Enriching Large-Scale Eventuality Knowledge Graph with Entailment Relations

Anonymous authors

Abstract

The knowledge about entailment relations between eventualities (activities, states, and events) can be helpful for many natural language understanding tasks. Conventional acquisition methods of such knowledge cannot be adapted to large-scale entailment graphs. In this paper, we construct an eventuality entailment graph (EEG) by establishing entailment relations in a large-scale discourse-relation-based eventuality knowledge graph and build the graph for million of eventuality nodes by using a three-step approach to improve the efficiency of the construction process. Experiments demonstrate the high quality of the proposed approach.

1. Introduction

Large-scale real-world knowledge graph construction is critical to the understanding of human language. Knowledge about noun phrases such as concepts (e.g., ‘apple’ IsA ‘fruit’) and named entities (e.g., ‘France’ BelongsTo ‘Europe’) have been well captured and represented in modern large-scale knowledge graphs such as Freebase [Bollacker et al., 2008], YAGO [Suchanek et al., 2007], and DBpedia [Auer et al., 2007]. On the other hand, how to capture and represent the large-scale knowledge about eventualities (activities, states, and events), which describes how entities and things act, has not been widely investigated. Recently, ASER [Zhang et al., 2020] proposes to build an eventuality knowledge graph extracted from the raw corpus with carefully designed linguistic patterns and the bootstrapping framework. In ASER, all relations among eventualities are discourse relations such as Causes or Conjunction and highly confident connectives from Penn discourse Treebank (PDTB) [Prasad et al., 2008] are used to extract those relations. One main limitation of this approach is the missing of some important relations due to the lack of corresponding linguistic patterns. One important relation missing is the entailment relation among eventualities, which describes whether one eventuality $h$ has more general meaning or is inferred by another one $p$ ($p \models h$). Thus in this work, we focus on exploring an efficient way to automatically acquire large-scale eventuality entailment relations.
Conventionally, acquiring entailment relations among textual units is known as the entailment graph (EG) construction task, where different works have different definitions of the node (i.e., verb phrase). Examples are shown in Figure 1. [Berant et al., 2011] and [Javad Hosseini et al., 2018] use typed predicates, whose predicted are marks (i.e., X and Y) rather than real words, and [Levy et al., 2014] use open-IE propositions, which are binary relations instantiated with two arguments, as the nodes. Different from their simplified definition, the structure of eventuality defined in ASER is more complex. Compared with the typed predicates, an eventuality contains real arguments, which can preserve more specific semantic meanings. Compared with the IE proposition, the eventuality extends to N-ary relations besides binary propositions, which includes more syntactic roles and also more accurate and complete semantic meanings. For example in Figure 1(c), the node ‘he post it on YouTube’ is a small dependency graph itself and contains a prepositional phrase to convey more precise meaning than the proposition he post it. At the same time, the complex structure of eventualities also brings extra challenge, which makes the approaches designed for typed predicates and IE propositions not applicable. Besides the node definition, another difference between eventuality and existing EG approaches is the scope. Existing EG construction works often focus on a specific domain such as healthcare [Levy et al., 2014] or email complaints [Kotlerman et al., 2015], where the high-quality expert annotation is available as the training signal. As a comparison, eventualities in ASER are from the open domain, whose knowledge belongs to the commonsense. Compared with the domain-specific knowledge, it is harder to get sufficient human annotation as the training signal for these commonsense knowledge. Last but not least, the efficiency of existing EG construction approaches also limits their usage for large-scale KG construction. For example, a standard method used by existing approaches is the Integer Linear Programming (ILP [Berant et al., 2011]), which could only handle 50-100 nodes. It is infeasible to directly apply existing construction approaches to ASER with millions of nodes. Though Javad Hosseini et al. [2018] uses global soft constraints for large-scale graphs, it only generalizes to type predi-
In this paper, to address the limitations of existing approaches, we propose a three-step eventuality entailment graph construction method. In the first step, we decompose each eventuality into (predicate, the set of arguments) pair and the set of arguments could be (subject, object), (subject, object, prepositional phrase) and (subject, adjective). And then, in the second step, local inference, which leverages the compositional inference on both the predicates and arguments, is conducted to build up local pair-wise entailment relations. Last but not least, to populate entailment edges globally, we carefully select the predicate entailment paths rather than all the transitive closure and then generalize to eventuality entailment paths. The global entailment inference is conducted along eventuality paths based on local entailment scores in the second step. Using those graph construction techniques, we obtain large and high-quality entailment graphs over eventualities containing more than 103 million edges. To the best of our knowledge, this is the first resource of enormous entailment relations in the general commonsense domain. Our proposed decomposition methods allow for large scale graphs to improve the coverage.

