Sentence Retrieval for Entity List Extraction with a Seed, Context, and Topic

Sheikh Muhammad Sarwar
University of Massachusetts Amherst
smsarwar@cs.umass.edu

John Foley
Smith College
jjfoley@smith.edu

Liu Yang
University of Massachusetts Amherst
lyang@cs.umass.edu

James Allan
University of Massachusetts Amherst
allan@cs.umass.edu

ABSTRACT

We present a variation of the corpus-based entity set expansion and entity list completion task. A user-specified query and a sentence containing one seed entity are the input to the task. The output is a list of sentences that contain other instances of the entity class indicated by the input. We construct a semantic query expansion model that leverages topical context around the seed entity and scores sentences. The proposed model finds 46% of the target entity class by retrieving 20 sentences on average. It achieves 16% improvement over BM25 in terms of recall@20.

KEYWORDS

Sentence retrieval; entity list extraction

1 INTRODUCTION

Consider a user searching for *a list of civilians killed by the New York Police Department*, who issues that query to a search engine. She lands on a web page where she finds the sentence: "*On Feb. 4, 1999, four NYPD officers in the Bronx fired 41 shots at a 22-year-old immigrant from Guinea named* ***Amadou Diallo***."[1].

Can she use the information she now has – a query and an example sentence with an instance of her goal identified – to find other mentions of killings? Knowledge bases such as Wikipedia rarely contain articles about non-popular entities such as "Amadou Diallo", so we cannot adopt entity retrieval based approaches that search through knowledge base articles on entities organized by entity categories [21]. Document co-occurrence based retrieval models would find a large number of unrelated entities co-occurring with *New York Police Department* in different contexts and thus result in low precision [2]. Corpus-based set expansion methods can be useful, but they require more than one seed to infer the context and type of the desired entity class and do not take a query as input [20].

An information extraction approach to this problem is to construct a weak supervised training dataset and estimate a statistical NLP model (e.g., feature-rich logistic regression, CNN, CRF) [11]. Such a dataset is usually constructed by automatically labeling sentences with relevant instances from a knowledge base or a historical list. In the case of our example, the absence of a manually curated historical database of NYPD police killing would make this process infeasible.

Active Learning (AL) based approaches could be used to gather informative sentences using a human-in-the-loop setting [7]. However, we have just one training sentence and a statistical model estimated using that would produce ineffective queries for users. We need to retrieve at least a handful of sentences containing target entity instances to support an AL approach [6]. If we can retrieve such sentences and annotate them, the model would have some positive instances and might then be capable of learning actively. We focus on sentences because a sentence offers more context for interpreting entity relevance [7]. Moreover, retrieval of these sentences is also useful for other downstream applications like summarization, entity comprehension and retrieving entities in context [13, 18, 24].

In this work, we take an IR approach, retrieving sentences containing entities from a user desired list. Our contributions are: (1) we show that distant supervision with an entity annotated in the input training sentence can be effective if a query expansion technique is used; (2) we describe a semantic relevance model for query expansion and show that it is able to exploit entity co-occurrence; and, (3) we perform experimental evaluation with TREC List QA datasets and show that on average the top 20 sentences found by our approach contain nearly half of the target entities.

2 RELATED WORK

Related work comes from Corpus-based Set Expansion (CSE) , List Completion (LC), and Entity Ranking (ER). Each task is similar to ours but uses more seeds, external resources or lacks an example sentence for context.

Seed Selection for Active Learning. Active learning (AL) is a setting where a classifier determines which unlabeled data would contribute most to its learning process, and asks a human judge to label those data points [19]. Dligach and Palmer addressed the problem of seed selection for AL [6]. They considered a scenario where rare class examples constitute 5% of the data. They stated that 10 randomly selected seeds for AL would have 60% chances of

[1]From 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

ACM ISBN 978-1-4503-6881-0/19/10…$15.00
https://doi.org/10.1145/3341981.3344250

selecting none of the rare class examples. They proposed Language Model (LM) based sampling for selecting seed words for the Word Sense Disambiguation task. Our work focuses on sentences selection for Information Extraction and hypothesizes that a semantic retrieval based approach is more suitable.

List QA. List questions are a special class of questions, where a system has to come up with a set of entities in response to a question [4, 22]. Systems built for solving this task often leverage some form of well-trained entity recognizer to collect the most promising answer passages [12, 16]. When the target type of a question is known in advance (e.g., *who* suggests *person*), identifying passages with that type of entity helps narrow down the set of candidate results. However, these approaches do not generalize to arbitrary types.

