南京理工大学 文本挖掘实验室
Text Mining Lab, Nanjing University of Science & Technology

Paper Reading Report

Xiangqing Shen

xiangqing.shen@njust.edu.cn

Text Mining Lab (NUSTM)
Nanjing University of Science and Technology

October 7, 2021

Contents

Contents

NeuroLogic Decoding

NeuroLogic Decoding:
(Un)supervised Neural Text Generation with Predicate Logic Constraints

Ximing Lu[†‡] **Peter West**[†‡] **Rowan Zellers**[†‡]
Ronan Le Bras[‡] **Chandra Bhagavatula**[‡] **Yejin Choi**[†‡]
[†]Paul G. Allen School of Computer Science & Engineering, University of Washington
[‡]Allen Institute for Artificial Intelligence
{lux32, pawest, rowanz, yejin}@cs.washington.edu
{ronanlb, chandrab}@allenai.org

Figure 1: Paper information. [1]

[1] Ximing Lu, Peter West, Rowan Zellers, Ronan Le Bras, Chandra Bhagavatula, Yejin Choi. NeuroLogic Decoding: (Un)supervised Neural Text Generation with Predicate Logic Constraints. NAACL2021.

Task Definition

COMMONGEN (Lin et al., 2019)

input

Concept-Set food | table | sit | front

Constraints

(food ∨ foods) ∧ (table ∨ tables) ∧
(sit ∨ sits ∨ sat ∨ sitting) ∧ (front ∨ fronts)

output

Scenario

The man sat with his food at the front of the table
The food is in front of you sit at the table.
a table of food sits in front of three people

Evaluate Gender Bias in MT (Stanovsky et al., 2019)

input

Source ✓Bäckerin ✗Bäcker

The physician told the baker that she had cancer.

Constraints (Ärztin ∨ Arzt) ∧ (Bäckerin ∧ ¬ Bäcker)

out

Target

Der Arzt sagte dem Bäckerin, dass er Krebs habe.

Recipe Generation (Kiddon et al., 2016)

input

Dish name garlic butter steak

Ingredients

2 tsp butter, 1 beef steak, 1/4 tsp soy sauce, 1 tsp parsley, 1/8 tsp salt , 1/2 tsp garlic

Constraints

butter ∧ (beef ∨ steak ∨ meat) ∧ soy sauce ∧
(parsley ∨ herb) ∧ salt ∧ (garlic ∨ vegetable) ∧
(¬ pork ∧ ¬ bean ∧ ¬...) ⟵ **any extra ingredients**

output

Recipe

Mix 1 tablespoon butter, parsley, garlic and soy sauce. Sprinkle steak with salt. In a large skillet, heat remaining butter over medium heat. Add steak; cook until meat reaches desired doneness, 4-7 minutes per side. Serve with garlic butter.

Figure 2: An example for conditional text generation.

Limitations of Previous Work

- **Finetuning LMs on a dataset of task-specific examples**. However, PLMs sturggle at learning to follow these constraints.
- **Mismatch caused by a fundamental under-specification of finetuning**. Improvements come from **constrained generation** or **learning the language style**? When increasing the finetuning data fed to GPT2 by **an order of magnitude**, constraint-satisfaction with standard beam search shows **only modest improvement**.

Contributions

- Proposing **NeuroLogic Decoding**, which effectively enforces the satisfaction of given lexical constraints by controlling the decoding stage of sequence generation.
- **Converting the hard logic constraints into a soft penalty term in the decoding objective**, and use a beam-based search to find approximately-optimal solutions.
- Empirical results demonstrate that NeuroLogic Decoding ensures the satisfaction of given constraints while maintaining high generation quality, in turn leading to **new SOTA results in both the supervised and zero-shot setting**.

Prerequisite

Predicate $D(\mathbf{a}, \mathbf{y})$

Let us define a predicate $D(\mathbf{a}, \mathbf{y})$ to be a boolean function indicating the occurrence of key phrase **a** in a sequence **y**, where **a** can be either unigram or multi-gram. $D(\mathbf{a}, \mathbf{y})$ will be true iff **a** occurs in **y**.

$$D(\mathbf{a}, \mathbf{y}) \equiv \exists\, i,\ \mathbf{y}_{i:i+|\mathbf{a}|} = \mathbf{a}$$

NeuroLogic accepts lexical constraints in Conjunctive Normal Form:

$$\underbrace{(D_1 \vee D_2 \cdots \vee D_i)}_{C_1} \wedge \cdots \wedge \underbrace{(D_k \vee D_{k+1} \cdots \vee D_n)}_{C_m}$$

Notation

- Each individual constraint $D_i \rightarrow$ a *literal*.
- The disjunction of literals $\rightarrow$ a *clause*, denoted as C_j.

Prerequisite

Objective

The method seeks optimal sequences in which all clauses are satisfied:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}\in\mathcal{Y}} P_\theta(\mathbf{y}|\mathbf{x}) \quad \text{where} \quad \sum_{i=1}^{L} C_i = L \tag{1}$$

By adding a high-cost penalty term for violated constrains, constrained optimization problem $\rightarrow$ unconstrained problem:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}\in\mathcal{Y}} P_\theta(\mathbf{y}|\mathbf{x}) - \lambda' \sum_{i=1}^{L} (1 - C_i) \tag{2}$$

While exhaustive search is intractable, we use a beam-based search to find approximately optimal solutions for this objective.

Constraint States

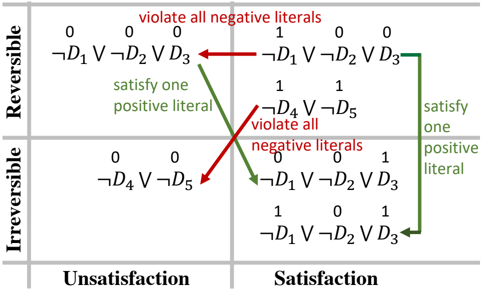


Figure 3: Clause states and possible transitions. D_i and $\neg D_i$ denote positive and negative literal respectively

S1: **reversible unsatisfaction**

S2: **irreversible unsatisfaction**

S3: **reversible satisfaction**

S4: **irreversible satisfaction**

Tracking Constraint States

Prefix Tries

- Prefix trie, $\mathcal{T}^+$ tracks *unsatisfied positive* literals from all clauses in states S1 and S3.
- Prefix trie, $\mathcal{T}^-$ tracks *satisfied negative* literals from all clauses in state S3.

