Event Semantic Knowledge in Procedural Text Understanding

Ghazaleh Kazeminejad
University of Colorado Boulder
ghazaleh.kazeminejad@colorado.edu

Martha Palmer
University of Colorado Boulder
martha.palmer@colorado.edu

Abstract

The task of entity state tracking aims to automatically analyze procedural texts – texts that describe a step-by-step process (e.g. a baking recipe). Specifically, the goal is to track various states of the entities participating in a given process. Some of the challenges for this NLP task include annotated data scarcity and annotators' reliance on commonsense knowledge to annotate implicit state information. Zhang et al. (2021) successfully incorporated commonsense entity-centric knowledge from ConceptNet into their BERT-based neurosymbolic architecture. Since English mostly encodes state change information in verbs, we attempted to test whether injecting semantic knowledge of events (retrieved from the state-of-the-art VerbNet parser) into a neural model can also improve the performance on this task. To achieve this, we adapt the methodology introduced by Zhang et al. (2021) for incorporating symbolic entity information from ConceptNet to the incorporation of VerbNet event semantics. We evaluate the performance of our model on the ProPara dataset (Mishra et al., 2018). In addition, we introduce Lexis, our purely symbolic model for entity state tracking that uses a simple set of case statements, and is informed mostly by linguistic knowledge retrieved from various computational lexical resources. Our approach is inherently domain-agnostic, and our model is explainable and achieves state-of-the-art results on the Recipes dataset (Bosselut et al., 2017).

1 Introduction

Language understanding in humans requires at least the knowledge of the semantics of events and entities. One needs to know the sequences of subevents that together make up a 'throwing' event, as well as the causal and temporal relationships between the subevents that distinguish a 'throwing' event from a 'pouring' event, or a 'running' event. Furthermore, reasoning about entities that are participating in these events requires a deep understanding of the properties of an entity. It is the distinction between such entity properties that enables us, for example, to distinguish between 'throwing a ball' vs. 'throwing a Molotov cocktail'. In contrast to humans, many high-performing NLP models do not depend on explicit knowledge of events and entities to process natural language; rather, they rely on the surface forms and patterns of word co-occurances in colossal amounts of language data to learn the mechanics of language as well as the interpretation of linguistic forms. Since human knowledge and reasoning capabilities benefit from knowledge of events and entities, we suggest that a neural model may also benefit from such explicit symbolic knowledge. This requires successful incorporation of such symbolic knowledge into a subsymbolic system.

Explicit semantic knowledge, such as entity knowledge extracted from ontologies, has often been used in the field of natural language grounding, where the connection between natural language and the physical world is sought (Bisk et al., 2020). There are yet other NLP tasks that are likely to benefit from explicit semantic knowledge as well, such as tasks focusing on machine comprehension of how things work (e.g. how plants make food), or how a certain physical result is achieved (e.g. how to make pizza using some ingredients). The NLP task that focuses on the machine reading comprehension of texts describing processes is called *Procedural Text Understanding* (Huang et al., 2021; Tandon et al., 2019; Mishra et al., 2019). One of the subtasks in this field is *Entity State Tracking* (Mishra et al., 2018; Bosselut et al., 2017; Faghihi and Kordjamshidi, 2021; Amini et al., 2020; Gupta and Durrett, 2019), formally defined as: Given a paragraph P that describes a process, and an entity e that is one of the participants in that process, did the state of e change during the process? If so, what was the type of change that occurred to e

*Proceedings of the The 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)*, pages 388–398
July 13-14, 2023 ©2023 Association for Computational Linguistics

(usually to be chosen among a desired set of types of state change)? When did the change happen (i.e. at which time step during the process)? And finally, what was the locus of change (i.e. the location of *e* before and after the change) (Mishra et al., 2018)?

