SearchIE: A Retrieval Approach for Information Extraction

Sheikh Muhammad Sarwar and James Allan
Center for Intelligent Information Retrieval
College of Information and Computer Sciences
University of Massachusetts Amherst
{smsarwar,allan}@cs.umass.edu

ABSTRACT

We address the problem of entity extraction with a very few examples and address it with an information retrieval approach. Existing extraction approaches consider millions of features extracted from a large number of training data cases. Typically, these data cases are generated by a distant supervision approach with entities in a knowledge base. After that a model is learned and entities are extracted. However, with extremely limited data a ranked list of relevant entities can be helpful to obtain user feedback to get more training data. As Information Retrieval (IR) is a natural choice for ranked list generation, we explore its effectiveness in such a limited data case. To this end, we propose SearchIE, a hybrid of IR and NLP approach that indexes documents represented using handcrafted NLP features. At query time SearchIE samples terms from a Logistic Regression model trained with extremely limited data. We explore SearchIE's potential by showing that it supersedes state-of-the-art NLP models to find civilians killed by US police officers with only a single civilian name as example.

KEYWORDS

Information retrieval; information extraction; search index

ACM Reference Format:
Sheikh Muhammad Sarwar and James Allan. 2019. SearchIE: A Retrieval Approach for Information Extraction. In *2019 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR '19), October 2–5, 2019, Santa Clara, CA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3341981.3344248

1 INTRODUCTION

Consider a user searching for *a list of civilians killed by Police*, who issues that query to a search engine. She lands on a web page where she finds the sentence: "*On March 1, 2000, just a few days after a jury acquitted the four police officers who killed* ***Amadou Diallo****, an undercover cop shot and killed 23-year-old* ***Malcolm Ferguson*** *at his Bronx home.*"[1].

Now, the user has one sentence with a couple of positive instances and a query to express her information need. She wants to build a model that would be able to extract more entities like *Amadou Diallo* and *Malcom Ferguson.* Entities such as these do not have a Wikipedia page as they are not popular entities. Hence, we cannot adopt entity retrieval based approaches that depend upon searching through knowledge base or articles on entities organized by entity categories [12]. Entity co-occurrence based models would suffer from lower precision if the co-occurring entity is too generic, such as *Bronx* that occurs in numerous contexts [2].

Another way to approach this problem is to construct a weakly supervised training dataset and estimate a statistical NLP model (e.g., feature-rich logistic regression, CNN, CRF) [6]. A weakly supervised dataset is usually constructed by automatically labeling sentences with relevant entities from a knowledge base or a historical list. In the case of our example, the lack of a manually curated historical database of police killing would make this process infeasible.

Active Learning (AL) based approaches could also be used in this setting to gather informative training data [11]. But a statistical model estimated using extremely limited data would be ineffective in determining informative examples for annotation. Given the circumstances, it is rather important to address the seed selection problem for AL to feed the learner with more positive examples [3]. We propose to construct a retrieval model using extremely limited data and rank sentences based on their likelihood of containing a police killing event and the entities involved with it. We score person entities from the top-k sentences in the ranked list to construct a ranked list of candidate entities. We assume that a user would be able to find more seeds by by inspecting the ranked list of entities. We expect that this retrieval based seed selection approach would help to bootstrap a classifier to effectively perform AL. Overall, by incorporating ideas from NLP and IR this study answers the following research questions:

- Given an extremely limited number of examples as input, how do extraction models perform compared to retrieval models in finding and ranking more entities similar to the examples?
- To take a retrieval approach, how can we effectively use the input examples to construct a search query? How effective is it to use only the surface form of the examples as a query? Can we use the surface forms along with a user provided keyword query such as "find me civilians killed by police"?
- Can we use NLP features computed from the sentences containing the examples as search query terms? How can we select terms from features and how can we construct a retrieval index for such query terms?

[1] https://www.huffingtonpost.com/2014/07/18/killed-by-the-nypd-black-men_n_5600045.html

ICTIR '19, October 2–5, 2019, Santa Clara, CA, USA
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6881-0/19/10…$15.00
https://doi.org/10.1145/3341981.3344248

2 PROBLEM DEFINITION AND APPROACH

We assume a user wants to extract list L of entities from a large sentence corpus S. The user provides a query q and a few exemplars $E \subset L$ annotated in a set of sentences $S_E \subset S$. Our task is to provide a framework using which the user would be able to efficiently and effectively retrieve all other elements of L – *i.e.,* $L - E$. In the extreme case, we will have q and one exemplar annotated in a sentence.