Set Expansion and List Completion. In the List or Set Completion task a small set of seed entities is given, with the aim of retrieving a complete set of target entities. Our work does not assume the existence of a knowledge-base, which makes it differ dramatically from set expansion.

Entity Ranking Systems. Our proposed task is also closely related to the entity ranking task in INEX 2009 [5] and TREC 2010 [1]. Approaches taken by participants in these tasks included the use of co-occurrence models, NER based type filtering, and context modeling [8]. Co-occurrence modeling was further applied based on context. The relation between a pair of entities was measured by their co-occurrence in documents (context-independent) and by computing similarity of the term vectors extracted from those documents (context-dependent) [3]. Some solutions augmented entity representation with entity synonyms and used search engines with that representation. Most of the participants of this task adopted external resources and were heavily dependent on Wikipedia for performing entity type filtering [8]. We study retrieving sentences with target entities by focusing only on textual content around entities.

3 TASK DEFINITION

We are given a collection of sentences, $S = \{S_1, \ldots, S_n\}$ drawn from a collection of documents, D_q, identified by a list query, q. We also have a training sentence S_q with a set of entities E_q from S_q identified. In our experiments, $S_q \in S$, but that is not required. E_q will normally have one entity in it, but we allow the possibility that a sentence lists more than one name of the target type. Broadly speaking, our objective is to define a function $F(S_i, S_q)$ that can score any sentence $S_i \in S$ based on the likelihood it contains a previously unseen entity of the same type characterized by E_q. Here, the type of E_q is implicitly indicated by the contextual information presented in the training sentence.

This task requires a relevant sentence to contain relevant as well as *novel* entities. Our evaluation set consists of a set of target entities, E_t belonging to the same type as E_q. Our scoring function $F(\cdot)$ scores each $S_i \in S$ to produce a ranked list of sentences. We define E_k as the union of E_q and the subset of E_t that has been seen in sentences at ranks 1 to k, with $E_0 = E_q$. We count a sentence at rank k as relevant only if $E_k \neq E_{k-1}$ – that is, if a new target entity occurs. Relevance thus incorporates novelty in this evaluation. Our goal is to show as many distinct entities as possible in the top k retrieved sentences, because more relevant entities with their context would give us more useful training data for AL.

4 PROPOSED APPROACH

We consider both topical and functional similarities to rank candidate sentences. To capture topical similarity, we use a sentence embedding-based approach to rank candidate sentences from S by the likelihood that their content matches the training sentence, S_q. For the embedding-based retrieval step, we compute the similarity of the sentence embedding of S_q to embeddings of all sentences from S. We find that despite the expansion implicit in embeddings, there is value in further expansion of S_q using explicit query expansion methods. Furthermore, we achieve functional similarity by looking at the NER tag of the entities in a candidate sentence, which we refer to as type refinement approach.

4.1 Sentence Embedding (SE) based Retrieval

In order to induce a semantic representation of our short sentences, we lean on the the distributed representation of words learned by the skip-gram model of `word2vec` [15]. The sentence embedding of any sentence S_i is computed as the average of word embedding vectors for each token $t \in S_i$. Wieting et al. [23] showed that a simple averaging over the embedding of the words in a sentence provides an effective representation for that sentence in two supervised NLP tasks: sentence similarity and sentence entailment. We do not have sentence similarity labels to train a model, and so we adopt this simple method of averaging.

We compute the similarity score for S_i given query S_q using cosine similarity as shown in Equation 1, with V being a function that returns the SE vector for its argument.

$$Score_{SE}(S_q, S_i) = \cos[V(S_q), V(S_i)] = \delta(S_i, S_q) \tag{1}$$

4.2 Query Expansion

To create a broader and more generalized representation of the training sentence S_q, we build on the well-known IR technique of query expansion (QE). For expanding S_q in this problem we select a set of $k < n$ sentences from D_q, $Q_e = \{S_1, S_2, ..., S_k\} \subset S$ and combine Q_e with S_q to create an improved representation, S_q^+. We hypothesize that we can obtain a better representation of user intent with query expansion using the entity set E_q from S_q. As a result we construct $Q_e = \{S_i \mid e \in E_q \wedge e \text{ is an entity in } S_i \wedge S_i \neq S_q\}$.