How Prefix Tries Changes

- a positive literal satisfied $\rightarrow$ its clause in state S1 or S3 henceforth irreversibly satisfied (state S4) $\rightarrow$ remove all literals of that clause from both tries and stop tracking.
- a negative literal violated $\rightarrow$ remove it from the trie $\mathcal{T}^-$ $\rightarrow$ switch back to S1 or S2 once all negative literals of a clause in state S3 has been removed.

Algorithm

High-level Intuition

At at each time step, NeuroLogic selects generation hypotheses in consideration of both the objective function and the diversity of the partially satisfied constraints. We achieve such by 3 steps: *pruning*, *grouping*, and *selecting*.

Pruning Step: We first discard any h with irreversible unsatisfied clause (state S2) to focus only on candidates that might satisfy all constraints.

Grouping Step: Next, we select the beam from the pruned candidates.

Selecting Step: To select best ones from each group, we first rank candidates within a group by score function:

$$\mathbf{s} = P_\theta\left(\mathbf{y}_t \mid \mathbf{y}_{<t}\right) + \lambda \cdot \max_{\substack{D(\mathbf{a_i}, \mathbf{y}) \\ \in \text{state} \mathbf{S}1}} \frac{|\hat{\mathbf{a}}_\mathbf{i}|}{|\mathbf{a_i}|} \tag{3}$$

Algorithm

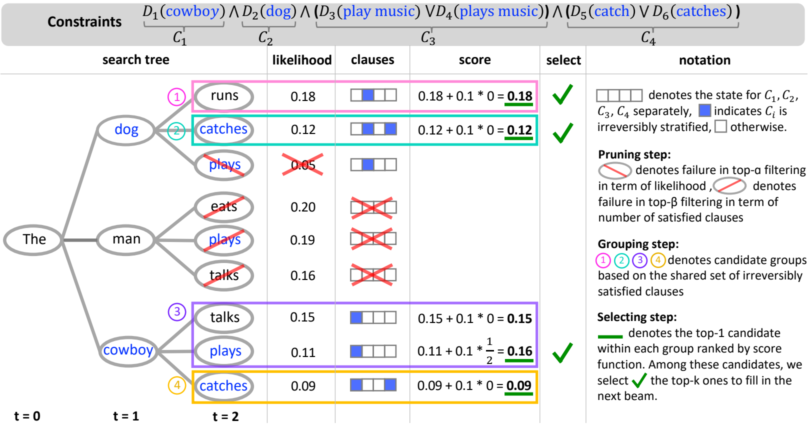


Figure 4: Illustration of the NeuroLogic decoding procedure. In this example, $k = 3$, $\alpha = 8$, $\beta = 2$, $\lambda = 0.1$.

Experiment I: Constrained Generation

CommonGen [2]

COMMONGEN is a benchmark dataset designed as a test of generative commonsense reasoning. Given a set of common concepts (e.g., dog, frisbee, catch, throw); the task is to generate a coherent sentence describing an everyday scenario using these concepts (e.g., "a man throws a frisbee and his dog catches i").

Problem Formulation

The input is an unordered set of n concepts $\mathbf{x} = \{a_1, a_2, \ldots, a_n\}$, where each concept a_i is a common object (noun) or action (verb). The expected output is a simple, grammatical sentence $\mathbf{y} \in \mathcal{Y}$ that describes a common scenario using all given concepts in $\mathbf{x}$ with correct morphological inflections.

[2] Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, Xiang Ren. CommonGen: A Constrained Text Generation Challenge for Generative Commonsense Reasoning. EMNLP2020, Findings.

Results I: NeuroLogic vs Other Decoding Methods

Decode Method	ROUGE-L	BLEU-3/4		METEOR	CIDEr	SPICE	Coverage
Greedy Decoding	35.3	25.2	16.7	25.8	10.2	24.4	80.3
Top-k Sampling	33.8	22.5	14.4	24.9	9.2	22.7	79.4
Top-p Sampling	35.3	25.0	16.5	25.7	10.2	24.1	80.1
Beam Search	40.3	34.2	24.7	27.6	13.4	27.1	82.2
Hokamp and Liu	37.6	25.6	16.8	25.9	11.1	25.1	97.2
Post and Vilar	38.3	28.1	18.6	26.7	11.8	26.0	97.4
Hu et al.	38.2	27.8	18.4	26.7	11.7	26.1	97.4
NeuroLogic	**42.8**	**36.7**	**26.7**	**30.2**	**14.7**	**30.3**	**97.7**

Table 1: Performance of different decoding methods using supervised GPT2-L on the COMMONGEN test set.

Results II: NEUROLOGIC across Different Supervised Models

Model	ROUGE - L	BLEU - 3 & 4		METEOR	CIDEr	SPICE	Coverage
GPT-2	40.3 → 42.8	34.2 → 36.7	24.7 → 26.7	27.6 → 30.2	13.4 → 14.7	27.1 → 30.3	82.2 → 97.7
BERT-Gen	42.4 → 43.8	37.5 → 38.9	27.0 → 28.2	29.5 → 30.9	14.9 → 15.5	29.8 → 31.4	89.2 → 97.3
UniLM	44.3 → **45.8**	40.6 → **42.8**	29.9 → **31.5**	30.1 → **31.7**	15.5 → 16.6	30.6 → **32.5**	90.5 → 97.8
UniLM-v2	43.5 → 44.2	39.2 → 39.5	28.3 → 28.5	30.6 → 31.3	15.2 → **16.8**	30.8 → 31.1	92.8 → 97.9
BART	43.3 → 44.7	39.9 → 41.3	29.1 → 30.6	30.4 → 31.0	15.2 → 15.9	30.6 → 31.0	95.0 → **98.7**
T5-Large	43.9 → 44.8	36.6 → 38.5	26.9 → 28.1	28.9 → 30.7	14.3 → 15.5	29.5 → 30.8	89.7 → 98.5

Table 2: Experimental results of different supervised models on the COMMONGEN test set.

