Query by Example for Cross-Lingual Event Retrieval

Sheikh Muhammad Sarwar
CIIR, College of Information and Computer Sciences
University of Massachusetts Amherst
smsarwar@cs.umass.edu

James Allan
CIIR, College of Information and Computer Sciences
University of Massachusetts Amherst
allan@cs.umass.edu

ABSTRACT
We propose a Query by Example (QBE) setting for cross-lingual event retrieval. In this setting, a user describes a query event using example sentences in one language, and a retrieval system returns a ranked list of sentences that describe the query event, but from a corpus in a different language. One challenge in this setting is that a sentence may mention more than one event. Hence, matching the query sentence with document sentence results in a noisy matching. We propose a Semantic Role Labeling (SRL) based approach to identify event spans in sentences and use a state-of-the-art sentence matching model, Sentence BERT (SBERT) to match event spans in queries and documents without any supervision. To evaluate our approach we construct an event retrieval dataset from ACE [20] which is an existing event detection dataset. Experimental results show that it is valuable to predict event spans in queries and documents and our proposed unsupervised approach achieves superior performance compared to Query Likelihood (QL), Relevance Model 3 (RM3) and SBERT.

KEYWORDS
Cross-lingual IR, Event Retrieval, Sentence BERT

ACM Reference Format:

1 INTRODUCTION
Query by Example (QBE) is an effective alternative to keyword queries for identifying user information need. It has been applied to retrieve entities and documents from unstructured text corpora [16, 17, 19], entities from knowledge graphs [9], and tuples from relational databases [5]. QBE approaches are motivated by the fact that it is often easier for a user to express an information need with examples rather than a natural language description. Consider the case where a user wants to find all the jail release events from a corpus. To start this process, she retrieves a few documents with combination of keywords such as jail, release, sentence, etc., and finds sentences from those documents that mention a jail release event. Although these sentences constitute a representation of her information need (query), traditional retrieval approaches do not provide support for such an event query. A sentence matching model that computes similarity between a pair of sentences can be a remedy to this problem. However, our experiments suggest that performance of a state-of-the-art unsupervised sentence matching model is sub-optimal for event matching.

We study the above mentioned event matching problem in a cross-lingual setting - i.e., we assume that the language of example sentences and corpus sentences are different. Although Cross-Lingual Information Retrieval (CLIR) is a well-studied problem, most CLIR studies are targeted towards document retrieval [18]. To the best of our knowledge, there is no study or available testbed for studying CLIR or even mono-lingual IR for example-driven event retrieval. Such a setting would be very useful for journalists, security agency personnel, and political scientists. This motivated us to create a testbed and evaluate standard retrieval approaches for our task, Cross-Lingual Event Retrieval with Query by Examples (CLER-QBE).

To solve CLER-QBE, we follow a popular CLIR approach that uses two stages: query translation and retrieval [12]. We translate example sentences that constitute our event query using a commercial Machine Translation (MT) system and focus on the retrieval problem. It is challenging to retrieve sentences containing a target event with translations of examples sentences for two reasons: i) translated example sentences are noisy because of MT error; ii) only a sub-sequence of tokens in the translated example sentences describes the target event that holds for corpus sentences too. Both these issues make it challenging to understand user intent and match event mentions in translated examples and corpus sentences. They result in a phenomenon we refer as noisy matching.

To alleviate the effect of the noisy matching problem, we assume to have event trigger annotation for our example sentences. Consider the sentence describing a jail release event: “Pasko, whose sentence included time served, was released in January for good behavior after serving more than two-thirds of the sentence”. Note mention of three events: sentence, jail release, and sentence serving completion. We assume that a user interested in the jail release event would provide us with the trigger keyword released along with the example sentence so that we can extract appropriate context around the trigger to understand the user intent. This is still problematic from the perspective of retrieval because even if we are able to extract the appropriate query, we do not know what span of the document we should match with the query context, as documents could also contain more than one event.

To extract event extents from documents and match them with query context we propose to use PredPatt, an unsupervised technique for Semantic Role Labeling (SRL) [22]. PredPatt identifies the predicates and their corresponding arguments from a sentence. We use that information to predict event spans in documents. Once the document event spans are identified, we match them with query context using a recently proposed Sentence-BERT (SBERT) model [14]. The original BERT model does not provide effective out-of-the-box sentence embeddings without fine-tuning [14]. SBERT is fine-tuned with Natural Language Inference (NLI) data and it is able
to create sentence embeddings that significantly outperforms other state-of-the-art models on semantic textual similarity tasks. Finally, to describe our contributions concisely, we propose the task of CLER-QBE, construct a standard testbed, evaluate classical retrieval approaches on that, and propose an effective SRL-based technique to predict document event spans as well as an unsupervised matching model to match query context with the predicted spans.

