Named Entity Recognition with Extremely Limited Data

John Foley, Sheikh Muhammad Sarwar, and James Allan
Center for Intelligent Information Retrieval
CICS, University of Massachusetts Amherst
[jfoley,smsarwar,allan]@cs.umass.edu

ABSTRACT

Traditional information retrieval treats named entity recognition as a pre-indexing corpus annotation task, allowing entity tags to be indexed and used during search. Named entity taggers themselves are typically trained on thousands or tens of thousands of examples labeled by humans.

However, there is a long tail of named entities classes, and for these cases, labeled data may be impossible to find or justify financially. We propose exploring named entity recognition as a *search* task, where the named entity class of interest is a query, and entities of that class are the relevant "documents". What should that query look like? Can we even perform NER-style labeling with tens of labels? This study presents an exploration of CRF-based NER models with handcrafted features and of how we might transform them into search queries.

1 INTRODUCTION

Consider a reporter, researcher, or policy maker interested in the Syrian refugee crisis, who has found an article discussing how Malta, as a country, has been impacted by the influx of refugees. Suppose this user now wants to find information that details how other island-nations in the Mediterranean have been affected. One technique that could help satisfy this information need is named entity recognition (NER). If a tagger had marked *countries* and *islands* or (better yet) *Mediterranean islands* and they were indexed by the system, the searcher could construct a query targeting documents that had mentions of refugees and islands (or Mediterranean islands). Why, though, would *islands* have been among the obvious taggers to construct in advance? Would someone bother to build a tagger for *Mediterranean islands*?

Collecting this kind of data may be difficult or impossible for a domain expert. Sequence labeling tasks like this require fine-grained labels and experienced annotators.

Query-log or knowledge-base analysis would allow one to derive a set of entity classes that are of interest, but entity types would have the same long-tail problem as language: there would always be classes missing. Even without approaching the long-tail, it is fairly simple to come up with entity classes that do not quite fit the traditional four-class paradigm (PER, LOC, ORG, other), such as *musical genre*, which is obviously none of the above, or *cigarette brand* which while being related to company or ORG is definitely a distinct entity class. As an experiment, we labeled a number of entities that would have been helpful for TREC QA questions, and found a wide variety of entities, from the common *country* and *city* to the rare: *black panther members*, *airlines*, *sub-atomic particles* and *comets*. Coming up with a comprehensive set of classes a priori seems destined to fail.

Instead, we propose an approach that treats named entity detection as a *retrieval* task, one performed at query time rather than in advance while indexing. We do not propose this as a replacement for NER, but as something to be used for an ephemeral or contextual class of entity, when it does not make sense to label hundreds or thousands of instances to learn a classifier. We will show that it is possible to achieve reasonable effectiveness using this approach, though it does require a substantially larger index. We will further show that this approach can respond rapidly, an important validation since we are moving a document pre-processing step to query time.

We are interested in *transforming* the recognition task into a search task and we refer to the task as Named Entity Search (NES). We do that by considering each word occurrence as a document and the words' features as "terms", allowing us to use traditional search engines in order to score words in response to a query, where the query is the entity class of interest.

In this work, we explore four core research questions:

(1) How do we translate CRF-based NER models with handcrafted features into a retrieval model where tokens are documents? (See §3 and Table 1.) By focusing on models with handcrafted features, our users could potentially edit or provide feedback on the features in our short queries [25].
(2) Can we produce a reasonable ranking over tokens for the user to interact with sentences when we only have a handful of labels to create an NER-classifier? (See Figure 1.)
(3) Can we make our NER models small enough that they can be efficiently executed by a search engine without loss of effectiveness? (See Figure 2.)
(4) What typical handcrafted NER features are most useful for our ad-hoc variant of NER? (Results in Table 5.)

We present experiments and analysis for these research questions, and begin by discussing related work.

2 RELATED WORK

Named entities have a long history of being important and useful for information retrieval tasks. Recently, work has focused on connecting entities present in text to knowledge bases to allow for improved reasoning [8, 31]. Entities have also been at the center of a large amount of research, including the entity recognition

LND4IR '18, July 12, 2018, Ann Arbor, Michigan, USA

and disambiguation challenge [5], the TREC temporal summarization track [1], the TREC knowledge-base population track [20], and the entity ranking tracks at INEX [11] and TREC [3]. Entities, specifically people with emails, were also the highlight of the expert-finding task within the TREC enterprise track [7]. Almost all of these problems and domains depend on entity recognition as a document-enrichment step.