Results III: NeuroLogic with Unsupervised Models

Domain Adaption	Model	ROUGE - L	BLEU - 3 & 4		METEOR	CIDEr	SPICE	Coverage
No	GPT	26.7 → 41.3	3.0 → 25.1	1.1 → 15.9	9.2 → 28.8	0.9 → 11.7	8.0 → 29.7	8.4 → **97.4**
	GPT-2	19.7 → **42.9**	4.1 → 34.4	1.5 → 23.5	11.2 → 30.7	0.4 → 13.6	7.1 → 31.4	8.3 → 96.0
Yes	GPT-2	29.8 → 42.4	9.5 → **36.1**	4.0 → **25.1**	11.7 → **31.3**	1.7 → **13.9**	8.0 → **31.8**	9.3 → 96.1

Table 3: Experimental results in zero-shot (unsupervised) setting on the COMMONGEN test set with and without language domain adaption.

Results IV: Ablation

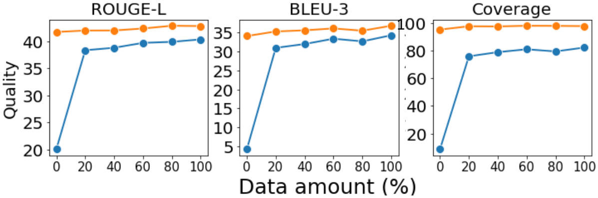


Figure 5: Performance (y-axis) of supervised GPT2-L on COMMONGEN, with a varying amount of training data for supervision (x-axis).

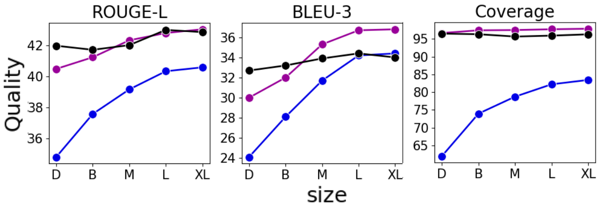


Figure 6: Performance (y-axis) of GPT-2 with varying model sizes (x-axis).

Experiment II Results: Recipe Generation

Decode Method	ROUGE-L	BLEU-3/4		METEOR	Coverage	Extra
Top-k Sampling	27.5	15.2	9.5	19.2	84.8	16.0
Top-p Sampling	28.7	17.6	11.7	19.4	86.4	15.4
Beam Search	29.4	17.4	12.0	19.7	86.5	14.3
Post and Vilar	26.1	13.6	8.8	16.5	89.6	1.15
Hu et al	26.1	13.6	8.8	16.5	89.6	1.13
NeuroLogic	**32.1**	**19.5**	**13.8**	**19.8**	**95.8**	**0.6**

Table 4: Experimental results of different decoding methods with RecipeGPT on the Recipe1M+ test set. Coverage indicates the average percentage of ingredients that are covered in the generated recipe, while Extra corresponds to the average ratio of hallucinated ingredients over the number of given ingredients.

An Example Illustration

Concept-Set {**lose**, **board**, **balance**, **fall**, **ride**}

Supervised Setting

Decode with Beam Search

[GPT-2]: Someone loses balance and falls off his bike.

[UniLM]: A man is trying to keep his balance as he falls off a board.

[BART]: A man loses his balance and falls off the balance while riding a skateboard.

[T5]: a man loses his balance on the board and falls.

Decode with NeuroLogic

[GPT-2]: A man loses his balance as he rides a roller coaster and falls off the board.

[UniLM]: Someone loses balance on the ride and falls off the balance board.

[BART]: A man loses his balance on a ride and falls off the board.

[T5]: a rider loses his balance and falls off the board.

Zero Shot Setting

Decode with NeuroLogic

[GPT]: a woman lost her balance riding a horse, falling off the horse, and hitting her head on a board

[GPT-2]: The boy lost his balance riding the bike, falling off the bike and hitting his head on the board.

Figure 7: Generated texts for the given concept-set.

Contents

TEMPORAL REASONING

Temporal Reasoning on Implicit Events from Distant Supervision

Ben Zhou[1,2] **Kyle Richardson**[1] **Qiang Ning**[3] **Tushar Khot**[1] **Ashish Sabharwal**[1] **Dan Roth**[2]
*[1] Allen Institute for AI [2] University of Pennsylvania [3] Amazon
{kyler,tushark,ashishs}@allenai.org {xyzhou,danroth}@cis.upenn.edu qning@amazon.com

Figure 8: Paper information. [3]

[3] Ben Zhou, Kyle Richardson, Qiang Ning, Tushar Khot, Ashish Sabharwal, D. Roth. Temporal Reasoning on Implicit Events from Distant Supervision. NAACL 2021.

Task Definition

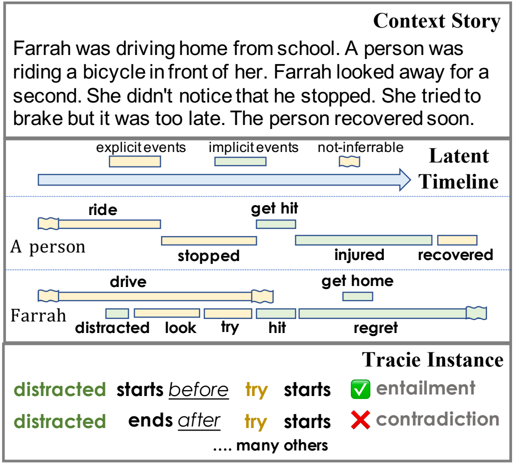


Figure 9: The task focuses on temporal relations on implicit events in short stories. A story, its latent timeline, and example TRACIE instances from it. For simplicity, events are shortened to single verbs and the timeline is exaggerated.

Contributions

- A temporal relation dataset TRACIE focusing on **implicit events**.
- A **distant supervision process** for temporal understanding of implicit events.
- A **reasoning** model that makes end-time comparisons using predictions of start-time distances and durations.