The rest of the paper is organized as follows. In Section 2, we introduce the eventuality entailment graph construction task as well as the notations. In Section 3, we introduce the details about the proposed approach. Implementation details and extensive evaluations are presented in Section 4 and 5 respectively. In the end, we use Section 6 to introduce the related works and Section 7 to conclude this paper.

2. Problem Definition

In this section, we formally define the eventuality entailment graph construction task. An eventuality entailment graph (EEG) is a directed graph where nodes are an eventuality $E_i \in \mathcal{E}$ and the edge between two eventualities $E_i$ and $E_j$ represents that $E_i$ entails $E_j$. Following the definition in [Zhang et al., 2020], each eventuality $E$ is extracted from certain syntactic patterns to keep complete semantics about an activity, state, or event and hence a verb-centric dependency graph. The task is to recognize large-scale high-quality entailment relations among eventualities.
3. Method

To acquire large-scale eventuality entailment knowledge accurately, we proposed a three-step inference framework, whose demonstration is shown in Figure 2. In the first step, we decompose complicated eventuality graphs into argument sets and predicates. Second, we map them to Probase [Wu et al., 2012, Song et al., 2011] and WordNet [Miller, 1998] and conduct local inference to acquire high-quality entailment relations. Third, to acquire more knowledge, we leverage the transitive rules to conduct global inference along the predicate entailment path. Details are introduced as follows.

3.1 Eventuality Pre-processing

For each eventuality $E_i$, as the verb can be connected to multiple arguments, it has a multi-way tree structure rather than a binary tree structure like typed predicates or propositions. Following the observations and methods in [Levy et al., 2014], propositions entail if their aligned lexical components entail. We extend from binary propositions to N-ary eventualities and decompose each eventuality $E_i$ into $(p_i, a_i)$ pairs where the predicate $p_i$ is the root node of dependency graph (i.e., center verb) and $a_i = \{t_i^l, l \in 1, ..., L\}$ is the set of arguments that have directed edge with root $p_i$. Table 2 summarizes the eventuality decomposition for the most frequent eventuality patterns in ASER. The number of argument terms in $a_i$ could be 1, 2 or 3, i.e., $L = 1, 2$ or 3. Take the eventuality ‘he post it on youtube’ in Figure 1(c) as an example, after the decomposition, we will get $p_i$ = post and $a_i$ = \{he, it, youtube\}. On top of the decomposed eventuality representation, we can then conduct local and global entailment inferences to construct the eventuality taxonomy.

Figure 2: Three-step EEG construction framework. In the first step, eventualities are decomposed into argument sets and predicates, which are mapped to Probase and WordNet to generate inference rules. In the second step, the local inference is conducted to generated entailment relations, which are denoted with red arrows. In the third step, we conduct global inference to further expand the entailment graphs. We indicate the edges from the global inference with green colors.
3.2 Local Inference

In this section, we present methods of computing local entailment scores based on aligned lexical components of decomposed eventualities. To be specific, we first compute the entailment score for the argument terms and predicates separately and then merge them with a compositional eventuality inference as the final local prediction.

