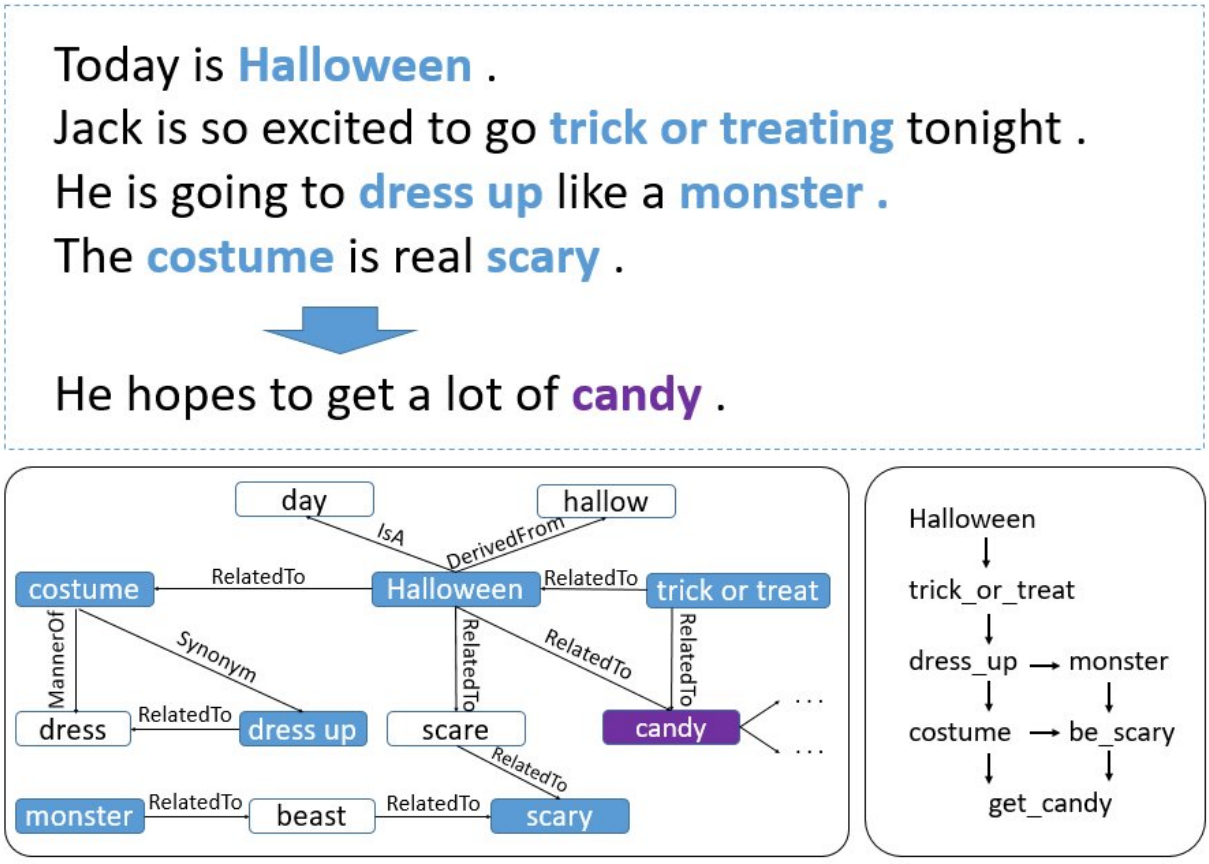
Story ending generation using incremental encoding