The TRACIE Dataset

Context Story (Premise)	**Hypothesis**	**Inference Label**
Tom needed to get braces. He was afraid of them. The dentist assured him everything would be fine. Tom had them on for a while. Once removed he felt it was worth it.	Tom avoids foods he can't eat with braces `starts` `before` the braces are removed.	`entailment`
We were all watching Spongebob as a family. It is a kid's show but all really enjoyed it. This one episode was especially funny for the adults. It has humor in it that is funny for kids and adults. It is something we can all watch...	The adults laughed at the jokes `ends` `before` we watch Spongebob as a family	`contradiction`
I was throwing the baseball with my son. He threw one past me that landed in the lake. I reached in to get the ball. I lost my balance and fell in. I got the ball and a bath all in one shot!	The ball was in the boys hand `starts` `after` he reached for the ball	`contradiction`

Figure 10: Example TRACIE instances. The **comparator** l ∈{`starts`,`ends`} and **relation** r ∈{`before`,`after`} in each hypothesis are highlighted, in addition to the corresponding explicit event from the story.

The Tracie Dataset

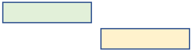
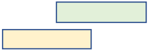
Illustration	Allen's Relation	Tracie's Relation
	Precedes, Meets	Starts Before Ends Before
	Overlaps, Finished-by, Contains, Starts, Equals, Started-by	Starts Before Ends After
	During, Finishes, Overlapped-by, Met-by, Preceded-by	Starts After Ends After

Figure 11: Tracie's label definition and its relation to Allen's interval algebra, with a graph illustration between an implicit event and an explicit event.

Implicit Event Generation

1. We randomly sample short stories from the ROCStories dataset [4].
2. For each story, one annotator writes 5 implicit event phrases that are not explicitly mentioned by the given story, but are inferable and relevant.
3. The annotator additionally rewrites two explicit events closest to the implicit event's start and end time, respectively.
4. With these two events, we can build two TRACIE instances (minus the *temporal-relation*) per implicit event, which accounts for 10 instances in total per story.

[4] Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. A Corpus and Evaluation Framework for Deeper Understanding of Commonsense Stories. NAACL2016.

Pattern-Based Pre-Training

High-level Intuition

we believe that it is more efficient to build a model that learns the prior knowledge needed for the task with distant signals and only subsequently learns the task definition through a small training set.

Distant Supervision Collection

text

I went to the park on January 1st. I was very hungry after some hiking. Luckily, I purchased a lot of food before I went to the park. I enjoyed the trip and wrote an online review about the trip on the 10th.

within-sentence

[I purchased food, I went to the park.]: **before**

cross-sentence

[I went to the park, I wrote a review]: **before**, weeks

Figure 12: Extraction for start-time comparisons applied to an example paragraph.

We describe the sources of distant supervision signals with the goal of understanding the relative order between two events' start times as well as the relative distance between them.

Supervision Instances Construction

1. Each instance comprises an event pair, a temporal relation, and an estimation on the temporal difference between the two start times.
2. Each event is a phrase constructed by taking all relevant arguments of the predicate verb in the SRL parses.
3. We represent the differences between the two start times as one of seven coarse temporal units: $\{\leq$minutes, hours, days, weeks, months, years, $\geq$decades$\}$.
4. In addition to the event pairs, we randomly sample sentences within the paragraph to use as the context that better defines the events.

Pattern-Based Temporal Model (PtnTime)

Data Format

Input sequences `event:[EventA] starts[Relation][EventB].story:[Paragraph].`

Output sequences `answer:[Label][Distance].`

PtnTime

- We use a pre-trained sequence-to-sequence model as our base model and additionally pre-train this model using the data collected.
- PtnTime serves as new set of *temporally-aware* model weights that can be used in place of existing pre-trained models and fine-tuned on Tracie.

TRACIE Task Formulation

comparator l	**relation** $r_l(\mathbf{e}_1, \mathbf{e}_2)=$	
ends	**before**	if $\mathbf{end}_1 < \mathbf{start}_2$
	after	*otherwise*
starts	**before**	if $\mathbf{start}_1 < \mathbf{start}_2$
	after	*otherwise*

Figure 13: Decomposition of the relation functions that solve TRACIE instances (equal timepoints ignored).

Neural-symbolic Model

Two Modules

- Distance function: $\mathrm{dist}(e_i, e_j) = \mathbf{start}_i - \mathbf{start}_j$.
- Duration function: $\mathrm{dur}(e_j) = \mathbf{duration}_j$.

By exploiting the rule that an end point $\mathbf{end}_j$ can be computed as $\mathbf{end}_j = \mathbf{start}_j + \mathbf{duration}_j$, we can, for example, decompose the relation $r_{ends}(e_1, e_2) = \mathbf{before}$ (i.e., e_1 *ends before* e_2) in terms of our two modules as follows via simple algebraic manipulation:

$$
\begin{aligned}
& r_{ends}(e_1, e_2) = \mathbf{before} \\
\Leftrightarrow \quad & \mathbf{end_1} < \mathbf{start_2} \\
\Leftrightarrow \quad & \mathbf{start_1} + \mathbf{duration_1} < \mathbf{start_2} \\
\Leftrightarrow \quad & (\mathbf{start_1} - \mathbf{start_2}) + \mathbf{duration_1} < 0 \\
\Leftrightarrow \quad & \mathrm{dist}(e_1, e_2) + \mathrm{dur}(e_1) < 0
\end{aligned}
$$

Distance Estimation

- We use the output from PtnTime to approximate the function $\mathrm{dist}(\cdot)$.
- Following the sequence formulation of PtnTime, we replace `[EventA]` with the textual description of e_1, `[EventB]` with the textual description of e_2, and `[Paragraph]` with the context (premise), and fix `[Relation]` to be *before*. By taking the values of the vocabulary indices corresponding to "positive" and "negative" from the logits of `[Label]` and applying a softmax operation, we get P_{before} and P_{after}. These are the probability of e_1 starting before and after e_2, respectively, and are used to define the vector $\mathbf{p} = [P_{\text{before}}, P_{\text{after}}]$.
- Similarly, we apply softmax to the logits of `[Distance]` over the 7 words representing the temporal units to obtain 7 values that approximate the probabilities of the distance between two events' start times being closest to each temporal unit. We place the 7 values in temporal units' increasing order in vector $\mathbf{d}$. To represent $|\mathbf{start_1} - \mathbf{start_2}|$ with a single value, we dot product the probabilities with an incremental constant vector $\mathbf{c} = [0, 1, 2, 3, 4, 5, 6]$. To get the direction, we apply the tanh function to the difference between the probabilities in $\mathbf{p}$.

