTRODUCTION

Named entity recognition is the task of locating mentions of entities in text and classifying them into pre-defined types that are of particular interest for information extraction. The most studied entity types are Person, Organization and Location, collectively known as ENAMEX types since the MUC-6 evaluation [6]. CoNLL 2003 [19] introduced the type Miscellaneous to anotate named entity mentions that fall outside of ENAMEX. Most NER systems (e.g., the Stanford Named Entity Recognizer [5]) also recognize TIMEX (types Date and Time) and NUMEX types (Money and Percent). Further, the Bio-entity recognition task at JNLPBA [8] was set to recognize the types of Protein, Dna, Rna, Cell Line, and Cell Type from a corpus of bioinformatics documents [7]. The resulted annotated collections (e.g., 30K annotations in CoNLL 2003, 60K annotations in Bio-entity task) have enabled researchers to continue system developing and model training in these domains.

NER has been traditionally framed as a sequence tagging problem in models using conditional random fields [5, 13, 18], maximum entropy models, and hidden Markov models [12, 20]. More recently, the approaches have shifted to neural network architectures [2, 4, 9, 10, 15]. Despite the empirical success in high-resource settings, NER is still an important open problem for low-resource collections in open-domains, mainly due to two challenges: (1) the lack of large annotated sets that state-of-the-art neural network models rely on; (2) the need of users to define and recognize new types of entities and concepts (e.g., Entity, Action, and Problem, as shown as in Figure 1), while existing public NER models [5, 16, 17] have not been trained for such settings. As with other NLP tasks, the data annotation cost is a bottleneck in training and deploying NER systems in practice.

To the best of our knowledge, there are no available annotated sets for open-domain NER. In this study, we focus on NER for low-resource collections, in which users are allowed to define any entity types of practical interest. We try to minimize the annotation effort for data from the target domain. In particular, we employ a collection of documents targeting IT help & support, which contains nearly 1 million sentences. The proposed system actively samples *high quality sentences* (as described in Section 2.3) from the collection for users to annotate. To achieve this, the system first processes the collection data by breaking it into *phrases* (as described in Section 2.1). Based on information-theory-centric measures, it identifies and clusters important phrases from the collection. In an attempt to maximize the benefits of sentence annotation, the following two principles are followed: (1) initially, sentences with high quality phrases from different clusters are selected; (2) iteratively, the sentences that cause the most controversial results from different trained models are further selected for annotation. The annotated sentences are fed into CRF models with a large set of hand-crafted features.

2 METHODOLOGY

This section is organized as follows: the system framework is introduced in 2.1, followed by the details on data processing in 2.2 and annotation sentence sampling criteria in 2.3. Features used in the CRF model are discussed in Section 2.4.

2.1 System Framework

Figure 2 shows the framework of the proposed active-learning system for entity recognition. The Stanford toolkit [11] is used to split each document from the text collection into sentences. Stopwords are then employed as delimiters to segment the text. To simplify

CIKM '19, November 3–7, 2019, Beijing, China

ACM ISBN 978-1-4503-6976-3/19/11…$15.00
https://doi.org/10.1145/3357384.3358109

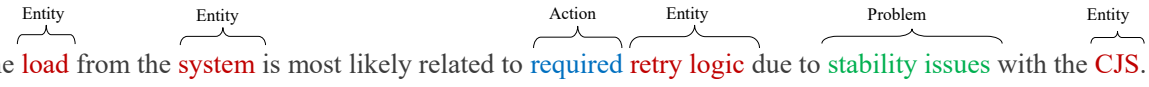

Figure 1: Sample sentence for the low-resource open-domain NER task. The sentence is selected from a corporate help & support corpus. ENTITY accounts for mentions of machines, systems, tasks, or projects; ACTION accounts for all the actions that can be performed on entities, including query, change, and configure; and PROBLEM accounts for reported or observed issues.

