Guided Generation of Cause and Effect

Zhongyang Li\textsuperscript{1,2}, Xiao Ding\textsuperscript{1*}, Ting Liu\textsuperscript{1}, J. Edward Hu\textsuperscript{2} and Benjamin Van Durme\textsuperscript{2}

\textsuperscript{1}Harbin Institute of Technology, China
\textsuperscript{2}Johns Hopkins University, USA
\{zyli,xding,tliu\}@ir.hit.edu.cn, \{edward.hu,vandurme\}@jhu.edu

Abstract

We present a conditional text generation framework that posits sentential expressions of possible causes and effects. This framework depends on two novel resources we develop in the course of this work: a very large-scale collection of English sentences expressing causal patterns (CausalBank); and a refinement over previous work on constructing large lexical causal knowledge graphs (Cause Effect Graph). Further, we extend prior work in lexically-constrained decoding to support disjunctive positive constraints. Human assessment confirms that our approach gives high-quality and diverse outputs. Finally, we use CausalBank to perform continued training of an encoder supporting a recent state-of-the-art model for causal reasoning, leading to a 3-point improvement on the COPA challenge set, with no change in model architecture.

1 Introduction

Causal knowledge acquisition is crucial for various Artificial Intelligence tasks, such as causal event graph construction, reading comprehension and future event prediction. We propose an approach for acquiring causal knowledge through generating multiple plausible causes (reasons, explanations) and effects (results, consequences) for a provided input sentence. As exemplified in Figure 1, we develop two conditional decoders, one per causal direction. To train such models we mine a large-scale corpus of causal expressions from open domain web text, at a scale greatly surpassing prior work. Our goal is to generate multiple distinct possible causes and effects, where each generated sentence is not intended to be a paraphrase of other candidates. To support this output diversity when conditioned on a single shared input sentence, we turn to lexically-constrained decoding [Post and Vilar, 2018; Hu \textit{et al.}, 2019a], which allows for efficiently forcing a model to produce output containing one or more provided phrases. Our constraints are derived from a resource we construct for this work, replicating a prior effort in lexicalized causal knowledge graph construction [Luo \textit{et al.}, 2016]. This graph captures causal relations as a mapping across lexical types, lemmato-lemma, but our goal is to generate naturalistic sentences with appropriately inflected morphology: we therefore develop an approach for disjunctive positive lexical constraints, where a decoder’s output must contain one of a set of provided words or phrases. In our case, these are morphological variants of the same base lemma, but our approach should benefit other applications of lexically-constrained decoding.

While there is recent work in generating story endings conditioned on a context [Guan \textit{et al.}, 2019; Wang and Wan, 2019; Luo \textit{et al.}, 2019], such work does not require generated sentences to be strictly causes or effects. The ability to propose explanations for an input sentence by generating multiple causes and effects complements this emerging line of research. To our knowledge, this is the first work to consider open-ended generation of causal sentences at a large scale.

We evaluate through carefully designed human evaluation by comparing outputs from various baselines and our proposed model, finding that our model’s outputs are preferred. We further demonstrate the usefulness of our new resource by taking a recent state-of-the-art causal reasoning system and boosting its results on the COPA test set by 3 points, relying only on continued training of the model’s encoder. Our models and resources are made publicly available.\footnote{http://nlp.jhu.edu/causalbank}

In this paper, we make the following contributions:

\footnote{Corresponding author. Performed while the first author was visiting Johns Hopkins University.}
Causal Patterns

- as, as a consequence/result of, as long as, because, because of, caused by, due/owing to, in response to, on account of, result from
- accordingly, consequently, bring on/about, give rise to, induce, in order to, lead to, result in, prevent/stop...from, and for this reason, cause, for the purpose of, if...then,
- so, so that, thereby, therefore, thus, hence

Table 1: Causal patterns (their morphological variants are ignored) used to get the CausalBank corpus. The first row of patterns belong to the EPC category, while the second row belong to the CPE category.

- proposing the task of open causal generation: producing possible causes and effects for any free-form textual event;
- construction of a corpus (CausalBank) containing 314 million CE (cause-effect) pairs;
- an extension to lexically-constrained decoding that supports disjunctive positive constraints (DPC);
- human and automatic evaluations illustrating our method can generate high-quality and diverse causes and effects.