Duration Estimation

- To obtain a model to estimate dur($\cdot$), we pre-train a sequence-to-sequence model with the duration data from [5].
- The data contains over 1 million events with their corresponding duration values.
- We map each instance to an input sequence `event:[Event]story:[Story]` and a corresponding output sequence `answer:[Value]`.

[5] Ben Zhou, Qiang Ning, Daniel Khashabi, and Dan Roth. Temporal Common Sense Acquisition with Minimal Supervision. ACL2020.

Computation and Learning

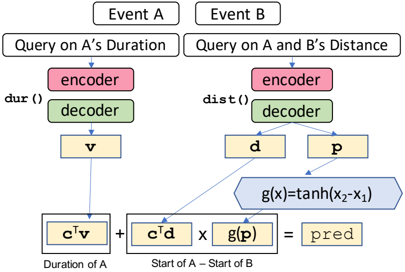


Figure 14: A schematic overview of SymTime to compare event A's end time with event B's start time via modular predictions about A's duration and distance from B and their symbolic combination (bottom).

$$\begin{aligned} \mathrm{dist}(\cdot) &= \mathbf{start_1} - \mathbf{start_2} \\ &= \mathbf{c}^T\mathbf{d} * \tanh(\mathrm{INT}_{max} * (\mathbf{p}_2 - \mathbf{p}_1)) \end{aligned} \tag{4}$$

$$\mathrm{dur}(\cdot) = \mathbf{duration_1} = \mathbf{c}^T\mathbf{v} \tag{5}$$

Results I: I.I.D Setting

System	Start	End	All	Story
Majority	57.3	69.8	64.1	18.1
BiLSTM	53.7	63.5	59.1	10.9
Roberta-Large	78.5	78.3	78.4	26.1
T5-3B	79.4	77.4	78.3	26.9
BaseLM (T5-large)	75.5	75.4	75.4	22.6
BaseLM-MATRES	76.7	76.3	76.5	25.3
PtnTime (ours)	81.4	77.5	79.3	31.0
SymTime (ours)	**82.1**	**79.4**	**80.6**	**32.0**
SymTime-ZeroShot	77.0	73.1	74.9	21.6

Table 5: Performance on Tracie, best numbers in **bold**. BaseLM is T5-large; Story is the percentage of story-wide exact match; Majority is based on the comparator and temporal-relation distribution; Zeroshot uses no TRACIE instance as supervision.

Results II: Uniform-prior Training Setting

System	Start	End	All	ΔAll
Random	50.0	50.0	50.0	-14.1
BiLSTM	50.5	51.2	50.9	-8.2
Roberta-Large	75.1	68.1	71.3	-7.1
T5-3B	72.8	68.6	70.5	-7.8
BaseLM (T5-large)	68.1	67.8	67.9	-7.5
BaseLM-MATRES	76.3	69.9	72.8	-3.7
PtnTime (ours)	80.6	73.2	76.6	-2.7
SymTime (ours)	**81.2**	**77.0**	**78.9**	-1.7
SymTime-ZeroShot	77.0	73.1	74.9	**0.0**

Table 6: Performance on TracieEa uniform-prior training setting. ΔAll compares the difference with Table 5; Majority is equivalent to random guessing.

Results III: MATRES for Explicit Events

System	OT-NS	OT	OT-MS	PT
Wang et.al.(2020)	85.9	-	-	-
BaseLM	86.0	87.5	77.4	69.0
PTNTIME	87.3	89.6	86.1	75.1

Table 7: Performance on MATRES[6] is not strictly comparable with the rest.

[6] Haoyu Wang, Muhao Chen, Hongming Zhang, and Dan Roth. Joint constrained learning for event-event relation extraction. EMNLP2020.

Results IV: Ablation

Sys.	BaseLM	PtnTime	SymTime	Human
Acc.	52.6	72.2	75.3	82.5

Table 8: Performance on *no-story* Tracie under the uniform-prior training setting.

Sys.	PtnTime	cross-sentence	within-sentence
Acc.	80.6	79.9	63.7

Table 9: Comparison of pre-training data sources on Tracie's start time prediction accuracy, under the uniform-prior training setting.

Contents

(Comet-)Atomic$^{20}_{20}$

(Comet-)Atomic$^{20}_{20}$:
On Symbolic and Neural Commonsense Knowledge Graphs

Jena D. Hwang[1*], **Chandra Bhagavatula**[1*], **Ronan Le Bras**[1], **Jeff Da**[1], **Keisuke Sakaguchi**[1], **Antoine Bosselut**[13] and **Yejin Choi**[12]
[1] Allen Institute for AI, WA, USA
[2] Paul G. Allen School of Computer Science & Engineering, WA, USA
[3] Stanford University, CA, USA
{jenah, chandrab, ronanl, jeffd, keisukes, antoineb, yejinc}@allenai.org

Figure 15: Paper information. [7]

[7] Jena D. Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, Yejin Choi. COMET-ATOMIC 2020: On Symbolic and Neural Commonsense Knowledge Graphs. AAAI 2021.

Motivation

- A new paradigm of language models as knowledge bases has emerged. In this setting, language models are **prompted** with natural language prefixes or questions, and they express knowledge through language generation.
- *Does scaling up language models actually endow them with commonsense knowledge?* They perform better when evaluated on knowledge bases that **prioritize ontological relations** and whose examples resemble language-like assertions (e.g., mango `IsA` fruit).
- Prior work has also shown that training language models on knowledge graph tuples leads them to learn to **express their implicit knowledge directly**, allowing them to provide commonsense knowledge on-demand.