There are two main challenges in solving this problem. First, the size of annotated data for this task is usually small since achieving reasonable inter-annotator agreement for the task is hard, making it expensive and time-consuming. Second, when facing implicit information, annotators frequently resort to commonsense knowledge – knowledge that state-of-the-art NLP models are not explicitly aware of. Existing models for this challenging problem use some flavor of learning-based approaches to NLP (see Section 2). One of the existing approaches that is closest in theory to ours is KOALA (Zhang et al., 2021) – a neurosymbolic model encoding entity-centric knowledge into a neural network that is used to track entity states and locations during a process. We re-implemented this model and adopted it as our baseline. One of our contributions in this work is offering a method for encoding symbolic event semantic knowledge into a neural model. In practice, we are proposing an approach to expose a neural model to sequences of latent universal concepts composing an event, allowing the network to learn from the spelled out event semantics as well as the surface forms of the events realized mostly as verbs.

In addition to our neurosymbolic model (SKIP: **S**emantic **K**nowledge **I**n **P**rocedural text understanding), we have also developed a purely symbolic model[1] called LEXIS. Error analysis and ablation tests on this model demonstrate other sources of external knowledge that show promise for inclusion in a neural model in future work. In addition, we show that our theory and approach are dataset- and domain-independent, and can be used in any NLP task where knowledge of event semantics plays a major role for humans to achieve the goal of the task. We will also briefly illustrate our explanation module for LEXIS.

We evaluated SKIP on the ProPara dataset (Mishra et al., 2018), and LEXIS on both the ProPara and Recipes (Bosselut et al., 2017) datasets[2]. LEXIS achieved a new state-of-the-art performance on the Recipes dataset (70.1% F1, improving over the existing state-of-the-art model by 11.7%), and SKIP performed better than our adopted neurosymbolic baseline model, (71.8% F1, improving over the re-implemented baseline model by 4.1%)[3].

Our contributions are two-fold: (1) We adapt the methodology introduced by Zhang et al. (2021) for incorporating symbolic entity-centric knowledge to the incorporation of VerbNet event semantics. We extract and encode event semantic knowledge for injection into a neural network, and present SKIP, an end-to-end neurosymbolic model developed using this method, in conjunction with data augmentation and transfer learning techniques. (2) We present a general knowledge-based approach to text understanding using existing NLP resources, and present LEXIS, a purely symbolic model we developed for entity state tracking that achieves a new state-of-the-art on the Recipes dataset, with an architecture that is adaptable to different genres of natural language text, and is explainable[4].

2 Related Work

This work is inspired by the concept of event semantics and event structure offered by the Generative Lexicon theory, in efforts such as Pustejovsky and Moszkowicz (2011), Mani and Pustejovsky (2012), and Brown et al. (2022), where event structure is enriched to encode and dynamically track object attributes that are modified during an event. The idea is that a complex event can be decomposed into simpler ordered subevents that explicitly label the transitions between entity states.

With regard to Entity State Tracking, most recent existing models mainly rely on large language models (Amini et al., 2020; Faghihi and Kordjamshidi, 2021; Zhang et al., 2021), while earlier models (prior to 2020) rely on neural (Gupta and Durrett, 2019; Das et al., 2018; Du et al., 2019) or learning-based approaches (Ribeiro et al., 2019). The only existing neurosymbolic model (to our knowledge) is KOALA (Zhang et al., 2021), which retrieves

[1] Here, purely symbolic is used as opposed to sub-symbolic models that learn by example. (Garcez et al., 2019; Hamilton et al., 2022)

[2] The reason we did not evaluate SKIP on the Recipes dataset was that we only exposed SKIP to the knowledge extracted directly from the VerbNet parser, which does not shed light on the types of state change the model is expected to predict in the Recipes dataset. We have access to such knowledge and will perform this evaluation in future work.

[3] Our code is publicly available at https://github.com/ghamzak/SKIP (for SKIP) and https://github.com/ghamzak/Lexis (for LEXIS).

[4] Disclaimer: the models developed and introduced in this work are for research purposes only and are not to be trusted in real-world applications.

informative knowledge triples from ConceptNet (Speer et al., 2017) and performs knowledge-aware reasoning while tracking the entities. To compensate for data scarcity, they perform (raw) data augmentation by automatically retrieving the top 50 Wikipedia articles closest in content and writing style to the raw paragraphs in the ProPara dataset (using tf-idf). This augmented corpus of raw procedural texts is then used to perform transfer learning, fine-tuning a BERT encoder in two stages, first on raw procedural texts collected from Wikipedia, and then further fine-tuning it on the raw text from the dataset. The whole model follows a multi-stage training schema (more details in section 3.1).