3.2.1 Argument Term Inference

We first introduce how to construct entailment inference rules among all argument terms \( t \in T \), where \( T \) is the argument term set. Web-scale concept taxonomy Probase [Wu et al., 2012], where each entry is a triple (concept, instance, frequency) extracted from precise patterns, is adopted to look up hypernymy pairs with higher probability. For each \( t \in T \), we conceptualize it and select top \( k \) hypernymy candidates w.r.t. the co-occurrence probability. An example of the conceptualization for ‘apple’ is \{fruit, company, food, brand, fresh fruit\} and the hypernyms hit in the \( T \) forms feasible argument term entailment rules such as \((apple \models fruit), (apple \models company), or (apple \models food)\). We use the co-occurrence probability in the Probase as the argument term entailment score denoted by \( L_{t_{ij}} \) between \( t_i \) and \( t_j \) (\( L_{t_{ij}} = 1.0 \) if \( t_i = t_j \)). We filter out low score pairs based on threshold \( \tau \) and recognize inference rules \( TR = \{(t_i, t_j, L_{t_{ij}})|t_i, t_j \in T, L_{t_{ij}} > \tau \} \) among argument terms.

Based on the derived \( TR \), we could compute the entailment score between two aligned argument set. Given argument set \( a_i = \{t_{i1}, l \in 1, ..., L\} \) and \( a_j = \{t_{j1}, l \in 1, ..., L\} \), we define argument set entailment score to be \( L_{a_{ij}} = P(a_i \models a_j) \). The probability that argument set entailment holds is the logical OR operation over all the probability that aligned argument term entailment holds, i.e.,

\[
P(a_i \models a_j) = 1 - \prod_{l=1}^{L} (1 - P(t_{il} \models t_{jl})) = 1 - \prod_{l=1}^{L} (1 - L_{t_{il}t_{jl}}) \tag{1}
\]

1. Based on the eventuality patterns shown in Table 2, the majority of argument terms are typically noun phrases, prepositional phrases, and adjectives.

---

Table 2: Eventuality decomposition for different types. eventualities are decomposed into the format of \((\text{predicate}, \text{set of arguments})\). ‘v’ stands for verbs except ‘be’, ‘n’ for nouns, ‘a’ for adjectives, ‘p’ for prepositions, ‘s’ for subjects, and ‘o’ for objects.
3.2.2 Predicate Inference

We then introduce how to construct entailment relations among predicates \( p \in \mathcal{P} \), where \( \mathcal{P} \) is the predicate set. One thing worth mentioning is that beyond the single verbs, for some eventuality patterns, we also consider the combination of verb and the associated preposition as the predicate, where the verb itself might be semantically incomplete. One example is the ‘v-p’ combination in pattern s-v-p-o such as ‘take over’ from ‘he takes over the company’. After removing predicates whose frequency are less than five, 5,997 single verbs and 13,469 verb-prepositions are left to generate the predicate inference rules. Following [Berant et al., 2011, Tandon et al., 2015], for each predicate \( p_i \) except light verbs,\(^2\) we extract the verb entailment and direct hypernymy of \( p_i \) from WordNet to form the predicate entailment rules such as (know \( \models \) remember). Considering that verb hierarchy in the WordNet contains non-negligible noises and predicate ambiguity is also common in the extracted raw rules, we define \( L_{p_{ij}} \) to be the entailment score between \( p_i \) and \( p_j \) to quantify the reliability of the extracted rule \( p_i \models p_j \). We adopt the asymmetric similarity measure - Balanced Inclusion (BInc) [Szpektor and Dagan, 2008] over the feature vectors of \( p_i \) and \( p_j \). Since predicate entailment depends on context, i.e., predicates with different argument may have different meaning, we first select the eventualities \( E' \in \mathcal{E} \) that share the same arguments between \( p_i \) and \( p_j \). We gradually augment the context of \( p_i \) by choosing the eventuality \( E_k = (p_k, a_k) \) with the same predicate \( p_i \) whose arguments have larger probability of entailed by \( a_i \) than threshold \( \lambda \), i.e., \( \{ E_k \in \mathcal{E} \mid p_k = p_i, L_{a_{ik}} = P(a_i \models a_k) > \lambda \} \). We use point-wise mutual information (PMI) between augmented arguments and the predicate \( p_i \) as the feature vector of \( p_i \) denoted as \( \mathbf{P}_i \). The entailment score \( L_{p_{ij}} \) under \( p_i \models p_j \) is \( BInc(\mathbf{P}_i, \mathbf{P}_j) \) and the predicate entailment rules are \( \mathcal{PR} = \{(p_i, p_j, L_{p_{ij}}) \mid p_i, p_j \in \mathcal{P}\} \).