September 13 2018, by Ingrid Fadelli



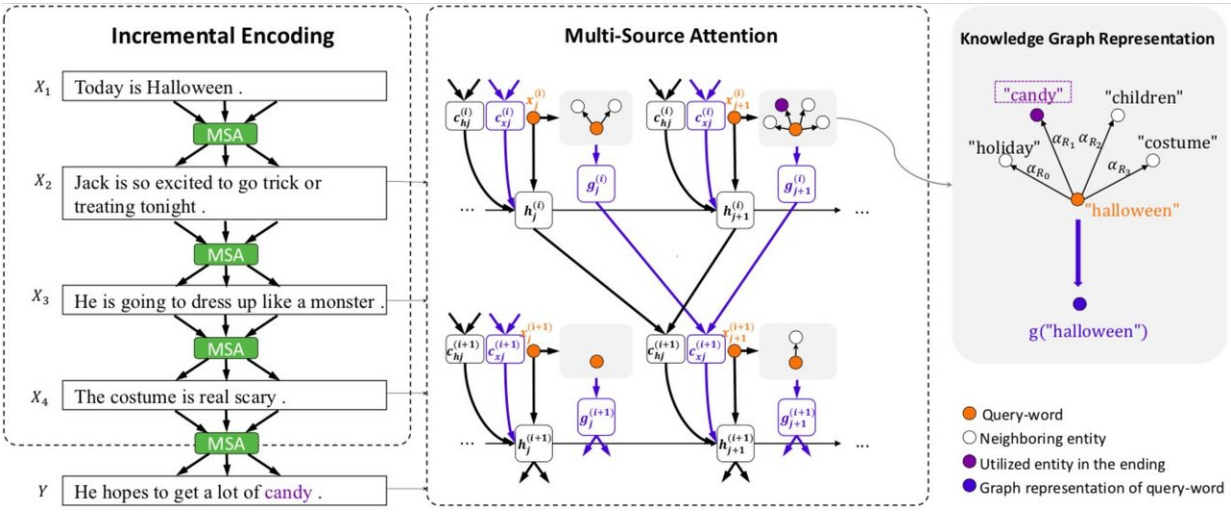
A story example. Words in bold are events and entities. The bottom-left graph is retrieved from ConceptNet and the bottom-right graph represents how events and entities form the context clue. Credit: Guan, Wang & Huang

Researchers at the AI Lab of Tsinghua University have recently developed an incremental encoder-based model that can generate story endings. An incremental encoder is a type of encoding compression algorithm that is often used to compress sorted data, such as lists of words or sentences.

The new model, outlined in a paper pre-published on arXiv, employs an incremental encoding scheme with multi-source attention to process context clues spanning throughout a story, generating a suitable ending.

Initially, the researchers were interested in the Story Cloze Test (SCT), in which a system chooses a correct ending for a story out of two available possibilities. Previous research focused on this particular test to develop story-ending generation tools, but the recent study takes this idea one step further.

"We thought, why not develop a model that can generate an ending on its own? So we came up with the story-ending generation task," Yansen Wang, one of the researchers who carried out the study told TechXplore. "However, soon, we found that generating a reasonable story ending is a much more challenging task than the original one because it requires to capture logic and causality information that may span throughout multiple sentences of a story context. Utilizing of common sense is also necessary in this task, which is not as important if two possible endings are given."



Model overview. The model is equipped with incremental encoding (IE) and multi-source attention (MSA) (including graph attention to represent knowledge graphs). MSA produces state context vectors (c (i) hj) by attending to the hidden states (h (i) j), and knowledge context vectors (c (i) xj) by attending to graph vectors (g (i) j) of the preceding sentence. IE uses the state and knowledge context vectors to build context encodings incrementally. Each graph is represented by a vector (g (i) j) using graph attention. x (i) j denotes the j-th word in sentence i. Credit: Guan, Wang & Huang