The main difference between KOALA and SKIP is the type of external symbolic knowledge introduced to the model. Whereas KOALA only leverages entity-centric knowledge, we introduce event semantic knowledge based on the Generative Lexicon theory and its implementation in the VerbNet lexical resource (Schuler, 2005; Brown et al., 2018, 2019). This allows the model to have access to direct and explicit knowledge about entity state transitions for all the participants in an event, the roles of each participant in the event, as well as causal relationships and temporal links between subevents (Brown et al., 2022).

On the Recipes dataset, Zhang et al. (2021) evaluate only for location prediction, because location change is one of the state change types needed by ProPara as well. To enable prediction for the rest of the state change types required by the Recipes dataset , a previously lacking knowledge resource has recently become available (Kazeminejad et al., 2022) which explicitly provides the lexical semantic components indicating state changes such as changes in temperature or form, giving our symbolic model (LEXIS) an edge over other competing models[5].

3 Methodology

Following Zhang et al. (2021), we develop SKIP by neural encoding of symbolic knowledge and allowing the model to selectively pay more attention to knowledge that is conducive to more accurate predictions. As mentioned in section 1, our main contribution is proposing a way to make a neural model utilize event semantic knowledge in its predictions, and use the obtained neurosymbolic model for downstream NLP tasks where knowledge of event semantics tends to be beneficial according to linguistic theory.

In order to obtain logical representations of subevent semantics as well as temporal and causal relations between the subevents for encoding into our neural model, we rely on VerbNet – a large English verb lexicon which expands event semantics into sequences of subevents. To automate this process, we use the state-of-the-art VerbNet semantic parser (Gung, 2020; Gung and Palmer, 2021) and obtain the symbolic logical representations for individual sentences corresponding to the steps in each process. These logical representations, illustrated in Table 1, are the horsepower of our approach.

¬Degradation_Material_Integrity(e_1, The sediment)
¬Has_Physical_Form(e_1, The sediment, V_Final_State)
Degradation_Material_Integrity(e_2, The sediment)
Has_Physical_Form(e_2, The sediment, V_Final_State)

Table 1: Logical representations generated by the VerbNet parser for the sentence "The sediment breaks down." The span 'breaks down' is identified as the verb, and verb sense disambiguation classifies it as belonging to the VerbNet class `break-45.1`.

In VerbNet, verbs are classified into different classes based on similarities in their syntactic and semantic behavior. For example, all verbs belonging to the `break-45.1` class (Table 1) indicate some sort of physical change of state that leads to the breaking into parts of a `Patient` argument. Different syntactic frames may incorporate more information such as the causal agent of the event, or the instrument used by the causal agent to achieve the result. The set of semantic predicates adopted by the VerbNet lexicon (such as Degradation_Material_Integrity or Has_Physical_Form in Table 1) are universal eventive concepts that lead human cognitive contsrual of events, and are based on cognitive linguistic theories such as Force Dynamics (Talmy, 1988; Croft, 2015, 2017; De Mulder, 2021). More details on event semantic knowledge extraction will follow in 3.2.

The VerbNet-extracted event semantic knowledge is then translated into natural language so that it is neurally encodable. We choose this method of encoding over direct encoding of the logical representations, because LLMs such as BERT are already familiar with the structure of natural language, and we want to hone this existing power instead of introducing a whole new representation

[5] Again, we have not yet exposed SKIP to this knowledge resource, but this will be done in future work.

system which might be harder to learn, especially given the small size of the dataset. In order to acquaint a vanilla text encoder with the language translated from the event semantic logical representations, we fine-tune a BERT encoder (Kenton and Toutanova, 2019) on the translated knowledge extracted for the training data. This will be explained in more detail in 3.3.

3.1 Neurosymbolic Framework

The base architecture of SKIP (shown in Figure 1) is developed on top of KOALA (Zhang et al., 2021), which we adopted as our baseline model. As explained in Section 2, our major point of departure is the introduction of event semantics to the model, and, for that matter, a method to obtain such representations for free. Before attending to our differences, however, we present a brief overview of our similarities with the KOALA framework.