In natural language processing, named entity recognition (NER) and part-of-speech tagging (POS) are considered examples of the sequence labeling problem. One of the common evaluations for this task is the CoNLL-2003 shared task [29], which introduced English and German datasets derived from expert-labeled news articles. Successful NER systems use a large number of features, including gazetteers, case information, word cluster ids, and part-of-speech tags. A 2007 survey by Nadeau and Sekine [23] provides more information about the origins of this task and traditional approaches. A recent work that discusses the challenges of generalization for modern NER approaches [2] may be an additional resource in lieu of a more recent survey.

2.1 Recent Approaches in NER

Most state-of-the art NER approaches are now based on neural networks and deep learning. While using neural networks for NER is not new (one of the CoNLL 2003 submissions used a LSTM network [12, 29]), the availability of more memory and more data has made these approaches the dominant research direction. Recent advances in deep neural networks have led to a reduction in the number of hand-tuned features required for achieving state-of-the-art performance.

However, since we expect our task to have very few labels, and we hope to minimize the complexity of our query models. Our work includes the "distributional similarity features" included in Stanford NLP, which are Brown clusters [4, 16]. In future work, we hope to explore explicitly learning neural representations from unlabeled data, e.g. Lample et al [14].

2.2 Many-classed NER

Some work explores more fine-grained representations of entities, namely a work by Lee et al. which explores a model for identifying named entities as a single step and classification into 147 different categories as a secondary step [15]. More recently, Ling and Weld present FIGER, a system for fine-grained entity recognition of 112 tags, trained using data collected from Wikipedia links used a similar approach: a CRF for segmenting into B (beginning), I (intermediate), and O (outside) tags, and a multiclass classification task on top [18]. Lin et al. assign more than 1000 types to noun-phrases to help classify out-of-knowledge-base entities [17]. In order to be as general as possible, we do not consider filtering to noun-phrases or classified spans in this work, but that is a promising approach to increasing efficiency in future work.

2.3 List Completion and QA

Two existing tasks are similar to NES in terms of their goal: generating a list of entities based on user intent. However, neither explores a limited-data setting.

2.3.1 *Question Answering Systems.*

First, question answering systems often leverage named entity recognition as a method of collecting potential answers [15, 22], for both factoid QA as well as list QA [10, 30]. However, historically, some of the best systems required intensive data resources that do not make sense for our ad-hoc task. The best system from 2007 used coverage from the lists available in Wikipedia, ahead-of-time construction of target NER classes, and sophisticated textual analysis based on logic systems [21]. Another consistently high-ranked system used a tiered response system that first checked a database built from "info boxes," then Wikipedia lists and tables, then a set of data from lexicons, and even web search hit counts [13]. Often, systems targeting factoid queries merely repeatedly executed their factoid method. Because of our reluctance to limit potential answers to those in external resources and our desire to incorporate interactivity we do not compare directly to any TREC-QA systems in this work. However, we do make use of the list questions in the TREC 2005 and 2006 Question Answering track [10, 30].

2.3.2 *List Completion.*

List completion exists in various forms, with some work inspired by the no-longer-running Google Sets [9]. However, this work presumes that data mining methods will be effective: that entities of interest occur in HTML or Wikipedia lists with some regularity. A similar task was explored at the INEX list completion task [11], which was based on selecting the best answers from a fixed Wikipedia corpus. In the language of existing List Completion approaches, we could call our task *list completion over unstructured text*, but we are not aware of any similar work.

2.4 Active Learning

Active Learning is a broad field that arose from the desire to have machine learning algorithms select a subset of the training data to learn the best classification function, but the tools and techniques in this field are applicable to interactive labeling tasks as well. Settles presents a comprehensive survey of this subfield [26].

In the realm of NER, active learning has specifically been used to minimize human annotation efforts. Shen et al. show that only 20% of labels are truly required to achieve strong performance with a good selection function [27]. Rather than minimizing human interactions to achieve a final high-accuracy classifier, our named entity search task focuses on *extremely* limited interaction and high-precision effectiveness.

3 METHODS

The core of our interactive Named Entity Search is the ability to leverage user-feedback in order to quickly and interactively construct useful named entity taggers. We briefly discuss conditional random field, as the dominant approach to non-neural NER as it is more suitable for our fewer label, interactive setting. And then we talk about the user interaction model.