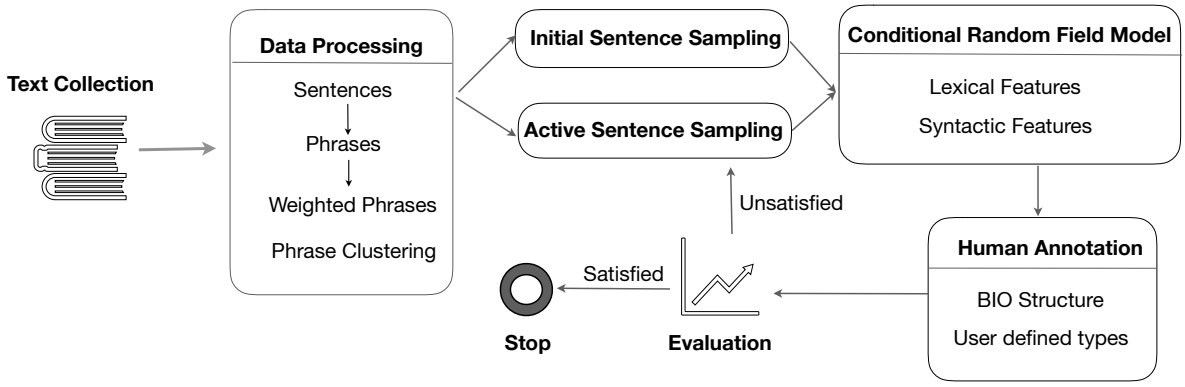

Figure 2: Framework of the proposed open-domain named entity recognition system.

the exposition, we refer to the obtained text segments as *phrases*. For example, the sentence in Figure 1, which is shown with the corresponding targeted annotation, is split into the phrases "load", "system", "likely related", "required retry logic due", "stability issues", and "CJS". Note that while some of the phrases correspond to entity mentions (e.g., "stability issues"), many others represent either incorrect segmentations of entity mentions (e.g., "required retry logic due") or spurious text segments (e.g., "likely related"). Nonetheless, this is not a severe issue because the simple phrase segmentation in this stage is done only to enable sampling high quality sentences for annotation. The final segmentations for mentions predicted by the system are not necessarily the same as phrases recognized in this phase. After segmentation, the phrases are weighted and clustered, as explained further in 2.2.

Each sentence is scored following the method presented in 2.3. An annotation pool is created by using the top 10k sentences according to the computed scores. Iteratively, the sentence with the highest current importance score is selected from the pool and presented for human annotation. Once annotated, the sentence is moved from the annotation pool to the training set for the CRF model (as presented in 2.4). When the annotated sentences in the training set cover all phrase clusters, the training set is randomly split into two subsets to train CRFs and obtain two different models. Sentences from the annotation pool are labeled in the order of importance with the two CRF models until a sentence with conflicting predictions is identified. The sentence is manually annotated and moved to the training pool, and the process is repeated. The active learning process stops when user is satisfied with the performance on sample dev/test or when annotation budget is reached. Human annotations follow the standard BIO (begin, inside, outside) structure. For each identified named entity mention, the annotator assigns a user defined type (e.g., ENTITY, PROBLEM, ACTION as defined in Figure 1).

2.2 Data Processing

Weighted pointwise mutual information (WPMI) is used to evaluate the collection-specific importance of the phrase, as explained further. Let N be the number of documents in the collection, $p = \{t_1, \ldots, t_m\}$ be the m terms in the phrase p, $f(t_i)$ be the number of documents that contain term t_i, and $f(t_1, \ldots, t_m)$ be the number of documents that contain phrase t, then WPMI is defined as a variant of PMI [3] as follows:

$$\text{WPMI}\ (p) = \frac{1}{m}\ log\ \frac{Pr(t_1, \ldots, t_m)}{\Pi_{i \in (1,m)} Pr(t_i)}, \tag{1}$$

where

$$Pr(t_i) = \frac{f(t_i)}{N}, \tag{2}$$

$$\text{and } Pr(t_1, \ldots, t_m) = \frac{f(t_1, \ldots, t_m)}{N}. \tag{3}$$

WPMI measures the importance of each phrase to a specific collection. K-means++ [1] is used to cluster the phrases and learn different types of foci of the collection. In our investigation, we set the number of clusters to 20. The distance between phrases is calculated by using cosine similarity on phrase embeddings derived additively from Word2Vec [14]. If a cluster contains fewer than five phrases, its center is replaced by a new center selected following the K-means++ initial center selection rule.

2.3 Sentence Sampling

There are two strategies for sentence sampling: the initial sampling and iterative sampling. The purpose of initial sentence sampling is to present to annotators high quality sentences that contain diverse aspects of the knowledge in the collection. *High quality* is measured by WPMI, while *aspect diversity* is targeted through phrase clustering. Formally, let $C = \{c_1, \ldots, c_k\}$ be the k clusters with each assigned with a cluster weight $W = \{w_{c_1}, \ldots, w_{c_k}\}$, where each $w_{c_i} \in W$ is set as 1 for initialization. The importance of a sentence S with phrases $P = \{p\}$ is computed as:

$$S = \sum_{c_i \in C}^{i \in [1,k]} w_{c_i} * \text{Max}(\text{WPMI}(p : p \in c_i)). \quad (4)$$

The sentence with highest importance is selected for initial human annotation. To avoid sampling bias on large clusters, after each sentence annotation, the weights for the clusters of phrases that occur in the selected sentence (i.e., $\text{Max}(\text{WPMI}(p : p \in c_i)) > 0$) are adjusted by using a 0.5 decay:

$$w_{c_i} = 0.5 * w_{c_i}. \quad (5)$$

The initial sampling stage ends when at least one phrase from each cluster has been sampled for annotation. In our experiments, the initial sampling usually contains 3 to 7 sentences for a cluster size of 20. Iterative sentence sampling stage aims at achieving maximum CRF model improvement with minimum human annotation effort. Sentences that have been annotated are randomly split into two sets, and each of them is used for training a CRF model. The two models are then used to predict the labels on the rest of the sentences (ranked by sentence importance) in the annotation pool. The first sentence with conflict predictions is selected for human annotation.