Following KOALA, we perform multi-stage training to obtain our text and knowledge encoders. To get contextualized embeddings for raw input paragraphs, we train a text encoder specialized in understanding procedural texts by fine-tuning a vanilla BERT encoder on a tf-idf-retrieved corpus of raw procedural texts from Wikipedia, and then on the raw paragraphs from the ProPara dataset. SKIP duplication of KOALA ends here. To obtain a knowledge encoder, since our sources of external knowledge are different, our knowledge extraction methods are different as well (see 3.2). Naturally, our post-knowledge-extraction translation rules are also different, with the event semantic translation rules being arguably more complex, the first reason being that the entities are always represented in triples, while events could be intransitive, transitive, or ditransitive, each requiring a different type of translation.

After knowledge translation, a knowledge encoder is obtained by training a vanilla BERT encoder on the knowledge translations (more details to follow in 3.3), learning to make sense of VerbNet-style event semantics, as well as ConceptNet-style entity semantics (see Figure 2). In the final training stage, SKIP (like KOALA) leverages an encoder-decoder architecture, and performs *state tracking* and *location prediction* in two separate yet parallel subtasks (as shown in Figure 1). The training objective of the model is to jointly optimize state and location prediction, as well as knowledge selection, which is attending to and selecting the best knowledge pieces that are instrumental in state and location prediction.

As shown in Figure 1, the state tracking module is endowed with a knowledge injector (see 3.4 for more details), a bi-LSTM state decoder, and a conditional random field (CRF) layer since we are performing multi-class classification for multiple target state change types. For location prediction, we use the same architecture except for the CRF layer which is changed to a linear classifier, because the model is learning to predict only one location for a given entity at a given time step in a given paragraph. Of course the learned weights and the knowledge triples selected by the attention module will be different from those in the state decoder, because the attention will need to attend to different predictor variables for state tracking and location prediction.

In the location prediction module, given that there are M location candidates for paragraph P (all nominal phrases and words extracted from P using a POS-tagger), the location decoder is executed M times, and the linear classification layer outputs a score for each location candidate at each time step t based on the decoder's hidden states. Using a Softmax function, the probability distribution for each location candidate for entity e at time-step t in paragraph P is obtained, and a loss function is used to train the optimal model for location prediction.

3.2 Event Semantic Knowledge Extraction

This paper describes the extraction and incorporation of event semantic knowledge into our neural architecture. For entity-centric knowledge extraction from ConceptNet, we simply follow Zhang et al. (2021): for each target entity, we find ConceptNet nodes representing the concept using exact string matching and fuzzy matching, finding the most similar nodes based on embedding distance. The extracted knowledge triples, which are two entities and the relation between them, are then translated into natural language using handcrafted rules that translate the relations, enabling fine-tuning for developing the knowledge encoder.

For SKIP, we selected a subset of VerbNet semantic predicates that are indicative of the types of state changes of interest for the ProPara dataset: `Move`, `Create`, and `Destroy`. It is imperative to note that selected subsets can change to match the requirements of the task at hand. For each sentence X_t in paragraph P and for entity e, the

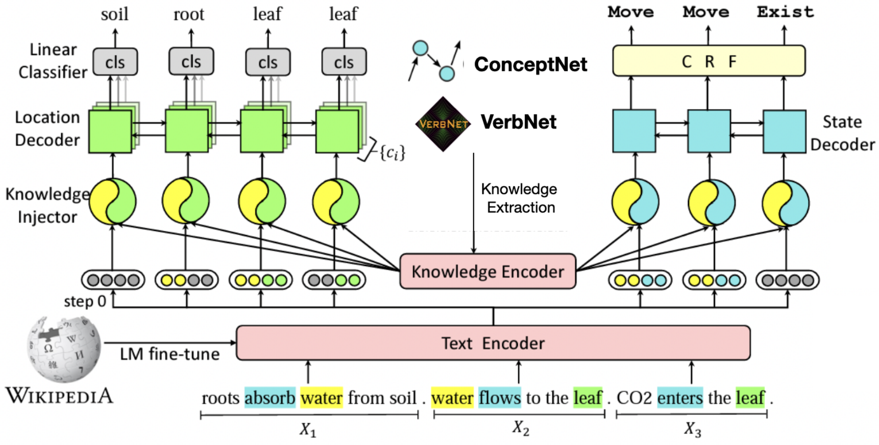


Figure 1: An overview of our model, adopted from Zhang et al. (2021). Compared to the baseline model, the sources of symbolic knowledge have been updated to include event semantics from VerbNet. Note that in the location prediction module (on the left), the whole module is applied to each entity separately and in parallel at each time step. While the text encoder is obtained by fine-tuning a BERT encoder on raw procedural texts from Wikipedia and the ProPara dataset, the knowledge encoder (see Figure 2) is obtained by fine-tuning a BERT encoder on various combinations of knowledge: knowledge from VerbNet only, from ConceptNet only, and from both.