3.2.3 Compositional Eventuality Inference

After obtaining the inference rules for both the arguments and predicates, the eventuality entailment decisions could be made by their composition. We define \( L_{e_{ij}} \) as the local entailment score between the eventuality \( E_i = (p_i, a_i) \) and \( E_j = (p_j, a_j) \), and adopt a geometric mean formulation of each aligned components’ scores following [Lin and Pantel, 2001]:

\[
L_{e_{ij}} = \sqrt{L_{p_{ij}} \cdot f_{ij} \cdot L_{a_{ij}}}. \tag{2}
\]

Although predicate entailment scores \( L_{p_{ij}} \) are computed based on similar context (approximately similar types of arguments), the seed eventuality pairs \( E' \) from distant supervision may be wrong. For example we start from predicate inference rule (see \( \models \) think) and extract all the eventuality pairs whose argument pairs are identical such as \( E_l = \text{she see towel} \) and \( E_r = \text{she think towel} \). Here \( L_{e_{lr}} = L_{p_{lr}} \) but obviously the probability of \( E_l \models E_r \) should be lower. At the same time, we observe that the extracted frequencies of two eventualities in ASER are 26 and 4 respectively. To remedy this issue and based on the above observations, we propose a penalty term \( f_{ij} \) to discard the effect of polysemous predicates.

---

2. Light verb is a verb that has little semantic content of its own and forms a predicate with some additional expression such as do, give, have, make, take.
Algorithm 1: Global inference algorithm over selected entailment paths.

Input: Predicate entailment path \( e = (p_1, p_2, \ldots, p_l) \), Argument Term rules \( TR \), Predicate rules \( PR \), threshold \( \tau_a \) , threshold \( \tau_e \)

/* Generalize from predicate entailment paths to eventuality entailment paths. */

\[
edge\_set = \emptyset
\]

for \( i \in \{1, 2, \ldots, l-1\} \) do

Generate bipartite graph \( G(p_i, p_{i+1}) = (U_{p_i}, V_{p_{i+1}}, \varepsilon) \)

for \( c = (e_l, e_r) \in \varepsilon \) do

\( e_l = (p_i, a_l) \) \( e_r = (p_{i+1}, a_r) \)

if \( a_l = a_r \) or \( L_{lr}^a > \tau_a \) and \( L_{le}^e > \tau_e \) then

\( edge\_set = edge\_set \cup \{c\} \)
end
end

end

Traverse over \( edge\_set \) to generate eventuality entailment paths.

or inherent wrong extractions of eventuality:

\[
f_{ij} = \frac{P(a_i|p_i)}{P(a_j|p_j)}, \tag{3}
\]

where the probability of the argument \( a_i \) co-occurred with predicate \( p_i \) are calculated from extracted frequency.

3.3 Global Inference

In this section, we introduce how to conduct global inference to acquire more edges and make the eventuality entailment graph denser. The backbone of the global inference is the transitive property of entailment relations. For example, if we are aware of \( E_i \vdash E_j \) and \( E_j \vdash E_k \), it is highly likely that \( E_i \vdash E_k \). As the centers of eventualities are verbs, we first construct the entailment chain by predicates and then leverage argument inferences to further enrich it.