2 Approach

As shown in Figure 2, our proposed approach for open-ended causal generation includes a data collection module (Section 2.1), a Cause Effect Graph (Section 2.2), and two DPC (disjunctive positive constraint) decoding based Transformer encoder-decoder models (Section 2.3).

2.1 CausalBank: A Sentential Causal Corpus

Existing causal corpora were not built to support our goal for open-ended causal generation given any free-form textual input: as in neural machine translation (NMT), we need a large training set with millions of examples. Thus we harvest a large causal dataset from the preprocessed large-scale English Common Crawl corpus (5.14 TB) [Buck et al., 2014]. The key guidelines of our dataset are as follows: 1) The causal relation is explicitly expressed in text with a causal pattern e.g. ‘because’; 2) The ‘cause’ and ‘effect’ arguments must both appear in the same sentence; 3) The ‘cause’ and ‘effect’ arguments can be of any length of contiguous text without overlaps between them; 4) Negative causal relations are filtered.

We do not rely on a supervised text extractor to pick out specific sub-spans of a sentence that represent a cause-effect pairing between propositions.\(^2\) We instead curate a series of patterns from previous studies [Mirza et al., 2014; Luo et al., 2016; Girju, 2003]. These patterns can be classified into two categories, according to how they are mostly used in language to convey a causal relation: 1. EPC (effect-pattern-cause) category: I am very sad because I lost my phone; 2. CPE (cause-pattern-effect) category: The earthquake resulted in many deaths. For EPC patterns, we simply take the text on the left of the pattern as effect, and take the text on the right of the pattern as cause. The case is reversed for CPE category patterns. These patterns (shown in Table 1) were applied to the Common Crawl corpus, followed by post-filtering: duplicate removal; filtering explicitly negated relations and verbs in passive voice; and restricting the cause and effect to each contain at least two tokens. This results in our CausalBank corpus, denoted here as \(B\), with 133 M EPC + 181 M CPE = 314 M \((c, e)\) \((c\) refers to cause and \(e\) refers to effect\) pairs in total. We manually evaluated 1,000 randomly sampled sentences from the corpus and found that 95% conveyed a meaningful causal relation.

2.2 Cause Effect Graph: A Lexical Causal KB

Following the method described in Luo et al. [2016] for creating a causal lexical knowledge base, we re-visit a variant of their CausalNet using the Common Crawl corpus [Buck et al., 2014]. Given a sentence such as “The storm caused a tremendous amount of damage on the landing beaches.”, this approach will harvest the lexical pairs (storm, tremendous), (storm, amount), (storm, damage), (storm, landing), and (storm, beach) as causal evidence. Stop words are removed and only pairs involving nouns, verbs, adjectives and

\(^2\)We found poor annotator agreement on span boundaries in an initial investigation on crowdsouring data for such a system; we intend to return to this in future work, investigating improvements to our results via trained extraction models for corpus pre-processing.
adverbs are retained. The extracted lexical pairs form a directed network of posited causal relations, where nodes in the network are lemmatized terms, and a directed edge between two terms indicates a causal relation, weighted by co-occurrence frequency. For comparison, Figure 3 gives a similar illustration as Figure 1 in Luo et al. [2016]. We refer to our artifact as a Cause Effect Graph (CEG); Table 5 illustrates CEG contains more causal relations than CausalNet,\(^3\) owing to the larger (5.14TB) and cleaner corpus used for extraction [Buck et al., 2014].

2.3 Guided Generation

We use Sockeye [Hieber et al., 2017] to train Transformer-based [Vaswani et al., 2017] conditional generation models, one for causes, one for effects. Sockeye supports decoding via N-best (each step greedily chooses the top best N words in beam search based on the generated tokens) and random sampling (each step randomly sampling N words from the softmax distribution based on the generated tokens). The training data (CausalBank) is processed through Byte Pair Encoding [Sennrich et al., 2016] to reduce vocabulary size.

Disjunctive Positive Constraints Decoding

Unlike in NMT, our intended outputs for a given input are diverse in meaning: we wish to generate multiple semantically distinct possible causes or effects. We induce diversity through hard lexical requirements during decoding, using causal keywords from our CEG as positive constraints on the output. A positive constraint forces the decoder to produce a sequence of tokens that contain the constrained sequence, which is achieved through a constrained beam search proposed by Post and Vilar [2018] and made efficient by Hu et al. [2019a].