Contributions

- We present ATOMIC$^{20}_{20}$—**a new commonsense knowledge graph** covering social, physical, and eventive aspects of everyday inferential knowledge.
- We compare ATOMIC$^{20}_{20}$ with other prominent CSKBs head-to-head and show that our new *symbolic* knowledge graph is **more accurate than any current CSKB**.
- We show that our new *neural* knowledge model COMET-ATOMIC$^{20}_{20}$ **successfully transfers ATOMIC$^{20}_{20}$'s declarative knowledge** to beat GPT-3, the largest pre-trained language model, in spite of using 400x fewer parameters.

Comparisons of Three CSKGs

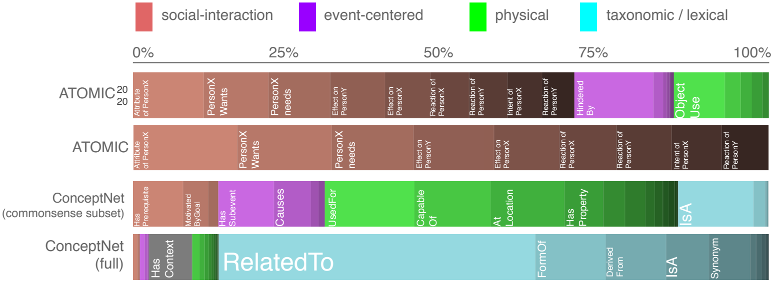


Figure 16: ATOMIC$^{20}_{20}$ tuple count distribution compared to ATOMIC and CONCEPTNET, either its commonsense subset or the full set.

Atomic$^{20}_{20}$ Illustration

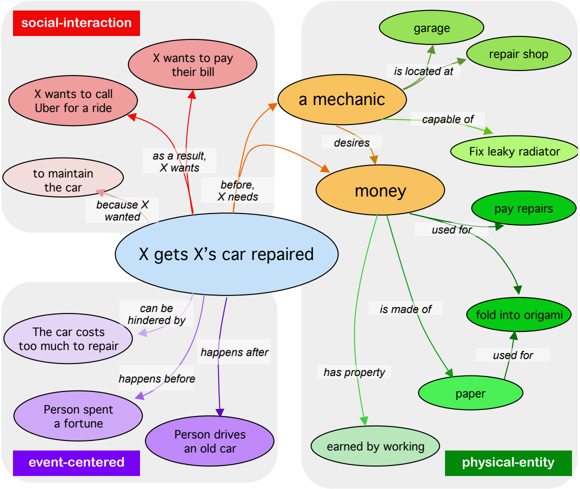


Figure 17: A tiny subset of Atomic$^{20}_{20}$, a large atlas of social and physical commonsense relations. Relations in the top-left quadrant reflects relations from Atomic.

Atomic$^{20}_{20}$ Relation

	Head	Relation	Tail	Size
Physical-Entity	bread	ObjectUse	make french toast	165,590
		AtLocation*	basket; pantry	20,221
		MadeUpOf	dough; wheat	3,345
		HasProperty*	cooked; nice to eat	5,617
	baker	CapableOf*	coat cake with icing	7,968
		Desires*	quality ingredients	2,737
		Not Desires*	bad yeast	2,838
Event-Centered	X runs out of steam	IsAfter	X exercises in the gym	22,453
		HasSubEvent	become tired	12,845
		IsBefore	X hits the showers	23,208
		HinderedBy	drinks too much coffee	106,658
		Causes	takes a break	376
		xReason	did not eat breakfast	334
Social-Interaction	X runs out of steam	xNeed	do something tiring	128,955
		xAttr	old; lazy; lethargic	148,194
		xEffect	drinks some water	115,124
		xReact	tired	81,397
		xWant	to get some energy	135,360
	X votes for Y	xIntent	to give support	72,677
		oEffect	receives praise	80,166
		oReact	grateful; confident	67,236
		oWant	thank X; celebrate	94,548

Table 10: Relations in Atomic$^{20}_{20}$ along with illustrative examples and their respective size. Relations that reflect semantically identical categories to ConceptNet is marked with an asterisk (*).

Accuracy Assessment

Knowledge Base	Accept	Reject	No Judgment
ATOMIC$^{20}_{20}$	**91.3**	**6.5**	2.2
ATOMIC	88.5	10.0	1.5
CONCEPTNET	88.6	7.5	3.9
TRANSOMCS	41.7	53.4	4.9

Table 11: **Accuracy** - Percentage (%) of tuples in the knowledge base evaluated by human crowdworkers as either always true or likely (Accept), farfetched/never or invalid (Reject), or unclear (No Judgment).

Accuracy Assessment (Breakdown)

ATOMIC$^{20}_{20}$	ATOMIC	*Relation*	CN	T-OMCS
92.3		AtLocation*	89.4	34.3
93.9		CapableOf*	84.4	50.0
94.6		Causes	90.0	50.0
96.9		Desires*	96.3	48.2
93.9		HasProperty*	86.3	52.4
82.3		ObjUse/UsedFor	**96.3**	31.6
98.5		NotDesires*	96.3	
96.9		HasSubevent	88.1	57.7
		HasFirstSubevent	**93.8**	52.4
		HasLastSubevent	**95.6**	38.2
		HasPrerequisite	**94.4**	30.0
75.4		MadeUpOf/MadeOf	**88.1**	15.9
		PartOf	**71.9**	46.5
		HasA	**77.5**	43.5
96.9		HinderedBy		
96.2		isAfter		
95.4		isBefore		
96.2		isFilledBy		
		ReceiveAction	**84.4**	56.4
91.5	86.3	oEffect		
91.5	87.7	oReact		
88.5	**89.5**	oWant		
87.7	**91.0**	xAttr		
80.8	**87.2**	xEffect		
93.1	89.9	xIntent/MotivByGoal	84.4	27.1
87.7	85.1	xNeed		
90.8	**91.3**	xReact		
96.2		xReason		
82.3	88.4	xWant/CausesDesire	**90.0**	35.9

Table 12: KG accuracy values broken down by relation. Gray cells indicate statistically significant difference from ATOMIC$^{20}_{20}$ values. Dark gray cells signal instances where ATOMIC$^{20}_{20}$ values are significantly higher than its counterpart KB. Relational *cognates* have been grouped together and *exact matches* are asterisked (*) (cf. Table 10).