3.1 CRF as a Retrieval Model

The retrieval model specifies how retrievable items (tokens represented by features) are ranked in response to the query (weights of features). Based on the success of CRFs, we start with its calculation of the probability that a token t has a label y_t. We use the notation of Sutton and McCallum [28]:

$$p(y_t|x) = \frac{1}{Z} \exp\left(\sum_{k=1}^{K} w_k f_k(y_t, y_{t-1}, x_t)\right)$$

To make this useful for search, we first apply some standard IR transformations to this model. We intend to rank tokens consistent with this probability rather than calculating it directly.

Because we are interested in ranking, rather than exact probabilities, we can perform a number of typical rank-safe transformations: removing the $\frac{1}{Z}$ normalization, and taking the log of each side. We cannot remove the dependency of y_t on y_{t-1} with pure mathematics, so we turn to experimentation to determine whether we can simplify this expression further and to quantify the loss of accuracy.

We also validate empirically that with linear CRF models, especially with fewer labels, the transition probabilities are of little benefit. Table 1 shows the results of downgrading a full CRF to doing a single class at a time and then removing transition probabilities on a standard NER dataset. By ignoring the transition ans sequence passes of learning and training classes independently, we have achieved a linear query model and measured the difference between that and the more typical CRF.

$$p(Q|x) = \sum_{k=1}^{K} Q_k f_k(x) = \vec{Q} \cdot \vec{f}(x)$$

This means that we treat each token in a collection as a document, and it is represented by a bag of traditional NER features (as extracted by Stanford NLP [19]). This means that we can effectively execute our NER models with an IR system that supports this sort of vector-space model.

Table 1: Classification Results on our NER Dataset. Measures are the F_1 of token classification experiments to explore the effect of our new assumptions.

System	PER	LOC	ORG	MISC	Micro	Macro
Full CRF	86.0	85.6	91.1	61.9	86.5	81.1
Query independence	85.2	87.3	90.5	59.2	86.3	80.6
Token independence	84.5	86.4	89.8	63.0	85.9	80.9

3.2 User Interaction

In early feedback, we discovered that returning tokens or spans to the user was not sufficient to understand whether the entities being discovered were correct, so we built our system around a sentence-feedback model, where users are presented a sentence at a time (which contains the highest ranked tokens) and users label the entire sentence before the system re-ranks and presents them with the next best sentence. In this way, a user works to solve their problem (collecting a list of entities) while generating a small amount of training data suitable for NER software [24].

4 DATASETS

We explore questions related to NES using a well-labeled NER dataset on which a classification task makes sense. Our fundamental research questions regarding whether NES is practical and effective are done using a novel QA-List dataset.

4.1 NER

We leverage the English dataset from the CoNLL-2003 shared task. Table 2 provides some statistics about that collection. It consists of 946 documents for training, 231 documents for validation, and 231 documents for testing. Our approach does not need to train parameters, so we did not use the validation set. In order to compare to previously published results, we choose to ignore that portion of the data rather than mix it into the training or test sets.

4.2 QA-List

In order to explore research questions that surround sparse entity types, we have adapted a dataset from the TREC Question Answering Track. We select data from the 2005 and 2006 challenges because they operate on the same dataset (AQUAINT) and the full set of annotator judgments are available for all surface forms reported by all participants in the challenge. Some statistics about AQUAINT in comparison to the smaller NER dataset are presented in Table 2.

Table 2: Comparison and Summary of Datasets.∗
We present a comparison of sizes here to emphasize that the naïve storage cost of our new task (storing per-token features as documents) expands the size of the collection, motivating our look into efficiency concerns.

Collection	NER	QA-List
Source	CoNLL'03	AQUAINT
#docs	1408	1,088,791
#words	203,621	488,789,688
#uniq words	20,386	1,113,614
Inv & Fwd Index	0.002G	2.8G
NER Features	0.276G	495G
NER Features Index	0.095G	158G
CRFSuite Tagging Time	2 seconds	2 hours

∗ G=2^{30} bytes, M=2^{20} bytes, as calculated by du -csh

Since we are explicitly curating this data to use less-common entity types, we preprocess it to skip the 25 queries that are explicitly seeking lists of countries and the 4 queries that are seeking lists of cities. We also drop queries that have 3 or fewer answers because we want to show learning curves for this task. After preprocessing, we have 66 queries from 2005 and 60 queries from 2006. Some examples can be found in Table 3.

We limit our processing to the provided ranking created by the track organizers. While this limits our recall slightly (there are positive judgments outside these rankings), this prevents our technique from pulling up large numbers of unjudged tokens, and makes our experiments more computationally efficient. In the future, we hope to fully adapt this dataset and collect the additional judgments.