Unfortunately, those prior works are restricted to conjunctive positive constraints: all items provided to the decoder must be present in the output. This is problematic in our case: our CEG maps lemmas to lemmas, and thus lemmas will form our constraints, but at generation time we do not require specific morphological inflections of our constrained terms. We wish not to constrain the decoder to a particular lemma, but to allow it to choose the best morphological form as appropriate in its context. For example, when generating a cause for “I brought an umbrella” with rain as the cause keyword, some valid cause sentences, e.g., “It rained” or “It was a rainy day.”, would not be permitted based on prior work. One may circumvent this limitation by enumerating all morphological variants of a term, then apply each in turn as a positive constraint in distinct decoding passes. However, this approach does not scale, as its run-time grows exponentially in the number of initial constraints, each with multiple morphological variants.

Here we propose a solution of disjunctive positive constraint decoding, where each constraint is represented by a set of token sequences, and the decoder needs to include only one sequence from each set of constraints in the final output. We modify the algorithm from Hu et al. [2019a] to allow the decoder to explore the disjunctively constrained space in a single forward sequence, without significant computational overhead.

\(^3\)89.1M in contrast to 13.3M, with relations with a frequency of 5 or lower removed.

Algorithm 1 Decoding with Disjunctive Positive Constraints. We consider the generation of one sentence with a beam size of 1 for simplicity. Note that while a beam size of 1 reduces the constrained beam search, the handling of DPC is not affected.

\[\text{input:} \text{a set of disjunctive constraint sets } t, \text{ for each set } s \text{ in } t, s_t = \{s_t^0, s_t^1, \ldots, s_t^n\} \text{ and } s_t^n = (w_t^{(n)(0)}, w_t^{(n)(1)}, \ldots, w_t^{(n)(m)}) \text{ where } w_t^{(n)(m)} \text{ is the } m^{th} \text{ token in } s_t^n, \text{ one of the sequences of the disjunctive constraint set } s_t \]

\[\text{output:} \text{a token sequence } o = (o_0, o_1, \ldots, o_k) \]

\[\text{trie} := \text{BuildTrie}(\{s_t^0, \ldots, s_t^n\}) \]

\[\text{while } o_{k+1} = \text{EOS} \text{ and } k < k_{\text{max}} \text{ do} \]

\[o_k := \text{ConstrainedBeamSearch}(o_0, \ldots, o_{k-1}, t) \]

\[\text{if } o_k \text{ finishes the sequence } s_p^n \text{ then} \]

\[\text{for } s_p^i \text{ in } s_p \text{ do} \]

\[\text{trie} := \text{trie.prune}(s_p^i) \]

\[\text{end for} \]

\[\text{Remove } s_p \text{ from } t \]

\[\text{end if} \]

\[k := k + 1 \]

\[\text{end while} \]

\[\text{return} \ (o_0, o_1, \ldots, o_k) \]

Figure 4: Trie states in positive constraint and disjunctive positive constraint, after generating the token ‘rained’ in beam search.

In that work, constraints are represented in a trie, where each constraint is represented by a path from the root to a leaf. One or more state pointers are used to track how many tokens have been generated for each constraint, and tokens that induce more progress are prioritized in a modified beam search proposed by Post and Vilar [2018]. When a constraint is satisfied, the algorithm prunes the path representing that constraint. The distinguishing property of a disjunctive constraint is that once a sequence in a disjunctive set is satisfied, others in the set are also removed and no longer constraints.

For decoding with disjunctive constraints, we represent all constrained sequences, whether they are from the same disjunctive set or not, on a single trie. When a sequence is generated, we prune all sequences in the set as opposed to just the generated sequence. This modification gives us an efficient algorithm for applying disjunctive constraints, as illustrated in Algorithm 1 and Figure 4. While here we use morphological variants in our disjunctive set, our algorithm is broadly applicable for constraining on a set of synonyms or different subword segmentations of the same sequence.

Outputs Reranking While DPC decoding supports arbitrary number of disjunctive constraints in one beam search process, in practice only a few preferred constraints under the model will dominate any N-best output. To encourage diver-
Table 2: Dev-set results: perplexity (Per), word accuracy (Acc (%)).