The output of such a system is measured by the number of unique relevant entities retrieved at the top ranks. The reason behind constructing a high precision system is to enable and support user feedback. With user feedback on retrieved entities it is possible to get more training data and build robust model that do not overfit [7] [4]. This work only considers the initial retrieval step and leaves approaches for interaction to future work.

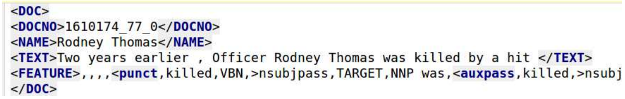
2.1 Proposed Retrieval Approach

In this section, we describe **SearchIE**, our retrieval approach for Information Extraction (IE) with extremely limited data. A similar approach was explored by Foley et al. [5] but it was focused on named entity recognition and did not index long-range features such as different length paths in a dependency parse tree of a sentence. Sarwar et al. [10] approached a similar problem with term relevance feedback from users which is costly to obtain in practice. We require no feedback from the users in the pre-retrieval stage and approach a contemporary extraction task. In the next subsections we describe the sentence retrieval and indexing as well as entity scoring approach.

2.1.1 Sentence Indexing. We propose to index sentences by considering extracted NLP features as terms. Even though complex NLP features appear as a sequence of unigram, bigram, POS tag or Named Entity tags, we consider each part of the sequence as a term and index a sentence against them. For example, if a sentence contains two features: “family, NN, TARGET, NNP, shot, VBN”, and “PERSON, speaks, to”, the sentence is treated as a bag of terms, B = {family, NN, TARGET, NNP, shot, VBN, PERSON, speaks, to} and the sentence is indexed against these terms. The sequence of these terms is preserved using a positional index that stores the positions of the terms in a document along with the terms themselves. A sample TREC style document with terms as features is shown in Figure 1.

The indexing approach is limited to entity types. This study assumes that we are searching for PERSON entities. At the time of indexing a sentence, all the person names in that sentence are replaced with the token PERSON. Finally, each PERSON token is replaced with a TARGET token in turn to create a mention. As a result, we have m mentions of a sentence if there are m person names in that sentence. For each mention in a sentence we extract features and by concatenating all the features from all the mentions in a sentence we create a large “document” from the sentence. We index that document against the DOCNO, and store the person names against that DOCNO.

2.1.2 Sentence Retrieval. Given surface forms of k example entities $E = \{e_1, e_2, \ldots, e_k\}$, we find the set of sentences $X = \{x_{e_1}, x_{e_2}, \ldots, x_{e_k}\}$, where these surface forms appear. A mention, $M_{x^j_{e_i}}$ of entity e_i is constructed by taking a single sentence $x^j_{e_i} \in x_{e_i}$ and replacing the entity surface form e_i in that sentence with the token “TARGET”. Now, mention $M_{x^j_{e_i}}$ becomes a positive training instance from which we can extract features. We extract the features mentioned in a study of identifying victims of police killing done by Keith et al. [6]. As we use their publicly available dataset, we compute the same features at indexing time and index sentences against those features.



```
<DOC>
<DOCNO>1610174_77_0</DOCNO>
<NAME>Rodney Thomas</NAME>
<TEXT>Two years earlier , Officer Rodney Thomas was killed by a hit </TEXT>
<FEATURE>,,,,<punct,killed,VBN,>nsubjpass,TARGET,NNP was,<auxpass,killed,>nsubj
</DOC>
```

Figure 1: A TREC document created from a sentence. In this document, DOCNO is the sentence ID, NAME field contains a person name, TEXT field contains the original sentence, and FEATURE field contains the features extracted from the sentence using feature templates shown in 2.