Coverage Assessment

	Target KB→			
Source KB↓	ATOMIC	CN	T-OMCS	ATOMIC_{20}^{20}
ATOMIC	-	0.1	0.0	100.0
CONCEPTNET	0.3	-	5.5	45.6
TRANSOMCS	0.0	0.4	-	0.3
ATOMIC_{20}^{20}	60.2	9.3	1.4	-

Table 13: **Coverage Precision** - Average number of times (in %) a tuple in Source KB is found in Target KB.

	Target KB→			
Source KB↓	ATOMIC	CN	T-OMCS	ATOMIC_{20}^{20}
ATOMIC	-	0.3	0.0	60.1
CONCEPTNET	0.1	-	0.3	8.9
TRANSOMCS	0.0	7.6	-	1.3
ATOMIC_{20}^{20}	100.1†	47.8	0.4	-

Table 14: **Coverage Recall** - Average number of times (in %) a tuple in Target KB is found in Source KB. †This value is greater than 100 because multiple tuples in ATOMIC_{20}^{20} can map to the same tuple in ATOMIC.

Human Evaluation (Crowdsource, $15 per hour)

KG	Model	Accept	Reject	No Judgm.
ATOMIC$^{20}_{20}$	GPT2-XL	36.6	62.5	0.9
	GPT-3	73.0	24.6	2.5
	COMET(GPT2-XL)	72.5	26.6	0.9
	COMET(BART)	**84.5**	**13.8**	1.7
ATOMIC	GPT2-XL	38.3	61.2	0.4
	COMET(GPT2-XL)	64.1	34.7	1.2
	COMET(BART)	**83.1**	**15.3**	1.6
CONCEPTNET	GPT2-XL	50.3	42.1	7.7
	COMET(GPT2-XL)	74.5	19.0	6.4
	COMET(BART)	**75.5**	**17.9**	6.6
TRANSOMCS	GPT2-XL	**28.7**	**53.5**	17.8
	COMET(GPT2-XL)	26.9	60.9	12.2
	COMET(BART)	23.8	65.9	10.3

Table 15: Human evaluation of generation accuracy (%). Each model uses greedy decoding to generate the *tail* of 5K randomly-sampled test prefixes (*head*, *relation*) from each knowledge graph. GPT2-XL, GPT-3 and BART have 1.5B, 175B and 440M parameters, respectively.

Automatic Evaluation

		Bleu-1	Bleu-2	Bleu-3	Bleu-4	METEOR	ROUGE-L	CIDEr	BERT Score
ATOMIC_{20}^{20}	COMET(GPT2-XL)	0.401	0.247	0.168	0.123	0.288	0.473	0.620	0.632
	COMET(BART)	**0.462**	**0.280**	**0.182**	**0.124**	**0.325**	**0.486**	**0.632**	**0.636**
ATOMIC	COMET(GPT2-XL)	0.429	0.300	0.225	**0.187**	0.297	0.527	0.754	0.638
	COMET(BART)	**0.521**	**0.330**	0.225	0.164	**0.351**	**0.552**	**0.766**	**0.650**
CONCEPTNET	COMET(GPT2-XL)	0.152	**0.115**	**0.092**	**0.080**	**0.131**	**0.193**	**0.421**	**0.552**
	COMET(BART)	**0.169**	0.108	0.069	0.046	0.127	0.180	0.350	0.532
TRANSOMCS	COMET(GPT2-XL)	0.298	0.000	0.000	0.000	0.179	0.300	0.249	0.677
	COMET(BART)	**0.351**	**0.216**	**0.004**	0.000	**0.201**	**0.352**	**0.298**	**0.681**

Table 16: Automated metrics for the quality of the *tail* generations for the knowledge models COMET(GPT2-XL) and COMET(BART). Each approach uses greedy decoding for all test prefixes for each KG. Similar results were obtained on the 5K sampled prefixes that were randomly selected for the human evaluation.

Discussion I

Do pretrained language models already encode commonsense knowledge?

The COMET training paradigm proposed by can perhaps be viewed less as a means of learning *knowledge* from KGs, and more as a method of learning an *interface* for language models to hypothesize encoded knowledge through language generation. We look forward to future work in this space that attempts to disentangle these two ideas.

Discussion II

What considerations should be made when designing commonsense knowledge resources?

Because certain types of knowledge are already encoded and expressible by pretrained language models, CSKG designers should focus on collecting examples and categories of knowledge that are less likely to be known by language models. For example, of the 378 test tuples evaluated by the GPT2-XL zero-shot model that contained the `HinderedBy` relation, only 1.3% were deemed plausible by human raters – jumping to 85% plausibility for COMET(BART) – pointing to an advantage in constructing ATOMIC$^{20}_{20}$ with this relationship in mind or per-relation accuracy.

Contents

Hashtags, Emotions, and Comments

Hashtags, Emotions, and Comments: A Large-Scale Dataset to Understand Fine-Grained Social Emotions to Online Topics

Keyang Ding Jing Li* Yuji Zhang
Department of Computing, The Hong Kong Polytechnic University, HKSAR, China
{keyang.ding,yu-ji.zhang}@connect.polyu.hk
jing-amelia.li@polyu.edu.hk

Figure 18: Paper Info [8]

[8] Keyang Ding, Jing Li and Yuji Zhang. Hashtags, Emotions, and Comments: A Large-Scale Dataset to Understand Fine-Grained Social Emotions to Online Topics. EMNLP2020.

Motivations

- Most of the related work focus on the feelings from writers and the existing studies concerning reader emotions mostly tackle well-written texts, such as news reports.
- Limited work has been done to characterize collective feelings from the public (henceforth **social emotions**) to an online topic described with fragmented and colloquial social media language.
- Where some previous efforts gather viewpoints from limited readers through user replies manual annotations, we focus on social emotions reflecting aggregated feelings from large amount of people.