<table>
<thead>
<tr>
<th>Method</th>
<th>Dataset</th>
<th>Cause</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Per</td>
<td>Acc</td>
</tr>
<tr>
<td>RNN-LSTM</td>
<td>CB_{10M}</td>
<td>66.0</td>
<td>29.6</td>
</tr>
<tr>
<td>RNN-GRU</td>
<td>CB_{10M}</td>
<td>67.6</td>
<td>29.5</td>
</tr>
<tr>
<td>CNN</td>
<td>CB_{10M}</td>
<td>37.6</td>
<td>36.1</td>
</tr>
<tr>
<td>Conv-Transformer</td>
<td>CB_{10M}</td>
<td>29.5</td>
<td>38.9</td>
</tr>
<tr>
<td>Transformer</td>
<td>CB_{10M}</td>
<td>28.3</td>
<td>39.1</td>
</tr>
<tr>
<td>Transformer_BIG</td>
<td>CB_all</td>
<td>31.4</td>
<td>38.0</td>
</tr>
<tr>
<td>Transformer</td>
<td>CB_all</td>
<td>29.9</td>
<td>38.5</td>
</tr>
</tbody>
</table>

Table 3: Human evaluation results of cause and effect generation.

<table>
<thead>
<tr>
<th>Method</th>
<th>Cause</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P@1</td>
<td>P@3</td>
</tr>
<tr>
<td>KNN</td>
<td>89.0</td>
<td>67.3</td>
</tr>
<tr>
<td>GPT-2</td>
<td>31.0</td>
<td>22.3</td>
</tr>
<tr>
<td>N-Best</td>
<td>59.0</td>
<td>45.3</td>
</tr>
<tr>
<td>Random</td>
<td>68.0</td>
<td>59.3</td>
</tr>
<tr>
<td>CN-Cons</td>
<td>72.0</td>
<td>71.3</td>
</tr>
<tr>
<td>Gold-Cons</td>
<td>78.0</td>
<td>75.3</td>
</tr>
</tbody>
</table>

3 CausalBERT

Previous studies [Phang et al., 2018; Li et al., 2019b] have shown that applying intermediate auxiliary task training to an encoder such as BERT can improve performance on a target task. We designed an intermediate task for BERT using CausalBank B, employing margin loss [Li et al., 2019a; Li et al., 2018a] in the objective function: $L(\Theta) = \sum_{(c,e) \in B} \max(0, m - f(c, e) + f(c', e')) + \frac{1}{2}\|\Theta\|^2$, where $f(c, e)$ is the score of true CE pair given by BERT model, $f(c', e')$ is the score of corrupted CE pair by replacing $c$ or $e$ with randomly sampled negative cause $c'$ or effect $e'$ from other examples in $B$. $m > 0$ is the margin loss function parameter, which is set to 0.3. $\Theta$ is the set of BERT model parameters. $\lambda$ is the parameter for L2 regularization, which is set to 0.00001.

By training BERT with this intermediate supervised task, we expect the model to acquire enhanced knowledge about the meaning of a causal relation, and can have better performance on downstream causal inference tasks.

4 Evaluation

We evaluate our proposed causal generation approach by both human and automatic metrics, and evaluate CausalBank by applying CausalBERT to COPA, which requires the model to choose the correct cause or effect from two candidates.

Model Selection

We first experiment on a small subset of our CausalBank corpus (CB_{10M}) – 10 million CE pairs from the causal pattern ‘because’ – considering different NMT encoder and decoder architectures (LSTM, CNN, Conv-Transformer [Gehring et al., 2017], and Transformer).4 For the cause generation model, $e$ is used as the source and $c$ is used as the target, which is reversed in training the effect model. Perplexity (Per) and word accuracy (Acc) are used to evaluate the model’s performance. We find that Transformer constantly achieves the best performance (Table 2).

Then we train two versions of Transformer on the whole CausalBank corpus (CB_all). The small model’s encoder and decoder both have 6 layers, with a hidden size and embedding size of 512. The big model’s encoder and decoder have 12 layers and 4 layers, with a hidden size and embedding size of 768, leading to 134M parameters in total. The vocabulary size is 15,000. The training is stopped when the validation loss stagnates for 20,000 batches. For the cause generation model, $e$ and $c$ from only the EPC category (c, e) pairs are used as the source and target. For the effect generation model, $c$ and $e$ from only the CPE category (c, e) pair is used as the source and target. This setting always generates the right part of the sentence conditioned on the left part, which we find to give more reasonable outputs than the above architecture exploration experiments. The bottom of Table 2 shows the large Transformer model constantly achieves the best performance on development set, which contains 5,000 CE pairs.