Given the sentence set X we form the training dataset $D_{TR} = \bigcup_{i=1}^{k} \bigcup_{j=1}^{|x_{e_i}|} M_{x^j_{e_i}}$ and use the feature function $f : M_{x^j_{e_i}} \in D_{TR} \rightarrow F$ to generate features from a mention. Then we label all of these mentions as positive with probability Q. The negative instances of our training set is also formed by considering all these mentions as negatives with probability $1 - Q$. We take this specific approach because our training data is weakly supervised *i.e.* an entity can appear in different contexts in different sentences. Then we learn a logistic regression model on D_{TR}. We use the following objective function that takes into account the weights of the samples:

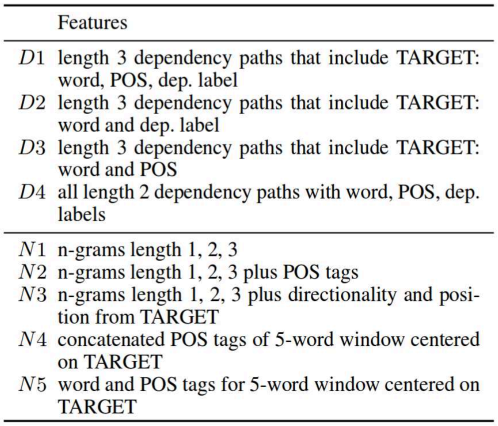
$$L(\mathbf{w}) = \sum_{j}^{m} \log(1 + e^{-y_j \mathbf{w}^T \mathbf{x}_j Q^{[y_j=1]}(1-Q)^{[y_j=-1]}}) + \lambda \mathbf{w}^2$$

For binary classification, a trained logistic regression model is a vector of weights. We only select a subset of features ordered by their weights and use those features as query to our retrieval system. However, we again create a term based representation of a feature as discussed in 2.1.1 that turns a feature into a bag-of-words. However, sequence of these words are important as some of the features are generated by traversing a dependency tree. In this case, we take the advantage of a widely studied proximity search approach that takes the number of words that can appear between the bag of words in a query as input [9].

2.1.3 Entity Scoring. For retrieving the entity list we first retrieve the top n sentences using our proposed IR model. Then we simply count the number of occurrences of each of the names in those sentences and rank those names by their frequency. It is easy for us to find those names as the target entities in our dataset are persons and NER taggers are quite accurate in annotating them. However, for arbitrary entity types this approach cannot currently be applied as entity type detection from free text is very challenging.

3 EXPERIMENTAL SETUP

In this section we discuss the dataset, our example based query sampling process, and baselines.



	Features
$D1$	length 3 dependency paths that include TARGET: word, POS, dep. label
$D2$	length 3 dependency paths that include TARGET: word and dep. label
$D3$	length 3 dependency paths that include TARGET: word and POS
$D4$	all length 2 dependency paths with word, POS, dep. labels
$N1$	n-grams length 1, 2, 3
$N2$	n-grams length 1, 2, 3 plus POS tags
$N3$	n-grams length 1, 2, 3 plus directionality and position from TARGET
$N4$	concatenated POS tags of 5-word window centered on TARGET
$N5$	word and POS tags for 5-word window centered on TARGET

Figure 2: Feature Templates [6]

3.1 Dataset

We evaluated our approach on cross-document entity-event extraction for police fatalities dataset created by Keith et al. [6]. The training examples of this dataset are Fatal Encounter (FE) knowledge base (human curated) entities collected from Jan, 2000 to Aug, 2016. The goal is to find the names of civilians killed by police in the period (Sep, 2016 - Dec, 2016) from Google News data. 258 entities from FE knowledge base were found in Google news data in that period of time.

Mentions of training examples were found in Google News data (Jan, 2016 - Aug, 2016) and sentences with positive mentions were extracted. Sentences with negative mentions contained person entities that were not available in the FE knowledge base. Even though this approach does not take advantage of all the examples available in the history, it was shown to be sufficient for model training [6]. As a result, the historical database contained 17,219 civilians and the training example set could only cover 916 of them. A full description of the dataset can be obtained from the work of Keith et al. [6]. The test example set covered 258 entities and their mentions are found from the news corpus of September, 2016 to December, 2016. Sentences that did not contain mentions from the FE database became the negative training data for both train and test splits.

To take the SearchIE approach, we constructed a corpus of 164,871 sentences by unifying all the training and test sentences. We indexed those sentences using the Indri Search Framework. We index both the original sentence and the feature based representation of the sentence. In fact a sentence becomes a large "document" of features and we index sentences against those features (see Section 2.1.1 for details on feature index construction). Feature extraction templates are listed in Figure 2, taken from Keith el al. [6].

The index contained approximately 146 million terms among which there were only 87 thousand unique terms. We also constructed a text-only index containing 5 million terms with 76 thousand unique terms. The reason behind constructing a text-only index is to compare the performance of corresponding feature based index in terms of extraction performance.