2 PROBLEM FORMULATION

\[ Q_e = \{s_1^{src}, s_2^{src}, \ldots, s_n^{src}\} \] is an event query that consists of \( n \) example sentences mentioning a target event, \( e = \{s^1, s^2, \ldots, s^n\} \) in src language. For example, \( Q_{\text{jail release}} = \{s_1^{\text{Arabic}}\} \) indicates that a user has provided an example sentence describing a jail release event in Arabic and wants to retrieve sentences describing jail release events in another language. \( Q_e \) is issued against a corpus, \( D_{trg} = \{d_1^{trg}, d_2^{trg}, \ldots, d_m^{trg}\} \) of \( m \) sentences written in trg language.

There is a relation, \( E V e n t(d_i^{trg}) \subset E = \{e_1, e_2, \ldots, e_l\} \) that maps a sentence \( d_i^{trg} \) to a set of events, \( E \). We assume query event \( e \) in \( E \) for the sake of evaluation. \( E V e n t(d_i^{trg}) = \{\varnothing\} \) indicates that \( d_i^{trg} \) does not mention any event. The task is to retrieve a ranked list \( R = \{d_1^{trg}, d_2^{trg}, \ldots, d_k^{trg}\} \) of \( k \) sentences mentioning \( e \). A sentence \( d_i^{trg} \) in the ranked list is relevant if \( e \subseteq E V e n t(d_i^{trg}) \); otherwise it is non-relevant.

Our problem setting assumes that the user has annotated example sentences with event triggers or nuggets. This assumption is based on event detection literature where an event mention contains a main word or phrase that evokes the event [7, 13]. To illustrate this, we provide an example from our dataset: “Pasko, whose sentence included time served, was released in January for good behavior after serving more than two-thirds of the sentence”. This example actually describes three events: i) Pasko was sentenced, ii) he was released from jail, and iii) he served in jail. If the user annotates the example sentence with the keyword released it probably means that she is looking for jail release events. As we have user annotated triggers, our query description is further enriched as \( Q_e = (s_1^{src}, t_{src}), (s_2^{src}, t_{src}), \ldots, (s_n^{src}, t_{src}) \). Our query is a set of 2-tuples where the second element of the tuple denotes the event trigger. We use \( Q_e = \{s_1^{\text{Arabic}}, s_2^{\text{Arabic}}, \ldots, s_n^{\text{Arabic}}\} \) and \( Q_{\text{jail release}} = \{t_1^{\text{Arabic}}, t_2^{\text{Arabic}}, \ldots, t_n^{\text{Arabic}}\} \) as sentence query and trigger query, respectively.

Sentence and trigger queries based on the above example would be \( Q_{\text{jail release}} = \{\text{Pasko, whose... released... sentence.}\} \) and \( Q_{\text{jail release}}^d = \{\text{released}\} \).

3 APPROACH

Our approach consists of four components: Query Translation, Document Scoring, Matching Model and Event Span Detection.

Query Translation. One common practice in cross-lingual information retrieval is to translate a search query using an off-the-shelf MT model, and perform mono-lingual retrieval using the translated query [12]. We take the same approach - i.e., we translate \( Q_e \) and \( Q_e^{d} \) into target language using a commercial MT model to obtain \( Q_e = \{s_1^{trg}, s_2^{trg}, \ldots, s_n^{trg}\} \) and \( Q_e^{d} = \{t_1^{trg}, t_2^{trg}, \ldots, t_n^{trg}\} \), respectively.

Document Scoring. Now that our sentence and trigger queries are translated into the target language, we use a mono-lingual sentence matching model, \( M_t \) to compute similarity between our queries and documents. Given \( M_t \), a sentence matching model we compute the score of a document in the target language as, \( \text{score}(d_i^{trg}) = \sum_{j \in Q_e} M_t(s_j^{trg}, d_i^{trg}) \). Similarly, we use a model \( M_t \) to match triggers with corpus sentences and compute similarity scores using \( \text{score}(d_i^{trg}) = \sum_{j \in Q_e^{d}} M_t(t_j^{trg}, d_i^{trg}) \). Sorting the documents using the scores computed by each model results in two ranked lists that we combine using the reciprocal rank fusion approach [3]. The intuition behind combining lists is that they capture different aspects of matching. The trigger matching model does not include context while the sentence matching model includes it. We provide discussion and justification for using the ranked list fusion approach in the experimental results section.