subevents	translation
$\neg has\_location(A, B)$ $has\_location(A, B)$	A moves towards destination B
$be(A)$ $\neg be(A)$	A is destroyed
$\neg be(A)$ $be(A)$	A is created

Table 2: Sample translation rules for extracted informative subevents from VerbNet.

model reads all the generated subevents in order, and keeps those that satisfy the following two conditions: (1) the VerbNet semantic predicate is a member of the hand-selected subset of VerbNet predicates[6]; and (2) one of the arguments in the subevent has an overlap in surface form with the entity *e*. Finally, the retained subevents are translated into natural language using a set of handcrafted translation rules, such that the translation exposes the type of state change undergone by entity *e* at time step *t*. Table 2 has one example for each of the state change types.[7]

3.3 Event Semantic Knowledge Encoding

As shown in Figure 2, after extracting symbolic event semantic knowledge from VerbNet, we fine-tune a BERT encoder on the extracted knowledge with the aim of familiarizing the BERT encoder with the vocabulary and style of translations of knowledge statements. Subevents have important structural information which we preserve in our fine-tuning stage by separating the argument spans and the translation of the chosen semantic predicates. We use BERT special tokens for token-level separation [SEP], and begin the translated sentence by the BERT [CLS] special token to mark sentence-level detachment. For example, for the sentence 'The sound waves hit an object', the first argument is a `Theme` corresponding with the span 'The sound waves', and the second one is a `Goal` corresponding with the span 'an object'. Since the subevents in the first row in Table 2 apply to this sentence , the translated sentence with preserved structure will be *[CLS] The sound waves [SEP] moves towards destination [SEP] an object [SEP].* For fine-tuning, we modify the conventional masked language modeling (MLM) objective to fit the structural features of the extracted event semantic knowledge from VerbNet (Figure 3).

Since BERT has a bi-directional architecture, we iteratively mask out tokens and ask the encoder to predict the masked tokens given the unmasked tokens (see Figure 3). This allows the BERT encoder to better understand the relationships between different entities (realized as arguments) and between entities and events (translated into a sequence of

[6] For a complete list of these selected VerbNet predicates, see Appendix A in Kazeminejad (2023)

[7] For the complete list of translation rules see Appendix E in Kazeminejad (2023).

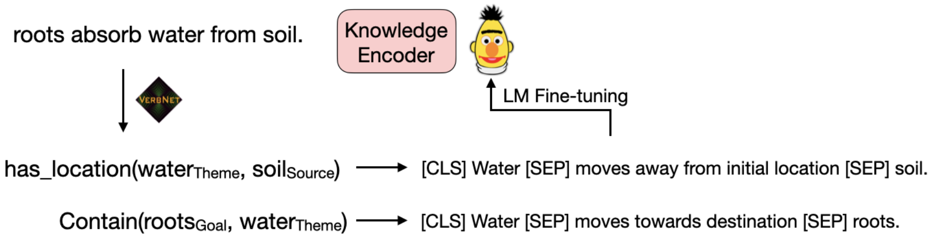

roots absorb water from soil.
Knowledge Encoder
VERBNET
LM Fine-tuning
has_location(water_Theme, soil_Source)
[CLS] Water [SEP] moves away from initial location [SEP] soil.
Contain(roots_Goal, water_Theme)
[CLS] Water [SEP] moves towards destination [SEP] roots.