Evaluating Generation

We evaluate the large Transformer model via human assessment, on two kinds of test sets. The first kind of test sets (TrainSub) contains 100 randomly sampled input examples from the model’s training data. The second kind of test sets (COPA_Dev) contains 100 randomly sampled examples from the development set of COPA [Roemmele et al., 2011] dataset, which are manually created gold sentences and never seen during the model’s training stage.

The compared methods include a simplified KNN method (when the input is “babies cry”, we match sentences exactly containing the input as the retrieved neighbors, e.g. “those babies cry loudly”, and get the corresponding causes and effects), the GPT-2 124M language model [Radford et al., 2018], and Transformer).4 For the cause generation model, $e$ is used as the source and $c$ is used as the target, which is reversed in training the effect model. Perplexity (Per) and word accuracy (Acc) are used to evaluate the model’s performance. We find that Transformer constantly achieves the best performance (Table 2).

Each of these models’ encoder and decoder use the same architecture, e.g. both are 6-layer LSTMs, with a hidden size and embedding size of 512. All models are trained for 10 epochs. The vocabulary size is 10,000.
Table 4: Results on COPA-Test, contrasting prior results to a model by Li et al. built atop BERT-base. This model is improved by 3 points through adoption of CausalBERT.

<table>
<thead>
<tr>
<th>Method</th>
<th>Acc (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PMI [Jabeen et al., 2014]</td>
<td>58.8</td>
</tr>
<tr>
<td>PMI_EX [Gordon et al., 2011]</td>
<td>65.4</td>
</tr>
<tr>
<td>CS [Luo et al., 2016]</td>
<td>70.2</td>
</tr>
<tr>
<td>CS_MWP [Sasaki et al., 2017]</td>
<td>71.2</td>
</tr>
<tr>
<td>Google T5-base [Raffel et al., 2019]</td>
<td>71.2</td>
</tr>
<tr>
<td>BERT-base [Li et al., 2019a]</td>
<td>75.4</td>
</tr>
<tr>
<td>CausalBERT-base (ours)</td>
<td>78.6</td>
</tr>
<tr>
<td>Google T5-11B [Raffel et al., 2019]</td>
<td>94.8</td>
</tr>
</tbody>
</table>

Table 3 shows the human evaluation results. Three metrics are adopted: Precision at 1 (P@1) (an average score of 1.5 or above is seen as a valid causal answer); P@3; and average human score for each evaluated pair (H). For the TrainSub test set, the KNN method shows strong performance, especially for P@1 and the human scores. However, KNN performs worse for P@3, due to the absence of many possible answers for the same input. Meanwhile, our two versions of DPC decoding strategies (CN-cons, Gold-Cons) also show relatively better performance compared to other generation methods (GPT-2, Random and N-best decoding). KNN performs poorly on the COPA dev set, because most of the inputs never appear in the training data. However, CN-Cons and Gold-Cons can still achieve good performance.

Lexical Diversity We used a modified BLEU score to evaluate lexical diversity (Div in Table 3) where a lower score means a greater lexical diversity. Specifically, we calculate the associated BLEU-1 score between the gold answers and the generated top 3 outputs without brevity penalty. This modification ensures that we don’t reward shorter outputs. In most cases, CN-Cons gets the lowest Div scores, showing that our DPC decoding and constraint tokens from CEG together, allows us to explore more in the causes and effects space, and generate more diverse outputs. Also we find that all of these BLEU scores are very low, compared with the BLEU scores in previous text generation studies [Hu et al., 2019b; Vaswani et al., 2017]. This is because our generation task is open-ended (as illustrated in Figure 1).

Evaluating CausalBank Table 4 shows our CausalBERT results on COPA test. Compared with prior strong knowledge-driven baseline methods, a BERT-base model trained with a margin-based loss [Li et al., 2019a] achieved good performance. Following the experimental settings of Li et al. [2019a], when training the BERT-base model with additional CE pairs from CausalBank, we get an improvement of 3.2%, from 75.4% to 78.6%, showing that our corpus successfully augments BERT base to make it better for causal inference, which is a sign the corpus contains useful causal knowledge. We find that the number of CE pairs in the intermediate task matters: performance first improves and then decreases, with more training data added. We get the best performance of 78.6% with 40 K training CE pairs. Though our result still has a big gap from the current SOTA performance on COPA (94.8% from the largest google T5-11B model), the intent of our experiment is just to illustrate how the only difference was in altering the pre-training with CausalBank. One could possibly get a SOTA model based on our corpus and the google T5 model, if publicly available.