Thus we enrich our query representation by drawing in context of the query entity set E_q. This approach is referred to as distant supervision and it is widely adopted by the NLP community [11]. However, distant supervision is generally applied when there is much more than one example. As it is difficult to learn a statistical model using distant supervision with a single training example, we construct a query model using the distantly supervised data. We take two different approaches to query model construction.

4.2.1 ***Query Expansion with Sentences (SQE).***. After obtaining Q_e, we compute the expanded representation S_q^+ of S_q using the following equation:

$$S_q^+ = \frac{1}{\alpha_0 + \sum_{i=1}^{k} \alpha_i} \left(\alpha_0 V(s_q) + \sum_{i=1}^{k} \alpha_i V(S_i) \right) \tag{2}$$

Informally, S_q^+ is the weighted average of the embeddings of the sentences in Q_e. The weight α_i of sentence $S_i \in Q_e$ is calculated as $Score_{SE}(S_q, S_i)$.

4.2.2 ***Query Expansion with Terms (TQE).*** We compute an embedding based query language model for expanding the sentence query S_q. Given a set of expansion sentences Q_e, we compute the expanded query model θ_q using:

$$P(t|\theta_q) = \sum_{S_i \in Q_e} P(t|S_i)P(S_i|S_q) = \sum_{S_i \in Q_e} P(t|S_i)P(S_i|S_q)$$
$$= \sum_{S_i \in Q_e} P(t|S_i) \frac{\delta(S_i, S_q)}{\sum_{S_i \in Q_e} \delta(S_i, S_q)} \propto \sum_{S_i \in Q_e} P(t|S_i)\delta(S_i, S_q)$$

For a sentence S_i, $P(t|S_i)$ can be estimated as $\frac{1}{len(S_i)}$, and $\delta(S_i, S_q)$ can be estimated as the cosine similarity of the embedding of S_i and S_q. After obtaining $P(t|\theta_q)$, we normalize the distribution, and finally compute the expanded representation S_q^+ of S_q using:

$$S_q^+ = \sum_{t \in \theta_q} P(t|\theta_q)V(t)$$

4.3 NER-based Refinement for Novelty

Our unit of retrieval is a sentence, and we want that sentence to contain *novel* entities of the target entity type. We use a standard NER tool to find the broad entity type from the query (e.g., person) and discard any candidate sentences that does not contain at least one unseen instance of the same broad entity type. As an example, assume that our target entity type is *presidents of United States*, and the broad entity type is *person*. Now, if a sentence at position k in the ranked list does not contain at least one new *person* instance, we assume that this sentence will thus not have any new president instance and we discard it to refine the ranked list.

This approach has a chance to hurt some queries when the NER tagger is wrong (e.g., the type of the target entity "Chicago" is *movie*, but the broad entity type identified by the NER tagger is *location*), but it improves overall performance in this study.

5 EXPERIMENTS

5.1 Experimental Setup

5.1.1 ***Dataset.*** We use a dataset introduced by Foley et al. for entity list extraction [10]. They selected 120 list questions and their corresponding answer set from the TREC 2005 and 2006 QA datasets. They excluded list questions that seek common entities like countries, cities, etc. An example question from the refined dataset is *list all the graduates of DePaw University*. For each list question the dataset provides sentences from the top 1000 documents retrieved against the question. The document ranking is provided by TREC track organizers using BM25 ranking algorithm. For reproducibility purposes, we use the same sentence dataset, even though using a more modern retrieval algorithm may result in better overall results.

The dataset is particularly suitable for us considering the motivating example we provided in introduction - a user queries a search engine at first with her list information need. We call that set D_q and draw our candidate set of sentences, S, from there, resulting in a sentence corpus containing an average of 25,229 sentences per query. If we randomly choose one sentence from the corpus, there is 0.7% probability of seeing a sentence that contains a target entity. Thus, only a small subset of these sentences are entity bearing ones. We found this small subset by matching all the sentences against the answer keys of the list question provided by TREC. Thus, for the list question asking about *the graduates of DePaw University*, we formed a large sentence corpus in which only a handful of sentences contain names of the graduates. We judged and selected our query sentence S_q from those (120 sentences for 120 list queries) and evaluated the effectiveness of our approaches in retrieving the remaining sentences.