Figure 2: Developing a knowledge encoder model by fine-tuning a BERT encoder on event semantic knowledge extracted from VerbNet subevents

tokens with explicit state change information). Following the empirical results obtained by Zhang et al. (2021), if the arguments are multi-word, we mask 50% of the argument tokens at a time to make sure the model is trainable. For the translation of the semantic predicate, we mask out all the tokens at once, because the set of semantic predicates in VerbNet is a closed one, and we want the model to learn the meaning of the predicate at once and as a whole. Such fine-tuning enables the encoder to learn to model the structural information conveyed in the retained subevents.

3.4 Attentive Knowledge Infusion

Having obtained the knowledge encoder, in the final training stage, the contextualized representations of the extracted (and translated) symbolic knowledge from both VerbNet and ConceptNet are calculated by mean pooling over the knowledge encoder outputs for all tokens.

Even though we have tried to keep only the informative subevents, not all of them may end up being useful in guiding the model to predict correct labels. To enable the model to select the most relevant knowledge, the knowledge injector module injects encoded knowledge into the model before each decoder as a query to attend to the encoded knowledge, helping the model attend to knowledge relevant to the context paragraph. Each decoder is equipped with an input gate to select information from the original input and the injected knowledge. Zhang et al. (2021) empirically found out that such gate integration performs better than simply concatenating the encoded text and knowledge. The training objective is to maximize the attention weights of all "relevant" triples. By the end of training and during inference, the model is expected to better identify the relevance between knowledge and prediction targets. Finally, the overall loss function is computed as the weighted sum of the loss functions for the three sub-tasks: state tracking, location prediction, and relevant knowledge selection.

4 Experiments

We evaluate SKIP on the ProPara dataset (Mishra et al., 2018), which is an entity state tracking dataset developed by AI2, containing 488 human-authored paragraphs describing scientific processes, with an 80/10/10 data split. While state change types (`Move, Create`, and `Destroy`) were expertly annotated, entity location annotation was crowed-sourced, resulting in lower quality and consistency. We perform document-level evaluation on ProPara, using the official evaluation code [8].

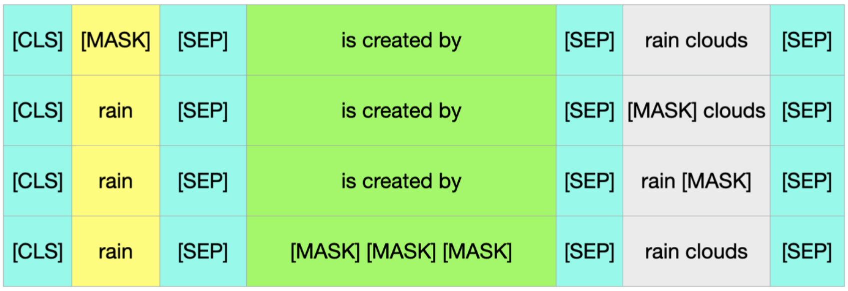
In re-implementing the baseline model (KOALA), we only changed the batch size, downsizing from 32 to 16 due to hardware limitations[9]. KOALA's reported results along with our re-implementation results are demonstrated in the first two rows of Table 3.

The whole model contains 235M parameters including 2 BERT encoders. In LM fine-tuning, we used the uncased $BERT_{BASE}$ model, and manually tuned hyper-parameters, setting the batch size to 16 and learning rate to 5×10^{-5}. While we used the same text encoder developed by Zhang et al. (2021), our knowledge encoder was different. It was trained for 2 epochs on external knowledge. In the final training stage, we used a batch size of 10 and a learning rate of 3×10^{-5} on the Adam optimizer. The hidden size of the LSTMs was set to 256 and the dropout rate to 0.4. We performed early stopping with an impatience of 20 epochs, by evaluating changes in model accuracy over the dev set (~1.5 GPU hours). We selected the best checkpoint in prediction accuracy on the dev set.

As shown in Table 3, our three main experimental settings included changes to the source

[8] https://github.com/allenai/aristo-leaderboard/tree/master/propara/evaluator

[9] TITAN Xp GPU with 12 GB Memory



[CLS]	[MASK]	[SEP]	is created by	[SEP]	rain clouds	[SEP]
[CLS]	rain	[SEP]	is created by	[SEP]	[MASK] clouds	[SEP]
[CLS]	rain	[SEP]	is created by	[SEP]	rain [MASK]	[SEP]
[CLS]	rain	[SEP]	[MASK] [MASK] [MASK]	[SEP]	rain clouds	[SEP]

Figure 3: Translation and masking of one VerbNet-extracted subevent for the entity 'rain', indicating that "rain is created by rain clouds".