5 Related Work

Conditional Text Generation Such efforts cover a large body of work, including machine translation, response generation and paraphrase generation. Most related is conditional story generation [Guan et al., 2019; Wang and Wan, 2019; Luo et al., 2019; Li et al., 2018b], which aims to generate story continuations based on a given context. These works do not require generated sentences to be strictly causes or effects.

For causal generation, Rashkin et al. [2018] aimed to generate the likely intents and reactions of the event’s participants, given a short free-form textual event. Sap et al. [2019] trained a multi-task model for fine-grained kinds of If-Then commonsense reasoning. However, the causal semantics considered in their work are restricted to a narrow space, and their models are trained on no more than one million examples. Further, their resource was based-on crowdsourcing, which carries risks of human bias [Rudinger et al., 2017; Poliak et al., 2018]. We harvest a significantly larger, open coverage causal corpus, related in approach to DisSent [Nie et al., 2019] but larger, focused on causality, and aimed primarily at generation rather than sentence representation learning.

Of various efforts in guided generation [Ammanabrolu et al., 2019; Tang et al., 2019; Clark et al., 2018; Hu et al., 2019b],

3This was not observed in related studies [Phang et al., 2018; Li et al., 2019b], where all training examples from the Multi-NLI dataset were used as an intermediate task. Similar behavior was observed in NMT in continued training for domain adaptation [Thompson et al., 2019]. We believe ours to be a similar setting, where the “in-domain” causal data overwhelms the benefits of pretraining; adapting strategies from Thompson et al. is an avenue for future work.

6While we avoid pitfalls of elicitation, we acknowledge that like any corpus-extracted resource ours may suffer from reporting bias [Gordon and Van Durme, 2013]: some types of causes or effects that are known to humans but rarely or ever explicitly stated.
and as causal or not causal. CaTeRS \cite{Mostafazadeh} (or must not) appear in the decoder’s output.

Table 5: Contrasting size with example prior works: only the causal to constant-time, which was made more efficient by Hu et al.\cite{ConstrainedDecoding}. Causal decoding that reduced complexity from linear portion of these corpora are listed. The top are sentential causal corpora, while the bottom are graph-structure causal knowledge bases.

Lexical Causal Resources Lexical semantic resources may encode causal properties on verbs (e.g., \cite{Schuler, Bonial et al., 2014}) and prepositions (e.g., \cite{Schneider et al., 2015}). Force dynamics theory \cite{Talmy, 1988} from cognitive psychology posits three primary kinds of causal semantics \cite{Wolff, 2007} – CAUSE, ENABLE and PREVENT – which were lexicalized as causal verbs \cite{Wolff and Song, 2003}. The annotation scheme of Dunietz et al.\cite{Dunietz et al., 2017b} distinguishes three types of causal semantics: CONSEQUENTIAL, MOTIVATION, and PURPOSE. In PDTB 2.0 \cite{Prasad et al., 2008}, “CONTINGENCY” has two subtypes (“Cause” and “Condition”). FrameNet \cite{Baker, 2014} represents causal relations through a variety of unrelated frames (e.g., CAUSATION and THwarting) and frame roles (e.g., PURPOSE and EXPLANATION). These efforts motivate our own causal patterns, categorized into: CAUSE (e.g., cause, result in, lead to), EXPLANATION (e.g., because, due to), CONDITION (e.g. if-then, as long as), PURPOSE (e.g. in order to, for the purpose of), and PREVENTION (e.g. stop/prevent-from).

Causal Knowledge Acquisition Causal knowledge acquisition \cite{Radinsky et al., 2012; Radinsky and Horvitz, 2013} is crucial for many AI systems, and it is often acquired via text. Hashimoto et al.\cite{Hashimoto et al., 2014} and Kruengkrai et al.\cite{Kruengkrai et al., 2017} applied supervised learning techniques using a benchmark training data with over 100K human-annotated CE pairs. Dasgupta et al.\cite{Dasgupta et al., 2018} explored general causal extraction using 5,000 labelled sentences. Do et al.\cite{Do et al., 2011} is an example of a minimally supervised approach. Recent studies \cite{Dunietz et al., 2017a; Dunietz et al., 2018} explored new supervised approaches on the BECauSe 2.0 \cite{Dunietz et al., 2017b} corpus.