4.3 Unique Average Precision (uAP)

For our person query (PER), we do not want to claim high effectiveness if the only person it tags in the top ranks is "Barack Obama", even if our system is technically correct about the class of all those repeated instances. We define *unique average precision (uAP)* to be standard average precision, but only after *removing* all subsequent mentions of any entity. Unlike evaluations of retrieving novel

Table 3: Examples of QA-List Questions

Title	QID	Items
Ben & Jerry's unusual flavors	172.7	Chubby Hubby, Phish Food, Cherry Garcia, . . .
Mammals cloned	197.4	rabbit, cow, mouse, pig, . . .
Vaccines for Avian Flu	166.6	Relenza, Tamiflu, amantadine, RWJ-270201, . . .
Boxers who defeated Foreman	77.7	Muhammad Ali, Shannon Briggs, Evander Holyfield, Tommy Morrison, . . .
Names of Meteorites	84.7	Lucky 13, ZAG, ALH84001, Leonid, SNC, . . .

material [6], uAP *ignores* subsequent occurrences of both relevant and non-relevant mentions rather than treating them all as non-relevant. Although F_1 is typically used for NER, it is unsuitable for NES because it is a set-based measure and we care about the actual ranking. (We do use F_1 when comparing NER results.)

This notion of unique-AP is our way of encoding the "distinctness" that was done by human assessors in the TREC-QA list track: they marked arbitrary instances as the distinct instance, and counted it as a recall point for AP if and only if it was both correct and distinct. Our measure does the same, except at the token level, and automatically.

5 INTERACTIVE NES FOR CHALLENGING ENTITY CLASSES

In this section, we use our derived QA-List dataset (Section 4) in order to explore learning curves on a more realistic task.

While the active learning literature [26] has thoroughly explored methods for classifiers to select instances for labeling, we have a slightly different twist on this task. Instead of collecting labels through active learning while running the full classifier, we want to see if our approximations still allow for a strong learning curve.

5.1 Is Interaction Helpful?

We explore three interaction models for the NES problem of labeling the entities of interest. The first model is our own, an interactive model that labels the top-ranking sentence, and immediately feeds it back to create a new query and thus a new ranking. The second model is one that labels sentences in the top-ranked documents (for the NER dataset, we use collection order in lieu of rank). The final model is that of a more traditional NER labeling: an assessor chooses random sentences. In the QA-List dataset, our random baseline is built on document pools, since in expectation a truly random selection of sentences from AQUAINT would lead to no positive labels over hundreds or thousands of iterations. Our final baseline "Unsure" is derived from active learning: it chooses an instance based on how unsure the classifier is about it, with the intuition that these instances will be most informative to the learning algorithm.

Interactivity curves for our QA-List dataset are presented in Figure 1a. It shows the change in effectiveness (uAP) as the query changes via interaction. Selecting random documents is useless and is easily out-performed by a strategy that labels sentences from high-ranking relevant documents.

Table 4: Query Efficiency vs. Model Size This table shows the trade-off between the interactive IR-approach to model evaluation and the offline-NLP approach over the full AQUAINT corpus.

Algorithm	$\lvert Q\rvert$	Median Time(s)	Best Unit
Lucene	1	0.135	< 1 sec
	5	0.319	< 1 sec
	10	0.499	< 1 sec
	50	5.45	< 10 sec
	100	22.4	< 1 min
	250	219.0	3-4 min
	500	566.0	9-10 min
	1000	1700.0	30 min
	5000	12200.0	3-4 hours
CRFSuite	Full	7200.0	2 hours

Similar curves over the NER dataset are presented in Figure 1b, but shows nearly the opposite effect. This corpus is unusual in that it has an extremely high concentration of named entities, so that a random sentence is very likely to contain one. As a result, randomly selected text creates a recall-enhancing effect.

We believe that the QA-List dataset captures our target problem more clearly: we expect users to have information needs regarding sparse, "tail" entities. However, these results point up the need for retrieval models that can find diverse entities to boost recall.

5.2 Is this efficient and effective?

In order to demonstrate our model's possibilities, we must be proposing something feasible. This is important because we can evaluate our smaller models with much less CPU-time over a much larger dataset than traditional CRF systems (see Table 4), which is important for any sort of interactivity on real, modern datasets. From an opportunity cost perspective, the inverted list algorithm beats direct scoring for a corpus of this size as long as it executes in less than an hour, but those times are still unreasonable for users.