2.4 Conditional Random Field Model

The CRF models, which are trained with annotated data splits as mentioned in Section 2.1, employ the following sets of features. One set of 10 features is based solely on token properties: a boolean indicator to account for numbers (e.g., "2019"); a boolean for words starting with uppercase (e.g., "Windows"); a boolean that indicates whether current word contains only uppercase letters (e.g., "CJS"); current token; previous token; following token; part of speech tag for the current token; part of speech tag for the previous token; part of speech tag for the following token; and word vectors (from Word2Vec [14]) for the current token. The other set contains 8 features to account for how the current token relates to the segmenting of the text into phrases: the BIO label of the current token is modeled by three boolean features; the other features are: part-of-speech tag sequence for phrase to which the token belongs (e.g., NN NNS for the tokens of "vertex failures", and empty marker for the token "failed", which does not belong to a phrase); lowest common ancestor for the part-of-speech tags of the phrase (e.g., NP for "vertex failures"); the residual inverse document frequency (RIDF) of the phrase t, computed as

$$log_e \frac{N}{f(t)} - log_e \frac{1}{1 - e^{\frac{-f(t)}{N}}}; \quad (6)$$

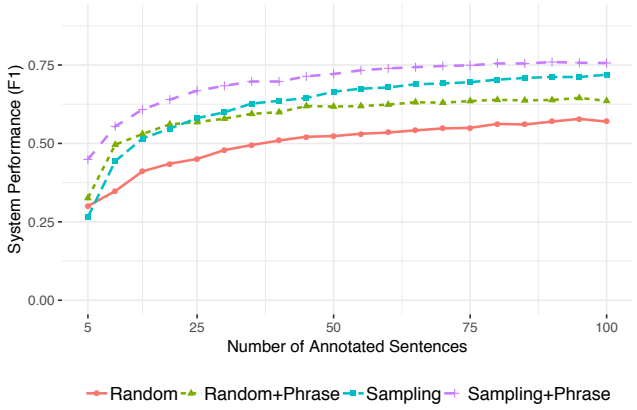

Figure 3: F_1 performance of the proposed system by using different sampling methods and feature groups. X-axis shows the number of annotated sentences in the training set; Y-axis shows the corresponding F_1.

the entropy (ENT) of the current phrase

$$-\sum_{t_i \in \{t_1, \ldots, t_m\}} Pr(t_i)\ log_e(Pr(t_i)); \quad (7)$$

pointwise mutual information (PMI) of the current phrase

$$log\ \frac{Pr(t_1, \ldots, t_m)}{\Pi_{i \in (1,m)} Pr(t_i)}; \quad (8)$$

WPMI of the current phrase; and Word2Vec calculated phrase vector.

3 EXPERIMENTS AND FINDINGS

To evaluate the performance of the system, 50 sentences were randomly selected and annotated, then removed from the collection and employed as test set. The annotators identified in these sentences 603 entity mentions, which are used for evaluation. Figure 3 compares the performance of the full system and several partial systems. To the best of our knowledge, there are no existing systems targeting at the same open-domain low-resource settings. Thus, we employ *Random*, a system that randomly selects sentences from the annotation pool and uses the feature group solely based on tokens, as our baseline. *Random+Phrase* randomly selects sentences for annotation, and uses all available features; *Sampling* selects sentences based on the strategies from Section 2.3 but uses only token-based features in CRF model training. Finally, *Sampling+Phrase* represents the complete system, which includes active sentence sampling and all investigated features.