Church and Hanks \cite{Church and Hanks} proposed the use of pointwise mutual information (PMI) for mining patterns via text co-occurrence. Many works have followed this strategy, e.g. \cite{Chambers and Jurafsky, 2008; Riaz and Girju, 2010; Gordon et al., 2011; Do et al., 2011; Luo et al., 2016}. Others have mined patterns via discourse patterns in the form of ‘A led to B’, ‘if A then B’, etc., e.g., \cite{Khoob et al., 2000; Girju, 2003; Zhao et al., 2017}. See Asghar \cite{Asghar} for review. Such efforts relate most closely to our CEGraph component, rather than our overall framework. Our concern is the generation of diverse potential causes and effects as natural language statements.

6 Conclusion

We investigate open causal generation for free-form textual input, and build a large sentential causal corpus which we used to train a generative model. We introduced a novel extension to lexically-constrained decoding that supports disjunctive positive constraints, where generated output is forced to contain one of a set of candidates. Automatic and human evaluations show that our method can generate high-quality and diverse causes and effects for new inputs.

Acknowledgements

We acknowledge the support of the National Key Research and Development Program of China (SQ2018AAA0101901).

<table>
<thead>
<tr>
<th>Sentential Causal Resource</th>
<th># CE Pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCR \cite{Ning et al., 2018}</td>
<td>172</td>
</tr>
<tr>
<td>SemEval-2007 Task4 \cite{Girju et al., 2007}</td>
<td>220</td>
</tr>
<tr>
<td>CausVar-TimeBank \cite{Mirza et al., 2014}</td>
<td>318</td>
</tr>
<tr>
<td>CaTeRS \cite{Mostafazadeh et al., 2016}</td>
<td>488</td>
</tr>
<tr>
<td>EventCausalityData \cite{Do et al., 2011}</td>
<td>580</td>
</tr>
<tr>
<td>RED \cite{O’Gorman et al., 2016}</td>
<td>1,147</td>
</tr>
<tr>
<td>SemEval2010 Task8 \cite{Hendrickx et al., 2009}</td>
<td>1,331</td>
</tr>
<tr>
<td>BECAUSE 2.0 \cite{Dunietz et al., 2017b}</td>
<td>1,803</td>
</tr>
<tr>
<td>EventStoryLine \cite{Caselli and Vossen, 2017}</td>
<td>5,519</td>
</tr>
<tr>
<td>PDTB 2.0 \cite{Prasad et al., 2008}</td>
<td>8,042</td>
</tr>
<tr>
<td>Altlex \cite{Hidey and McKeown, 2016}</td>
<td>9,190</td>
</tr>
<tr>
<td>PDTB 3.0 \cite{Webber et al., 2019}</td>
<td>13 K</td>
</tr>
<tr>
<td>DisSent \cite{Nie et al., 2019}</td>
<td>167 K</td>
</tr>
<tr>
<td>CausalBank (Ours)</td>
<td>314 M</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Causal Knowledge Graph</th>
<th># CE Edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event2mind \cite{Rashkin et al., 2018}</td>
<td>25 K</td>
</tr>
<tr>
<td>ConceptNet 5.7 \cite{Speer et al., 2017}</td>
<td>473 K</td>
</tr>
<tr>
<td>ASER Core \cite{Zhang et al., 2019}</td>
<td>494 K</td>
</tr>
<tr>
<td>Atomic \cite{Sap et al., 2019}</td>
<td>877 K</td>
</tr>
<tr>
<td>CausalNet \cite{Luo et al., 2016}</td>
<td>13.3 M</td>
</tr>
<tr>
<td>Cause Effect Graph (Ours)</td>
<td>89.1 M</td>
</tr>
</tbody>
</table>

Table 5: Contrasting size with example prior works: only the causal portion of these corpora are listed. The top are sentential causal corpora, while the bottom are graph-structure causal knowledge bases.
the National Natural Science Foundation of China (NSFC) via Grant 61976073 and 61702137; the China Scholarship Council; and DARPA KAIROS (Hu and Van Durme).

References