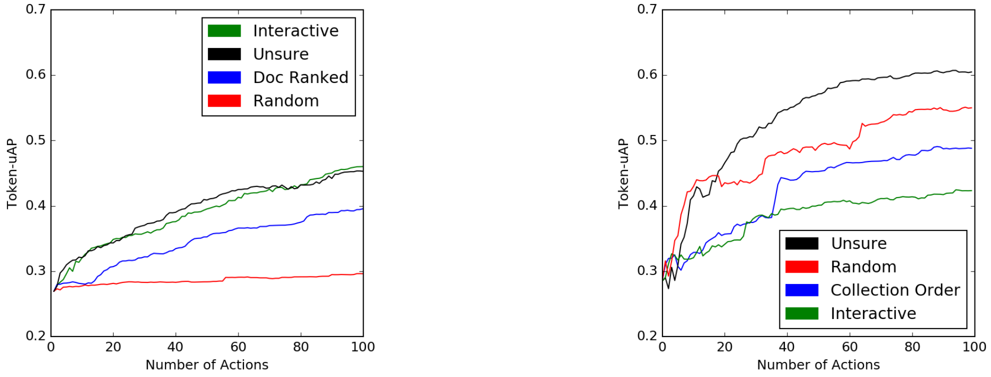
In our challenging QA-List task, we have been able to show that the learning curves of down-sampled models are insignificantly different than the full models. When we consider this in conjunction with our efficiency results, this means that we could execute our interactive system on the full AQUAINT dataset in just a few CPU-minutes rather than hours. See Figure 2 for representative curves. If we limit ourselves to a few hundred of the most important features, we get nearly all of the benefits without using hours of CPU-time.

5.3 What features are most effective?

We studied the top-10 features selected by each model (weighted by their occurrence across models and by those models' uAP score), and found the top 5 most important features by domain. These results are presented in Table 5.

The most useful features for the NER domain were word clusters [4, 16] trained on the RCV1 corpus provided pre-trained by Stanford NLP [19]. We briefly explored training[1] our own word clusters on the AQUAINT corpus but we observed no significant difference in learning curves. We hope to explore more distributional similarity features in the future.

[1] https://github.com/percyliang/brown-cluster



(a) QA-List dataset.

(b) NER dataset.

Figure 1: Interactive effectiveness on our two datasets.
The NER dataset (right) is deeply labeled, but is less informative for our desired problem as a result of the entity classes being too frequent and too simple. On our new QA-List dataset, however, we can see that the unsure and interactive baselines are equally competitive, which validates our new, interactive approach to NER for more difficult datasets and sparse labels.

Table 5: The top features in our analysis.

Rank	NER	QA-List
1	Brown Cluster Curr	Words to the Right
2	POS Tag	Words to the Left
3	Char N-grams	Char N-grams
4	Brown Cluster Next	Word Shape Next
5	Words to the Right	Word Shape Prev,Curr

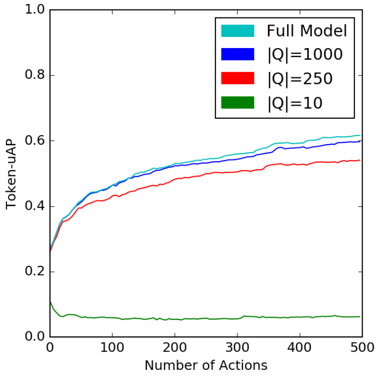


Figure 2: Effectiveness on QA-List with limited features for efficiency. On this dataset, the learning curve for models limited to 1000 features is nearly indistinguishable from the curve with full models. Limiting model size does not result in a significant performance drop within a few hundred interactions.

6 CONCLUSION

In this study, we explore the possibility of viewing a traditionally offline pre-processing and token classification task as an exploratory search task to deal with extremely limited data. In so doing, we introduce a novel problem that could help analysts, reporters, and other expert users understand and solve their information needs.

We explore effectiveness, modeling, and efficiency while focusing on handcrafted features rather than neural models so that we can explore human-edited queries in the future. We conclude that our approach will be feasible for expert use. In the future, we hope to fuse more research from the active learning, NLP, and IR fields to develop new approaches to this task.

ACKNOWLEDGEMENTS

This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF grant #IIS-1617408. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsors.

The authors would additionally like to thank Jiepu Jiang, who helped us to extract the NLP features at the core of these experiments.