We first decompose the whole graph into predicate-centric sub-graphs and obtain 15,302 predicate inference rules \( PR \) from about 19K predicates. The organization of those \( PR \) follows the forest-like structure (437 trees totally) and most of the trees have low height (average 3) and ‘fat’ child nodes. After that, we traverse the trees from the root nodes and get the transitive paths\(^3\) such as (perceive, smell, sniff), (perceive, see, glimpse), and (perceive, listen, hark). And then, we leverage predicate entailment paths \( S = \{(p_1, p_2, \ldots, p_l) | p_i \vdash p_{i+1}, p_i \in P, i \in \{1, 2, \ldots, l-1\}\} \) to conduct the transitive inference of eventualities.

For each edge \( (p_i, p_{i+1}) \) in the entailment path \( s \in S \), we automatically construct a bipartite graph \( G(p_i, p_{i+1}) = (U_{p_i}, V_{p_{i+1}}, \varepsilon) \), which fulfills two requirements: (1) All eventualities in \( U_{p_i} \) and \( V_{p_{i+1}} \) have same or entailed arguments; (2) the predicates of eventualities

\(^3\) Edges connected to certain root nodes are removed from paths since the root nodes are either light verbs or general verbs like ‘change’, ‘act’, or ‘move’.
Anonymous authors

in $U_{p_i}$ and $V_{p_{i+1}}$ are $p_i$ and $p_{i+1}$ respectively. We use the edge weights as the local entailment eventuality score. One example is shown in Figure 2, $(chew, eat)$ is an edge in the predicate entailment path of $(crunch \models chew \models eat)$. Hence $U_{chew} = \{boy chew apple, boy chew food\}$ and $V_{eat} = \{boy eat apple, boy eat food\}$. After we traverse all the edges in the path $s$, we could generalize from the predicate entailment paths to eventuality entailment paths, i.e., $(boy crunch food \models boy chew food \models boy eat food)$. We filter out eventuality pairs whose entailment scores are less than threshold $\tau_e$ to get rid of the low-quality edges. On top of the collected eventuality entailment paths, we could further expand each node by selecting local inference eventuality relations from argument term rules e.g., $(boy crunch nut \models boy crunch food)$ and $(boy chew apple \models boy chew food)$ in Figure 2.

The overall algorithm is shown in Algorithm 1, where we start from predicate entailment paths and then generalize to eventuality entailment relations. The time complexity of algorithm is $|S| \cdot O(n^2)$, where $|S|$ means the number of predicate entailment paths and $n$ means the summation of all eventualities whose predicate are in the predicate path.

4. Implementation Details

We use ASER core version to construct the EEG and only keep the eventualities whose patterns appear in Table 2, which leads to the scale of around 10 millions. After the first step of eventuality pre-processing, we obtain 413,503 argument terms ($|T| = 413,503$) and 19,466 predicates ($|P| = 19,466$), which includes 5,997 single-word verbs. For the second step, we harvest 277,667 argument entailment rules ($|TR| = 277,667$) from Probase and 15,302 predicate entailment rules ($|PR| = 15,032$) from WordNet. When applying the above entailment rules for compositions, we consider ten possible types of eventuality entailment listed in Table 3. For eventuality pairs with different size of argument set like $(s-v-o-p-o) \models (s-v-o)$, we assume $(p-o)$ terms have little effect of entailment hence ignore them. For global inference, we traverse totally 473 predicate entailment trees and get 7,321 predicate entailment paths.

5. Evaluation and Analysis

In this section, we present detailed evaluation and analysis to show the effectiveness of the proposed approaches and the value of the resulted resource.

5.1 Evaluation Details

Following previous works [Tandon et al., 2015, Zhang et al., 2020], we employ the Amazon Mechanical Turk to annotate the quality of the collect eventuality entailment knowledge. For each type of entailment rules, we randomly sample 100 eventuality pairs and for each of them, we invite five workers to label whether the former one has more specific meaning than the latter or the latter could be inferred from the former and answer binary yes/no questions. We consider it to be agreed annotation if at least four annotators out of five select the same answer. As a result, the overall agreement is 93.5%, which indicates that the survey is clearly designed and all workers can clearly understand the task.