3.2 Query Construction

Our queries are examples – names of civilians in the context of this dataset. We randomly sample 30 names from a set of all the civilian names in the training (916) and test (258) data. Then we create 50 k-example queries by random selection from $\binom{30}{k}$ possibilities. As a result, we have 50 queries for number of examples ranging from 1 to 30. Note that all of the 50 queries for 30-examples queries are the same. The queries and other data used in our experiments have been released publicly.[2]

At the time of evaluation, for SearchIE and all other baselines, no credit was given to a system for retrieving entities belonging to the set of examples since the examples are already known.

3.3 Baselines

We experiment and compare the effectiveness of SearchIE with both ad-hoc IR (Information Retrieval) and IE (Information Extraction) baselines. We considered Query Likelihood (QL) [8] and Relevance Model 3 (RM3) [1] as IR baselines and we used the model proposed by Keith et al. [6] as our IE baseline. For convenience, we refer to this model as **Weak-LR**: a logistic regression model that is trained on weakly supervised data. The performance of Weak-LR is driven by a soft labeling approach, which assumes a mention sentence to be positive with some confidence. Even though Weak-LR is the state-of-the-art for this dataset, it was not designed for and has not previously been tested in the limited examples scenario.

Our baseline models take different types of inputs based on their solution approach. IR models take user-specified keywords concatenated with examples as query. We used three keywords for the user-specified query: *civilians*, *police*, *killed*. Weak-LR and SearchIE takes only examples as input. The output of SearchIE and other IR approaches is a ranked list of sentences, from which a ranked list of entities is computed using the approach of Section 2.1.3. Weak-LR outputs probabilities for all the mentions generated from a sentence and we perform mention level aggregation to generate a score for that sentence. Given m mentions generated from a sentence, the probability for each of those mentions is computed, and the maximum of those probabilities is selected as the score for that sentence. Finally, sentences are ordered based on scores and entity ranked list is constructed using the same frequency based aggregation approach we used for SearchIE and all other baselines to ensure fair comparison.

3.4 Experimental Result

3.4.1 *Feature Effectiveness.*

We ranked the features based on their weights estimated from our Logistic Regression model. Some of the highest ranked features resulted from training with 30 examples are: (TARGET, TARGET O, police, TARGET NN, shot, TARGET NNP, police NN, officers NNS, killed VBN). Some of the lowest ranked features from the same model are: (PERSON NN Talks NNS TO, county NNP courthouse NN, supporters NNS, of cumberland county, supporters 18 on 17, talks to supporters, PERSON talks to, vigil NN case following VBG, steps NNS det the DT). The highest ranked features are more general – recall oriented. The lowest ranked features, which we reject at the time of forming the search

[2] https://github.com/sarwar187/SearchIE

query, are very specific and and comprise long sequence of nodes in dependency path trees. Though they might be useful for making decision about a mention they are not useful for ranking.

3.4.2 Number of Examples. Figure 3 shows the effect of adding more examples with SearchIE and other baselines. SearchIE supersedes the baselines both for very limited number of examples and as the number of examples increase. Please note that we only used 200 highest weighted features regardless of the number of examples to generate this figure. The SearchIE approach has top performance and it generally becomes better as more exampls are provided. The Weak-LR approach is surprisingly unstable, varying substantially with different numbers of examples. We have shown 95% confidence interval for the performance metrics, illustrating that Weak-LR is has wider intervals in general, also supporting the hypothesis that it is more sensitive to the specific set of examples selected.

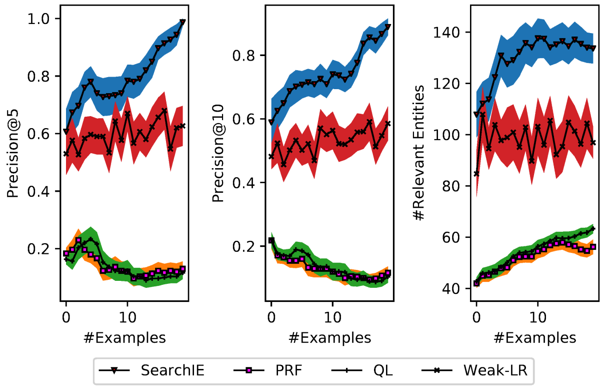


Figure 3: Effect of including more examples

3.4.3 Number of Features in Query. We perform experiments to find the optimal number of features in a query. We found that SearchIE does not achieve any gain in terms of precision@5, precision@10, precision@20, precision@30 and the number of relevant entities in top-1000 sentences after adding 200 features to the query. The results are shown in Figure 4. We used 50 randomly sampled 20-example queries to generate the figure. It is interesting that even with a very few number of selected features SearchIE provides good precision at different ranks.