REFERENCES

[1] J. Aslam, M. Ekstrand-Abueg, V. Pavlu, F. Diaz, and T. Sakai. Trec 2013 temporal summarization. In *TREC'13*, 2013.

[2] I. Augenstein, L. Derczynski, and K. Bontcheva. Generalisation in named entity recognition: A quantitative analysis. *arXiv preprint arXiv:1701.02877*, 2017.

[3] K. Balog, P. Serdyukov, and A. P. d. Vries. Overview of the TREC 2010 entity track. Technical report, DTIC Document, 2010.

[4] P. F. Brown, P. V. Desouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479, 1992.

[5] D. Carmel, M.-W. Chang, E. Gabrilovich, B.-J. P. Hsu, and K. Wang. ERD'14: entity recognition and disambiguation challenge. In *ACM SIGIR Forum*, volume 48. ACM, 2014.

[6] C. L. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon. Novelty and diversity in information retrieval evaluation. In *SIGIR 2008*, pages 659–666. ACM, 2008.

[7] N. Craswell, A. P. de Vries, and I. Soboroff. Overview of the TREC 2005 Enterprise Track. In *TREC*, volume 5, pages 199–205, 2005.

[8] J. Dalton, L. Dietz, and J. Allan. Entity query feature expansion using knowledge base links. In *SIGIR'14*, pages 365–374. ACM, 2014.

[9] B. Dalvi, J. Callan, and W. W. Cohen. Entity list completion using set expansion techniques. 2011.

[10] H. T. Dang, J. Lin, and D. Kelly. Overview of the TREC 2006 Question Answering Track. In *TREC 2006*. NIST, 2000.

[11] G. Demartini, T. Iofciu, and A. P. De Vries. Overview of the INEX 2009 entity ranking track. In *Focused Retrieval and Evaluation*, pages 254–264. Springer, 2010.

[12] J. Hammerton. Named entity recognition with long short-term memory. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 172–175. Association for Computational Linguistics, 2003.

[13] A. Hickl, K. Roberts, B. Rink, J. Bensley, T. Jungen, Y. Shi, and J. Williams. Question Answering with LCC's CHAUCER-2 at TREC 2007. In *TREC*, 2007.

[14] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*, 2016.

[15] C. Lee, Y.-G. Hwang, H.-J. Oh, S. Lim, J. Heo, C.-H. Lee, H.-J. Kim, J.-H. Wang, and M.-G. Jang. Fine-grained named entity recognition using conditional random fields for question answering. In *AIRS'06*.

[16] P. Liang. *Semi-supervised learning for natural language*. PhD thesis, Massachusetts Institute of Technology, 2005.

[17] T. Lin, O. Etzioni, et al. No noun phrase left behind: detecting and typing unlinkable entities. In *EMNLP'12*. ACL, 2012.

[18] X. Ling and D. S. Weld. Fine-grained entity recognition. In *AAAI*, 2012.

[19] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The Stanford CoreNLP natural language processing toolkit. In *ACL'14 Demo*, pages 55–60, 2014.

[20] P. McNamee and H. T. Dang. Overview of the TAC 2009 knowledge base population track. In *TAC'09*, volume 17, pages 111–113, 2009.

[21] D. Moldovan, C. Clark, and M. Bowden. Lymba's poweranswer 4 in trec 2007. 2007.

[22] D. Mollá, M. Van Zaanen, D. Smith, et al. Named entity recognition for question answering. 2006.

[23] D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[24] N. Okazaki. Crfsuite: a fast implementation of conditional random fields (crfs). http://www.chokkan.org/software/crfsuite/manual.html, 2007.

[25] S. Sarwar, J. Foley, and J. Allan. Term relevance feedback for contextual named entity retrieval. In *CHIIR*, pages 301–304, 2018.

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

[27] D. Shen, J. Zhang, J. Su, G. Zhou, and C.-L. Tan. Multi-criteria-based active learning for named entity recognition. In *ACL'04*, 2004.

[28] C. Sutton and A. McCallum. An introduction to conditional random fields. *arXiv preprint arXiv:1011.4088*, 2010.

[29] E. F. Tjong Kim Sang and F. De Meulder. 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*. ACL, 2003.

[30] E. M. Voorhees and H. T. Dang. Overview of the TREC 2005 Question Answering Track. In *TREC 2005*. NIST, 1999.

[31] C. Xiong and J. Callan. EsdRank: Connecting Query and Documents through External Semi-Structured Data. In *CIKM'15*.