4. https://www.mturk.com
5.2 Result Analysis

The overall results are shown in Table 3, from which we can make the following observations:

- In terms of different patterns, s-v ⊨ s-v gets the worst performance and the reason might be that the eventualities with unary relations between predicates and arguments contain a certain amount of ambiguity, which conveys incomplete semantics such as (people find, people get). Entailment can benefit from N-ary relations and hence s-v-o-p-o ⊨ s-v-o-p-o achieves much better results.

- For the eventuality inferences of states, the perfect accuracy may result from the simple modality of s-v patterns, i.e., linking verbs could naturally entail be-verb such as smell, taste, prove etc.

- Global inference has increased the scale of entailment rules by averagely three times while it doesn’t lead to the heavy drop of the performance, which proves the effectiveness and usefulness of our proposed global inference algorithm.

5.3 Discussion

Comparison with existing EG construction approaches. Compared with conventional learning based EG construction methods [Berant et al., 2011, Javad Hosseini et al., 2018], the proposed method is unsupervised, which makes it suitable for handling large-scale open domain eventualities. Moreover, traditional methods are originally designed for typed predicates and struggle at handling the complex N-ary patterns of eventualities. In terms of the efficiency, especially for the global inference, our algorithm also outperforms previous methods [Javad Hosseini et al., 2018, Berant et al., 2011] due to its simplicity.
Error analysis. We analyze all the false positive examples. Most of errors are due to unclear semantic representations such as the eventuality whose arguments are pronouns, where the ability of disambiguation provided by pronouns is pale. On example is it leave in water and it die in water. The remaining errors mainly come from the data sparsity, which leads to failure of penalty term imposed on the local eventuality entailment score.

Potential applications. As one of the core semantic knowledge, the entailment knowledge among eventualities can be used to help many downstream tasks such as textual entailment and question answering [Javad Hosseini et al., 2018]. From another angle, the acquired entailment knowledge could also serve as the ideal testbed of probing tasks such as general taxonomic reasoning [Richardson and Sabharwal, 2019] and abductive natural language inference [Bhagavatula et al., 2019].

6. Related Work

Eventuality Knowledge Graph. Traditional event-related knowledge organizations are either compiled by domain experts (FrameNet [Baker et al., 1998]) or crowd-sourced by ordinary people (ConceptNet [Speer et al., 2017]). These resources often face the data sparsity issue due to their small scale. To solve this problem, Tandon et al. [2015] built an activity knowledge graph with one million activities mined from movie scripts and other narrative text. After that, Zhang et al. [2020] use verb-centric patterns from dependency grammars to harvest more than 194 million unique eventualities from a collection of various corpus. By doing so, these two approaches acquire much larger scale eventuality knowledge as they can get rid of the laborious human annotation. As these two approaches manually select rules to extract relations among eventualities, many important relations such as the entailment relation are still missing and we still need to devote further efforts to construct a complete eventuality knowledge graph.

Entailment Graph. Conventionally, textual entailment has been utilized in the pairwise manner [Dagan et al., 2013] while ignoring the relation dependency. To remedy that, the community started to construct entailment graphs which could help high-order/complex reasoning. Different graphs may have different node definition such as words [Miller, 1998], typed predicates [Berant et al., 2011, Javad Hosseini et al., 2018], and propositions [Levy et al., 2014] for different purposes. Typically, these graphs are high-quality, domain specific, and small scale because they are carefully annotated by domain experts. Different from them, we directly apply an unsupervised framework to construct entailment relations among open domain eventualities, which is effective and efficient.

7. Conclusion

In this paper, we propose a three-step framework of acquiring eventuality entailment knowledge. Compared with existing approaches, the proposed framework can handle much larger scale eventualities without sacrificing quality. As a result, we successfully built entailment relations among a ten-million eventuality knowledge graph. Experiments and analysis prove the quality of the collected knowledge and effectiveness of the proposed framework. All the codes and data will be released.
References