Matching Model. Our trigger matching model, \( M_t \), is query likelihood approach. As triggers do not contain any contextual information, unigram statistics are sufficient to establish matching. As sentence matching model, \( M_t \), we use a very recent architecture, Sentence BERT (SBERT) proposed by Reimers and Gurevych [14]. SBERT adds a pooling operation to the output of BERT to derive a fixed sized sentence embedding. Similar to the authors we use the mean pooling strategy to compute a fixed size representation for sentences. With a fixed size representation of a pair of sentences we use cosine similarity to compute the similarity between them. However, one problem with event retrieval is a sentence usually mentions more than one event, which holds for both query and document sentences in our setting. To match the query event with the document event accurately we focus on the relevant part of the example sentence and the corpus sentence. The next section describes how we find these relevant parts.

Event Span Detection. Given \( Q_e \) we compute matching scores of each \( s_j^{trg} \in Q_e \) with each \( d_i^{trg} \in D_{trg} \) using \( M_t \). Before doing that we need to consider that a target event \( e \) is usually mentioned by a subsequence of tokens in the example sentence \( s_j^{trg} \). Considering the entire sentence as the search intent would result in noisy matching. To alleviate this problem we locate the trigger \( t_j^{trg} \in s_j^{trg} \) and take a window of information around \( t_j^{trg} \). As \( t_j^{trg} \) and \( s_j^{trg} \) are translations of \( t_j \) and \( s_j \), sometimes \( t_j^{trg} \) cannot be located in \( s_j^{trg} \) even if \( t_j \) appears in \( s_j \). In that case we compute word embedding similarity of \( t_j^{trg} \) and all others tokens in \( s_j^{trg} \) and select the location of the highest scored token. Assuming the location is \( l \), we consider a token span starting from \( l - w \) to \( l + w \) to capture a window \( w \) of tokens around the translated event trigger. We refer to this token span as query context. This approach also needs to be applied to documents as they may also mention more than one event.

In order to find event spans in a document we use a Semantic Role Labeling Approach (SRL) to find predicate argument structure from a sentence. Given a sentence SRL is used to answer basic questions about sentence meaning, including “who” did “what” to “whom,” etc [2]. We use an unsupervised SRL approach Predictive Patterns (PredPatt) [21] to find predicate and arguments and use those to predict event spans from documents. PredPatt is lightweight, fast, and unlike other supervised SRL approaches, it does not need to adapt to a target domain with further training [6, 22]. It uses a set of non-lexicalized, extensible and interpretable patterns on
the Universal Dependency (UD) [4] parse of a sentence to extract predicates and arguments. An important reason to select PredPatt is it works over Universal Dependency (UD) parse that enables it to extract predicate and arguments in almost any language.

To illustrate how we use PredPatt to predict event spans, consider the example provided in our problem definition section: “Pasko, whose sentence included time served, was released in January for good behavior after serving more than two-thirds of the sentence”. The predicates and their corresponding arguments found by running PredPatt on the example are shown in Table 1. We predict event spans by considering the minimum size token window that covers a predicate and all its arguments. As a result, a document \( d_{trg} \) is decomposed into \( f \) token spans i.e. \( d_{trg} = \{d_{trg}^1, d_{trg}^2, \ldots, d_{trg}^F\} \). In order to compute the score of \( d_{trg}^i \) with respect to example sentence \( z_i^e \), we take \( \max_{i_1,2,\ldots,f} M_S(z_{i_1,2,\ldots,f}^e, d_{trg}^i) \). We use the Stanford CoreNLP Arabic, and Chinese with event types annotated by human judges. There are different number of sentences in different languages and the number of event types also vary. We pre-processed the original ACE 2005 dataset and then performed an analysis of the dataset based on event types. We construct a bag-of-words query from the example sentences by including all tokens around the trigger words in example sentences to determine query context. We use existing implementations of PredPatt [22] and SBERT [3].

We use TrecTools [4] to evaluate our retrieval runs and perform reciprocal rank fusion. We use window size of five around the trigger words in example sentences to determine query context. Our adopted ACE dataset and source codes to generate all the experimental results are available [5].

**4 EXPERIMENTAL SETUP AND RESULTS**

**Dataset Construction.** We adopt the ACE 2005 multilingual event detection dataset provided by the Linguistic Data Consortium (LDC) [20] to evaluate CLER-QBE. ACE 2005 provides sentences in English, Arabic, and Chinese with event types annotated by human judges. There are different number of sentences in different languages and the number of event types also vary. We pre-processed the original ACE 2005 dataset and then performed an analysis of the dataset based on event types. We report a few frequent event types along with the number of sentences that mentions those types in Table 2.

In our processed version of ACE, each sentence is POS tagged, annotated with golden (truth) event type with event trigger span indicated, and annotated with golden entity type with entity token span indicated. We used the Stanford CoreNLP English, Arabic and Chinese libraries [8] for preprocessing. Our processed version of ACE contains 16249, 1458, and 2088 sentences in English, Chinese, and Arabic, respectively. Among them 5224, 487, and 2059 sentences mention at least one event. As English has the largest number of sentences, we construct our retrieval corpus from English. Each sentence in this retrieval corpus is relevant to a specific type of event or does not indicate an event at all. We assume each event type as a query, randomly draw Arabic and Chinese example sentences for that event type, and retrieve sentences from the English corpus to perform evaluation.