Model	Precision	Recall	F1
KOALA reported results	**77.7%**	64.4%	70.4%
KOALA reimplementation (baseline)	73.0%	63.1%	67.7%
SKIP – fine-tuned on VN only	76.5%	**67.6%**	**71.8%**
– fine-tuned on both VN and CN	72.0%	61.4%	66.3%
– fine-tuned on CN only	74.1%	63.3%	68.3%

Table 3: The top two rows show the reported and re-implementation results of the KOALA model. The bottom three rows demonstrate the results of our three main experimental settings, where the knowledge encoder used in model training is obtained by fine-tuning on VN (VerbNet) only, CN (ConceptNet) only, or both. These are all evaluations on the ProPara dataset.

of semantic knowledge in developing our knowledge encoder. SKIP performed better compared to the baseline in the experimental setting where the source of knowledge for developing the knowledge encoder was only VerbNet event semantics. Note that we use both entity and event knowledge during the final training stage, and it is only the changes in knowledge source for LM fine-tuning to obtain different knowledge encoders that leads to the best experimental results.

5 Discussion

Our experimental results were interesting in two ways. First, the fact that LM fine-tuning on both VerbNet and ConceptNet lowers the performance compared to fine-tuning on only one knowledge source could be an indication that, given the size of the data, two different sources of knowledge seems to confuse the knowledge encoder more than helping it. Secondly, comparing fine-tuning on VerbNet only vs. ConceptNet only, the former proved to be more effective. This might indicate that knowledge of event semantics may better help the model track entity states and locations during a process, just as we had initially hypothesized based on lexical semantic theories. While entity-centric knowledge may give the model a better understanding of entities and their properties, such as their typical location, state changes are eventive concepts and often lexically encoded in verbs. Since VerbNet provides explicit labels for transitions between entity states, a successful VerbNet parse ensures explicit symbolic knowledge which clarifies the types of state change lexically encoded in verbs.

5.1 Error Analysis

An error analysis on the test set showed that 52.49% of the state change type misclassifications were in fact correct model predictions and incorrect gold annotations, with a further 6.69% examples where both the gold and predicted labels were incorrect. To illustrate, given the two subsequent time steps 'Animals eat plants.' and 'Animals make waste.', for the target entity 'plants', the gold labels include two `Move` events, one at each time step: first from an unknown location to 'animal', and then from 'animal' to an unknown location. In contrast, SKIP predicts a `Destroy` event at the end of the first time-step. Arguably, an entity that is eaten and converted to waste is destroyed, because it has lost its physical integrity, such as a glass that breaks. What returns to nature is not a plant anymore, but waste. This assumption is also confirmed elsewhere in the data. For example, in the sentence 'They absorb nitrates from the soil into their roots.', the gold label for the entity 'nitrates' is `Destroy`. This both suggests inconsistency in human annotation, and the accuracy of SKIP.

Overall, the error analysis demonstrates that the annotation task for entity state tracking is quite complex and challenging, and obtaining acceptable inter-annotator agreement is hard. A knowledge-aware model such as SKIP could be quite beneficial for annotation quality control.

5.2 Purely Symbolic Entity State Tracking Model

LEXIS was designed based on an approach to simulate the cognitive construal of events by humans. This approach is inherently domain-independent and can be readily adapted to other natural language domains or NLP tasks. As an example model founded on this approach, LEXIS relies on the same informative subevents and semantic features from VerbNet that benefited SKIP. In addition, PropBank SRL is used as a backoff for gaps in VerbNet parses. For more details on an earlier version of LEXIS, see (Kazeminejad et al., 2021a)[10].