3.4.4 Reasons for Gain Over Weak-LR. One advantage of indexing multiple mentions against the same sentence is that sentences with more mentions will get more importance. That is, a sentences with more persons mentioned will naturally get boosted by the indexing approach because of the repeated appearance of terms for multiple mentions. It has the equivalent effect of weighting the training data more if there are multiple PERSON mentions in a sentence. Furthermore, feature selection is an important component of SearchIE. Weak-LR computes a huge and sparse feature matrix and the hashes the features to obtain a dense matrix. In this current setting, it is not possible to perform feature selection in Weak-LR.

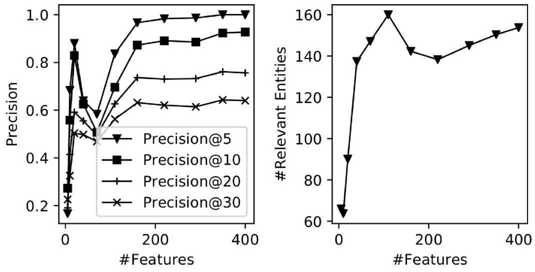


Figure 4: Effect of including more features in query

4 CONCLUSION

We proposed SearchIE as an IR approach to information extraction. SearchIE combines the benefits of inverted index based search and NLP approaches for feature selection. We illustrated SearchIE's potential by showing that it outperforms state-of-the-art NLP model for finding civilians killed by US police. One interesting property of SearchIE is its ability to deal with large feature space. SearchIE stores and indexes all the features and computes important features only from the sentences where examples appear. In contrast, an NLP approach would map all the corpus sentences in the same sparse feature space and learn feature weights from that space based on training sentences.

ACKNOWLEDGEMENT

We thank Brendan O'Connor for very useful discussions and access to the police fatality corpus. 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.

REFERENCES

[1] Nasreen Abdul-Jaleel, James Allan, W Bruce Croft, Fernando Diaz, Leah Larkey, Xiaoyan Li, Mark D Smucker, and Courtney Wade. 2004. UMass at TREC 2004: Novelty and HARD. *Computer Science Department Faculty Publication Series* (2004), 189.

[2] M. Bron, K. Balog, and M. de Rijke. 2010. Ranking related entities. In *CIKM '10.*

[3] D. Dligach and M. Palmer. 2011. Good seed makes a good crop: accelerating active learning using language modeling. In *ACL '11.*

[4] A. Esuli, D. Marcheggiani, and F. Sebastiani. 2010. Sentence-Based Active Learning Strategies for Information Extraction. In *IIR' 10.*

[5] John Foley, Sheikh Muhammad Sarwar, and James Allan. 2018. Named Entity Recognition with Extremely Limited Data. In *1st International Workshop on Learning from Limited or Noisy Data for Information Retrieval.*

[6] K. Keith, A. Handler, M. Pinkham, C. Magliozzi, J. McDuffie, and B. O'Connor. 2017. Identifying civilians killed by police with distantly supervised entity-event extraction. In *EMNLP '17.* 1547–1557.

[7] H. Oiwa, Y. Suhara, J. Komiya, and A. Lopatenko. 2017. A Lightweight Front-end Tool for Interactive Entity Population. *CoRR* abs/1708.00481 (2017).

[8] Jay M Ponte and W Bruce Croft. 1998. A language modeling approach to information retrieval. In *SIGIR '98.* ACM, 275–281.

[9] Yves Rasolofo and Jacques Savoy. 2003. Term proximity scoring for keyword-based retrieval systems. In *European Conference on Information Retrieval.* Springer, 207–218.

[10] Sheikh Muhammad Sarwar, John Foley, and James Allan. 2018. Term Relevance Feedback for Contextual Named Entity Retrieval. In *CHIIR '18.* 301–304.

[11] B. Settles. 2011. Closing the loop: Fast, interactive semi-supervised annotation with queries on features and instances. In *EMNLP '11.*

[12] A. Vercoustre, J. Thom, and J. Pehcevski. 2008. Entity ranking in Wikipedia. In *SAC '08.*