**Experimental Setting.** We use Indri search framework to index our English corpus and create relevance judgments based on ground truth event annotations. We use existing implementations of PredPatt.

<table>
<thead>
<tr>
<th>Event Type</th>
<th>English</th>
<th>Chinese</th>
<th>Arabic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Movement: Transport</td>
<td>741</td>
<td>99</td>
<td>392</td>
</tr>
<tr>
<td>Conflict: Attack</td>
<td>1510</td>
<td>74</td>
<td>455</td>
</tr>
<tr>
<td>Contact: Meet</td>
<td>280</td>
<td>44</td>
<td>190</td>
</tr>
<tr>
<td>Transaction: Transfer-Money</td>
<td>187</td>
<td>24</td>
<td>42</td>
</tr>
<tr>
<td>Life: Die</td>
<td>584</td>
<td>34</td>
<td>213</td>
</tr>
</tbody>
</table>

**Table 2:** Highly occurring events in ACE with the number of sentences describing them in different languages

---

2. https://github.com/hltcoe/PredPatt
3. https://github.com/UKPLab/sentence-transformers
5. URL provided upon publication
6. https://cloud.google.com/translate
We proposed the CLER-QBE task and took a first step to evaluate with fixed number of event classes [11]. Event detection from social word “killed” as a trigger for the event “Death”. Event detection task (SRL + Fusion), which is a reciprocal rank fusion of QL-T and information retrieval approaches as well as state-of-the-art sentence it using an existing event detection dataset. We explored classical different language from corpus language.

such as earthquake fundamentally focused towards single keyword or phrase queries an Integer Linear Programming approach. These approaches are constructed from example event descriptions, and they are in such as earthquake, for example “airport shutdown”. Finally, they summarized the contents associated with the sub-events using of time-spans in which a target event occurred and summarization of the contents in that time-span for describing the event. Rudra et al. [15] explored a similar approach retrieve disaster related information e.g., about infrastructure damage, urgent needs of affected people. They identified sub-events using noun-verb pairs that closely occur in different tweets, for example “airport shutdown”. Finally, they summarized the contents associated with the sub-events using an Integer Linear Programming approach. These approaches are fundamentally focused towards single keyword or phrase queries such as earthquake to detect events from Microblogs. Our queries are constructed from example event descriptions, and they are in different language from corpus language.

5 RELATED WORK

Event detection from unstructured text is a closely related task where an event mention in text is classified into a set of predefined event types. For example, given the sentence “A police officer killed a civilian in New Jersey today”, an event detection system identifies the word “killed” as a trigger for the event “Death”. Event detection task is generally solved using supervised machine learning approaches with fixed number of event classes [11]. Event detection from social media streams is a slightly different task which is solved using classification and summarization. For example, Alsaedi et al. [1] used a classification approach to detect disruptive events from social media and the summarized contents of such events to show sub-events of interest to users. In our setting, users will be able select examples from such summarized contents and use them as queries. This makes event detection orthogonal to what we want to achieve.

Metzler et al. [10] proposed microblog event retrieval task and used keyword queries to perform retrieval on Twitter corpus constructed over a period of time. Their approach involved detection of time-spans in which a target event occurred and summarization of the contents in that time-span for describing the event. Rudra et al. [15] explored a similar approach retrieve disaster related information e.g., about infrastructure damage, urgent needs of affected people. They identified sub-events using noun-verb pairs that closely occur in different tweets, for example “airport shutdown”. Finally, they summarized the contents associated with the sub-events using an Integer Linear Programming approach. These approaches are fundamentally focused towards single keyword or phrase queries such as earthquake to detect events from Microblogs. Our queries are constructed from example event descriptions, and they are in different language from corpus language.

We randomly sample ten sets of k-examples query and plotted the mean with 95% confidence interval.

We found that event triggers as examples are much more effective queries than example sentences. However, success of our approach in predicting event spans in examples and corpus sentences indicate that there is value in combining information from triggers and sentences. In future, we plan to extend the retrieval corpus in this dataset with ambiguous triggers so that leveraging event context from example sentences becomes more useful.

ACKNOWLEDGMENTS

This research is based upon work supported in part by the Center for Intelligent Information Retrieval, and in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via Contract No. 2019-19051600007 under Univ. of Southern California subcontract no. 124338456. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

REFERENCES

[9] Steffen Metzger, Ralf Schenkels, and Marcin Sydow. 2017. QBIES: query-by-example entity search in semantic knowledge graphs based on maximal aspects,