All sentences in the dataset are annotated with the Stanford Named Entity Recognizer [9]. Each sentence term receives a label from the set *{OTHER, ORGANIZATION, NUMBER, DOLLAR, LOCATION, TIME, MISC, PERSON, ORDINAL, DURATION, PERCENT, MONEY}*. Among the 120 query sentences 10, 17, 1, 51, 5 and 36 contain instances of LOCATION, ORGANIZATION, NUMBER, PERSON, MISC, and OTHER entities, respectively. We note that 34% of the entities identified are of OTHER type, which suggests that this dataset is challenging and perhaps interesting to the NER community as well. Moreover, the task gets harder with the novelty requirement, as a sentence would not be relevant in the ranked list if it does not contain a new entity of the target entity class.

5.1.2 ***Training Word Embeddings.*** We train word embeddings using the Stanford-NLP lemmatized form of the AQUAINT text corpus. Using the `word2vec` tool by Mikolov et al. [15], we derived 200-dimensional skipgram embeddings with a context window of 5, no negative samples, and the recommended sampling threshold of 10^{-5}. The raw text of the AQUAINT corpus is 1.6 GB and it contains 517M words. These skipgram models were recommended as the most robust by Levy et al. [14]. For reproducibility, we release our word embeddings alongside our data.

5.2 Experimental Results Analysis

The upper portion of Table 1 shows a comparison of our baselines for the proposed task: SE, BM25 and Conditional Random Field (CRF) model. We found that semantic search works better than keyword based search, and classic information extraction approach, CRF fails with one training sentence and a tagged entity. SE brings 90% of the entities in the top 1000 sentence, whereas BM25 finds 82% (not reported in table). High entity recall in the top 1000 sentences is necessary for our type-refinement approach. As a neural sentence retrieval baseline, we tried to train a siamese recurrent neural network [17] on Stanford SNLI sentence similarity corpus and applied it on our dataset for scoring candidate sentences. It performed poorly because of domain adaptation issues. We do not report those results for that reason.

The middle portion of the table shows the performance of sentence based query expansion explained in Section 4.2. Using the weight set $W = \{\alpha_0, \ldots, \alpha_n\}$ in Equation 2, we put uniform weights on the ranked query entity-bearing sentences to compute a weighted query representation. We also used the semantic similarity scores between query sentence and expansion sentences to obtain W. Both

Table 1: Performance of embedding based approaches with query expansion where MAP is measured at depth 1000. $*$ represents significant ($P < 0.05$) improvement over the BM25 baseline and † represents significant ($P < 0.05$) improvement of term based QE method over the best sentence based QE method measured by the Student's paired t-test.

QE	QE-Weight	Ranking	R@10	R@20	P@10	P@20	MAP
None	N/A	BM25	0.268	0.396	0.125	0.098	0.186
None	N/A	SE	0.277	0.405	0.139	0.107	0.192
None	N/A	CRF	0.164	0.221	0.091	0.072	0.121
Sentence	SQE with uniform weights to Q_e	SE	0.352*	0.461*	0.167*	0.121*	0.249*
	SQE with $Score_{SE}(S_q, S_i)$ as weights to Q_e	SE	0.351*	0.461*	0.167*	0.120*	0.249*
Term	TQE with top k unweighted terms from Q_e	SE	0.280*	0.362	0.134*	0.092	0.209*
	TQE with top k weighted terms from Q_e	SE	0.375*	0.460*	0.167*	0.117*	0.282*†

of the query expansion based approaches perform quite well compared to the non-expanded versions. However, it is evident that weights on expansion sentences do not affect the performance.

The lower portion of Table 1 focuses on term based QE (TQE) approaches explained in Section 4.2.2. We used the cosine similarity score from SE as document priors for computing term weights for TQE. The first term based method set uniform weights to all top k retrieved terms and it performs significantly worse compared to the weighted version. It shows that our term weight estimation is effective. Overall, the term and sentence based methods perform comparably in terms of precision and recall. The term based method achieves significant increase in MAP at cutoff 1000.

Interestingly, when we consider $P(t|S) = 1$ and $\delta(S, S_q) = 1$ we obtain results very close to the weighted version (Section 4.2.2). That indicates that term count in expansion sentences is the most important component of the expanded query. To understand why this happens note that for each of the expansion sentences the query entity terms are common, so they get a high probability in the final query model. After looking at the top-ranked search results we found that a majority of the sentences contain the query entity itself. It seems that the model is thus considering query entity-bearing sentences from across D_q and selecting the sentences with novel co-occurring entities at the top ranks.