Performance improves in every setting with more sentences being annotated and added to the training pool. Starting from similar F_1 when only 5 sentences are annotated, the performance of *Sampling* improves faster than *Random*. This shows that the proposed sentence sampling method accelerates the learning rate and therefore, reduces in practice the annotation cost. With the same sentence sampling method, *Sampling+Phrase* starts from a higher F_1

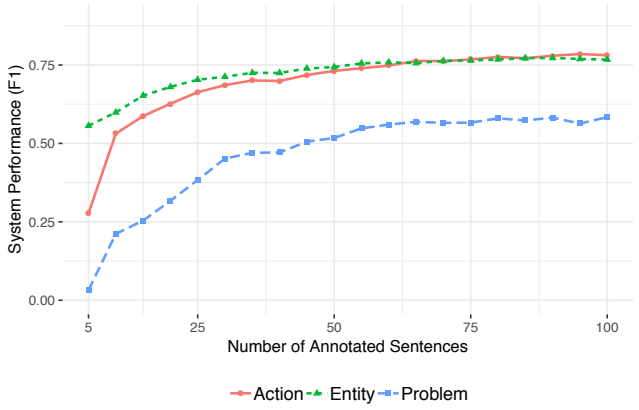

Figure 4: F_1 performance for different entity types.

than *Sampling* and keeps its advantage throughout the learning process. This is likely because phrases recognized and ranked by using the methodology from Section 2.2 are more likely to be valid entity mentions, which reduces the learning cost for mention spans. The curves of *Sampling* and *Random+Phrase* cross with 23 sentences annotated; this shows that the automatically identified phrases play an important role when the number of annotated sentences is very small; as more sentences get annotated, the sampling procedure becomes more important to the accelerate learning. When these techniques are used in conjunction, *Sampling+Phrase* outperforms the baseline *Random* in a statistically-significant manner, and achieves F_1 0.77 with only 100 annotated sentences.

We observed several factors that influenced performance with respect to the custom entity types handled by the system. As shown in Figure 4, the difficulty of recognizing entity mentions varies across types. Recognizing mentions of Problem was found to be harder than recognizing Action or Entity mentions, partially because the lack of expressive lexical clues for this type (e.g., both *imbalanced job* and *starting and shutting down continuously* are categorized as Problem). External knowledge (e.g., domain-specific dictionary, expert provided knowledge) could potentially further improve the performance based on the current system. The second factor is the number of user-defined types. Adding more types increases the risk of label confusion and decreases the overall performance. Last but not least, another finding is that tokens that can be used both in entities and as functional words are difficult to handle when very little annotated data are available. For example, the token "starting" is tagged both as Problem in "starting and shutting down continuously" and as Other in "Starting from 11/1".

4 CONCLUSION AND FUTURE WORK

We proposed an active learning framework for the open-domain low-resource NER, which provides flexibility in terms of entity types to be recognized. The learning process is independent from external general sources (e.g., Wikipedia) or pre-existing collection-specific resources. We evaluated the contributions of different components through ablation experiments, and showed that the full system achieves performance that is usable for real scenarios. For future work, we plan to apply and test the framework in other low-resource domains. In addition, we plan to build an online interface to collect real user feedback for this framework, and enable the creation of publicly-available open domain NER datasets for research.

REFERENCES

[1] David Arthur and Sergei Vassilvitskii. 2007. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 1027–1035.

[2] Jason Chiu and Eric Nichols. 2016. Named Entity Recognition with Bidirectional LSTM-CNNs. *Transactions of the Association of Computational Linguistics* 4, 1 (2016), 357–370.

[3] Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational linguistics* 16, 1 (1990), 22–29.

[4] Cıcero Dos Santos, Victor Guimaraes, RJ Niterói, and Rio de Janeiro. 2015. Boosting Named Entity Recognition with Neural Character Embeddings. In *Proceedings of NEWS 2015 The Fifth Named Entities Workshop*. 25.

[5] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 363–370.

[6] Ralph Grishman and Beth Sundheim. 1996. Message Understanding Conference-6: A Brief History. In *COLING*, Vol. 96. 466–471.

[7] J-D Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. 2003. GENIA corpus-a semantically annotated corpus for bio-textmining. *Bioinformatics* 19, suppl. 1 (2003), i180–i182.

[8] Jin-Dong Kim, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, and Nigel 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*. Association for Computational Linguistics, 70–75.

[9] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural Architectures for Named Entity Recognition. In *Proceedings of NAACL-HLT*. 260–270.

[10] Xuezhe Ma and Eduard Hovy. 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Vol. 1. 1064–1074.

[11] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*. 55–60.

[12] Andrew McCallum, Dayne Freitag, and Fernando CN Pereira. 2000. Maximum Entropy Markov Models for Information Extraction and Segmentation. In *ICML*, Vol. 17. 591–598.

[13] Andrew McCallum and Wei Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, 188–191.

[14] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[15] Matthew Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 1756–1765.

[16] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, Vol. 1. 2227–2237.

[17] Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, 147–155.

[18] Burr Settles. 2004. Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the international joint workshop on natural language processing in biomedicine and its applications*. Association for Computational Linguistics, 104–107.

[19] 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 the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, 142–147.

[20] GuoDong Zhou and Jian Su. 2002. Named entity recognition using an HMM-based chunk tagger. In *proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 473–480.