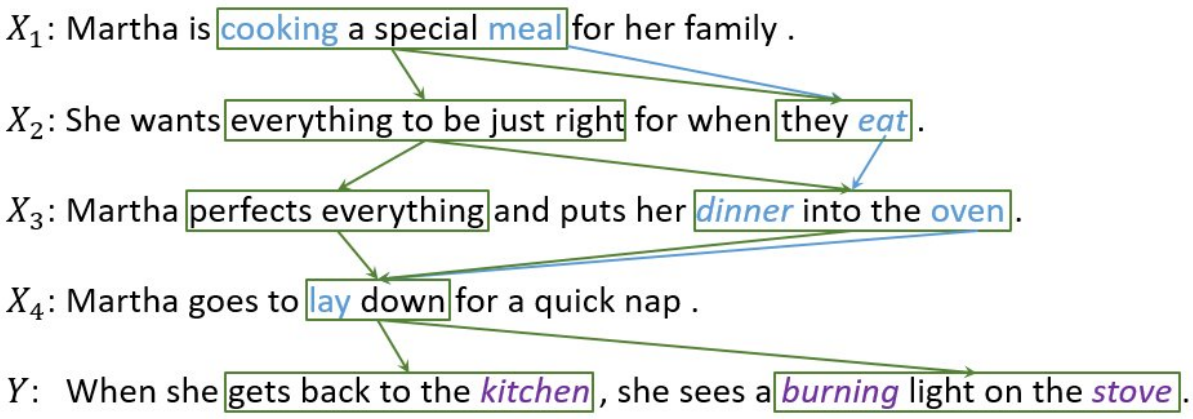
To address these two challenges, the researchers developed an incremental encoding scheme with a multi-source attention mechanism that can generate effective story endings. This system works by encoding a story's context incrementally, with its multi-source attention mechanism using both context clues and common sense knowledge.

"The incremental encoding scheme we developed can encode the previous states containing information and relationships between words incrementally," Wang said. "The multi-source attention mechanism will find and capture the chronological order or causal relationship between entities or events in adjacent sentences. To utilize commonsense

knowledge, one head of the multi-source attention will point to a logical representation of words, which contains commonsense knowledge retrieved from ConceptNet."

Designing this model proved to be a difficult and complex task, as several challenges had to be overcome to ensure that the system produced sensible endings. In fact, an effective story ending should consider several aspects of the story, fit well with its context and also make reasonable sense.

"Story ending generation requires capturing the logic and causality of information," Wang explained. "This kind of information is not point-to-point only. In most cases, it forms a more complex structure, which people call 'context clue.' We spent much time designing our model, then the incremental encoding scheme came up. The attention between sentences naturally forms a net-like structure, and the logic information passed by attention is just what we wanted."



Entity	Commonsense knowledge
cook	(cook, AtLocation, *kitchen*)
	(cook, HasLastSubevent, *eat*)
meal	(meal, AtLocation, *dinner*)
	(meal, RelatedTo, *eat*)
eat	(eat, AtLocation, *dinner*)
oven	(oven, AtLocation, *stove*)
	(oven, RelatedTo, *kitchen*)
	(oven, UsedFor, *burn*)

An example illustrating how incremental encoding builds connections between context clues. Credit: Guan, Wang & Huang

The researchers evaluated their model and compared it with other story ending generation systems. They found that it could generate far more appropriate and reasonable story endings than state-of-the-art baselines.

"When testing the model, we achieved charming results," Wang said. "In the following experiments, we also found that this scheme could pass more information, including commonsense knowledge, only if we can

represent this kind of information properly. This demonstrates the flexibility of our scheme."

The model designed by Wang and his colleagues proves just how far the latest technology can go, even in tasks that have so far been primarily completed by humans. While it has achieved highly promising results, the researchers believe that there is still great space for improvement.

"We are now trying to apply this framework on longer story corpus, since the length of the stories in SCT is not too long," said Wang. "What's more, since the incremental encoding framework can carry different kinds of information, we are trying to apply it to other kinds of tasks that involve long-term information passing, such as multi-turn conversation generation."

More information: Story Ending Generation with Incremental Encoding and Commonsense Knowledge. arXiv: 1808.10113v1 [cs.CL]. arxiv.org/abs/1808.10113