6 CONCLUSION

We proposed retrieval approaches using a single training sentence to retrieve more training data. In general, this approach can be taken for any task that suffers from data sparsity. Our approach could retrieve almost half of the target entities by considering only 20 sentences. If these sentences are annotated by a user, it might then be possible to build a model capable of doing active learning. In future work, we expect to combine IR and statistical models for active learning. A limitation of our model is its struggles to retrieve non-co-occurring entities in the top ranks. We aim to improve this model to further increase recall, perhaps by de-emphasizing the original selected target.

ACKNOWLEDGEMENT

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] K. Balog, P. Serdyukov, and A. Vries. 2010. Overview of the TREC 2010 entity track. *TREC* (2010).
[2] M. Bron, K. Balog, and M. de Rijke. 2010. Ranking related entities. In *CIKM '10.*
[3] M. Bron, K. Balog, and M. Rijke. 2009. *Related entity finding based on co-occurrence.* Technical Report. Amsterdam University, NL.
[4] Hoa T. Dang, J. Lin, and D. Kelly. 2006. Overview of the TREC 2005 Question Answering Track. *TREC* (2006).
[5] G. Demartini, T. Iofciu, and A. De Vries. 2010. Overview of the INEX 2009 Entity Ranking Track. In *INEX '09.* 254–264.
[6] D. Dligach and M. Palmer. 2011. Good seed makes a good crop: accelerating active learning using language modeling. In *ACL '11.*
[7] A. Esuli, D. Marcheggiani, and F. Sebastiani. 2010. Sentence-Based Active Learning Strategies for Information Extraction. In *IIR' 10.*
[8] Y. Fang and L. Si. 2015. Related Entity Finding by Unified Probabilistic Models. *World Wide Web* (2015), 521–543.
[9] Jenny R. Finkel, T. Grenager, and C. Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *ACL '05.*
[10] J. Foley, S. Muhammad Sarwar, and J. Allan. 2018. Named Entity Recognition with Extremely Limited Data. In *LND4IR '18.*
[11] K. Keith, A. Handler, M. Pinkham, C. Magliozzi, J. McDuffie, and B. O'Connor. Identifying civilians killed by police with distantly supervised entity-event extraction. In *EMNLP '17.* 1547–1557.
[12] C Lee, Y. Hwang, H. Oh, S. Lim, J. Heo, C. Lee, H. Kim, J. Wang, and M. Jang. 2006. Fine-grained named entity recognition using conditional random fields for question answering. In *AIRS '06.*
[13] J. Lee, A. Fuxman, B. Zhao, and Y. Lv. Leveraging Knowledge Bases for Contextual Entity Exploration. In *KDD '15.*
[14] O. Levy, Y. Goldberg, and I. Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *TACL* (2015), 211–225.
[15] T. Mikolov, T. Chen, G Corrado, and J. Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).
[16] D Mollá, M. Van Zaanen, D. Smith, and others. 2006. Named entity recognition for question answering. In *ALTW '06.*
[17] J. Mueller and A. Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. In *AAAI '16.*
[18] S. Muhammad Sarwar, J. Foley, and J. Allan. Term Relevance Feedback for Contextual Named Entity Retrieval. In *CHIIR '18.*
[19] B. Settles. Closing the loop: Fast, interactive semi-supervised annotation with queries on features and instances. In *EMNLP '11.*
[20] Jiaming Shen, Zeqiu Wu, Dongming Lei, Jingbo Shang, Xiang Ren, and Jiawei Han. 2017. Setexpan: Corpus-based set expansion via context feature selection and rank ensemble. In *ECML PKDD '17.*
[21] A. Vercoustre, J. Thom, and J. Pehcevski. 2008. Entity ranking in Wikipedia. In *SAC '08.*
[22] E. Voorhees and Hoa T. Dang. 2005. Overview of the TREC 2005 Question Answering Track. *TREC* (2005).
[23] J. Wieting, M. Bansal, K. Gimpel, and K. Livescu. Towards Universal Paraphrastic Sentence Embeddings. In *ICLR '16.*
[24] S. Hwang H. Wang X. Sean Wang W. Wang Y. Zhang, Y. Xiao. 2017. Entity Suggestion with Conceptual Expanation. In *IJCAI '17.* 4244–4250.