Task Definition

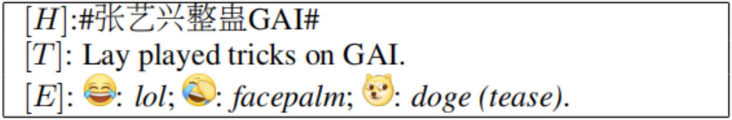


Figure 19: A Weibo hashtag and its resulting social emotions.

Data Collection

Our dataset is built based on a Weibo emotion vote, where it provides users to vote for an emoji from a total of 24 emojis in the form of a questionnaire to represent their feelings to a trending hashtag.

Data Collection

First,we tracked the trending hashtags following the everyday Weibo topic summary list from Apr to May 2020.

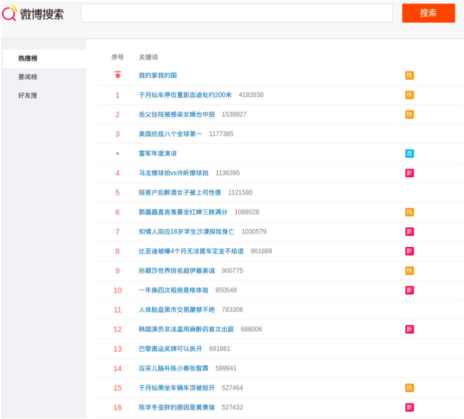


Figure 20: Illustration of the summary list.

Data Collection

Then, we searched and parsed their emotion vote webpage via querying the hashtag in HTTP requests with the selenium package.

Figure 21: Illustration of the vote webpage.

Data Collection

- Next, the crawled pages were parsed and analyzed using lxml package to gather the topics' emotion voting results. At last, hashtags with less than 100 voters were removed to filter out biased results.
- As Weibo only keeps emotions gaining the top three votes, we will hence focus on the top three emotions in the following discussions. These emotions were selected by over 83% voters on average and can still reflect feelings from the majority.
- Furthermore, to access the contexts of hashtags, we collected some user comments involved in a hashtag's discussion.

Data Statistics

Dataset	Size	Len	Voters	Emos
Zhou et al. (2018)	5,586	702.4	157	6
Bostan et al. (2019)	5,000	11.3	331	8
Our dataset	13,766	5.4	3,250	24

Figure 22: Statistics: our data vs. prior resource. Size and emos are the number of instances and emotion types. Len and voters are the average number of words (after Chinese word segmentation) and the involved voters per instance.

Data Statistics

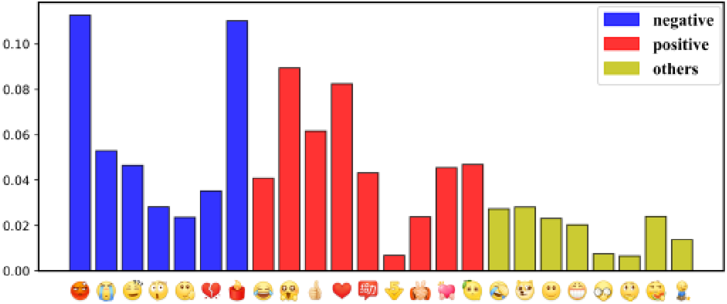
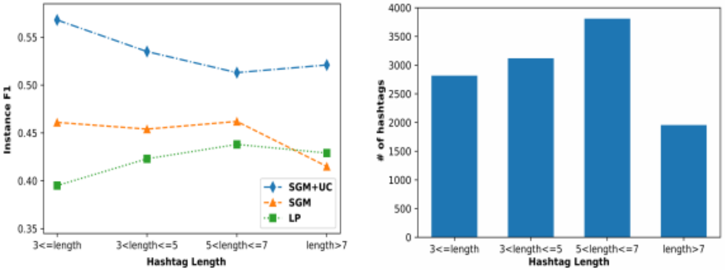


Figure 23: User preferences over varying emotions.

Result Analyses



(a) Length vs. Instance F1 (b) Length vs. Hashtag Count

Figure 24: Instance F1 (left y-axis) in prediction and training hashtag number (right y-axis) over hashtag length (Chinese word count shown in x-axis).

Thanks for Listening.

NUSTM
http://www.nustm.cn/member/rxia/index-cn.html
https://github.com/NUSTM

References I

[1] X. Lu, P. West, R. Zellers, R. Le Bras, C. Bhagavatula, and Y. Choi, "NeuroLogic Decoding: (Un)supervised Neural Text Generation with Predicate Logic Constraints," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (Online), pp. 4288–4299, Association for Computational Linguistics, June 2021.

[2] B. Y. Lin, W. Zhou, M. Shen, P. Zhou, C. Bhagavatula, Y. Choi, and X. Ren, "CommonGen: A constrained text generation challenge for generative commonsense reasoning," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, (Online), pp. 1823–1840, Association for Computational Linguistics, Nov. 2020.

[3] B. Zhou, K. Richardson, Q. Ning, T. Khot, A. Sabharwal, and D. Roth, "Temporal Reasoning on Implicit Events from Distant Supervision," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (Online), pp. 1361–1371, Association for Computational Linguistics, June 2021.

References II

[4] N. Mostafazadeh, N. Chambers, X. He, D. Parikh, D. Batra, L. Vanderwende, P. Kohli, and J. Allen, "A Corpus and Evaluation Framework for Deeper Understanding of Commonsense Stories," in *NAACL*, 2016.

[5] B. Zhou, Q. Ning, D. Khashabi, and D. Roth, "Temporal Common Sense Acquisition with Minimal Supervision," in *ACL*, 2020.

[6] H. Wang, M. Chen, H. Zhang, and D. Roth, "Joint constrained learning for event-event relation extraction," in *EMNLP*, 2020.

[7] J. D. Hwang, C. Bhagavatula, R. L. Bras, J. Da, K. Sakaguchi, A. Bosselut, and Y. Choi, "(Comet-) Atomic 2020: On Symbolic and Neural Commonsense Knowledge Graphs," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 6384–6392, May 2021. Number: 7.

[8] K. Ding, J. Li, and Y. Zhang, "Hashtags, Emotions, and Comments: A Large-Scale Dataset to Understand Fine-Grained Social Emotions to Online Topics," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Online), pp. 1376–1382, Association for Computational Linguistics, Nov. 2020.