In addition to the ProPara dataset, we also evaluated LEXIS on the Recipes dataset, which contains 866 human-annotated recipes, with an 80/10/10 data split, with each recipe containing an average of 8.8 sentences. Recipes state change types include changes in `composition`, `cookedness`, `temperature`, `rotation`, `shape`, `cleanliness`, and `accessibility`, as well as `location`. Apparently, there is very little overlap with ProPara state change types of interest. Neither are these state change types normally found in VerbNet. However, we were able to use the recently developed semantic layer added to the VerbNet lexicon that includes more fine-grained semantic features specific to each verb, hence called verb-specific features (Kazeminejad et al., 2022). For instance, the `Other_cos-45.4` class with more than 300 verb members is generally about some physical change of state occurring to a `Patient` argument. These semantic components provide details such as the physical property that is changing (e.g. temperature, speed, intensity, etc.), or the final state of the `Patient` entity (e.g. ±clean, ±open, etc) that LEXIS can use to predict state change types.

LEXIS also uses spaCy (Honnibal et al., 2020) dependency parsing and POS tagging for conjunction analysis, compound identification, extracting objects of prepositions and heads of noun phrases. ConceptNet was used to identify whether an entity is ontologically considered locative, and also to perform fuzzy search (using the spaCy large model) to find the most likely typical location if not explicitly mentioned in a given sentence. We also used fast-coref (Toshniwal et al., 2021), a high-performing generalizable domain-independent coreference resolution module, to identify co-referring entities given a paragraph, and substitute pronominal forms with their content word counterpart. Finally, we used the the logical rule of location transitivity to enable the model to update entity locations accordingly.

On the ProPara dataset, LEXIS achieves an overall F1 score of 55.6% on the test set.

P	R	F1
72.8	45.0	55.6

Table 4: LEXIS results on the ProPara dataset

On the Recipes dataset, LEXIS achieves a new state-of-the-art both in F1 score and accuracy (see Table 5).

Model	P	R	F1	Acc
Lexis	67.9	**72.4**	**70.1**	**94.6**
SGR*	**69.3**	50.5	54.8	-
KOALA	60.1	52.6	56.1	-
REAL**	55.2	52.9	54.1	-
IEN†	58.5	47.0	52.2	-
NCET†	56.5	46.4	50.9	-
NPN‡	-	-	44.64	55.05

Table 5: LEXIS evaluation results on the test set of the Recipes dataset. * (Tang et al., 2022); ** (Huang et al., 2021); † (Tang et al., 2020); ‡ (Bosselut et al., 2017).

A series of ablation tests on the ProPara dataset showed that the best model performance was achieved when all the proposed knowledge components were included in the model. In addition, following VerbNet and PropBank parses which had the greatest impact[11], the single component with the most significant impact on LEXIS results used the verb-specific features (the semantic layer recently added to VerbNet), the removal of which lowered model performance by 3.5% (F1).

In addition, an error analysis of the model showed that within the 24.5% prediction mis-

[10] However, keep in mind that the latest version of this system that is referenced here is yet to be published.

[11] Note that the first version of LEXIS (Kazeminejad et al., 2021b) only used VerbNet parses.

matches, only 7.18% were due to cross-label confusion. For the main part, the mismatches were either false negatives or false positives, with the false negatives being about two times the number of false positives. This is due to the design of the model which tends to avoid labeling if there is any ambiguity or uncertainty. In other words, the model is deterministic by design.

Regarding explainability, LEXIS contains an explanation module which traces back on the prediction path and explains every step in making decisions, including the provenance of that decision. For instance, for the sentence 'They are buried in sediment', LEXIS predicts a `Move` event for the entity 'plants', from an unknown location to 'sediment'. Here is what the explanation module generates:

The verb 'bury' is in `put-9.1-1` VerbNet class. *(provenance: VerbNet parser).*
'they' moves to 'sediment'. *(provenance: VerbNet parser).*
'they' refers to 'plants'. *(provenance: fast-coref).*
'plants' move to 'sediment'. *(provenance: substitution).*

6 Conclusions and Future Work

We presented a method to extract event semantic knowledge and encode it in neural architectures for NLP applications where event semantics theoretically promises to enhance the predictive power of the model. We showed that this method was effective in SKIP – our neurosymbolic model designed for procedural text understanding. Our error analysis demonstrated that SKIP can be relied on to perform annotation quality control. Furthermore, LEXIS, our purely symbolic entity state tracking model designed based on our domain-independent approach, achieved a new state-of-the-art on the Recipes dataset. We explained why this approach is domain-independent and can be adapted to other domains and NLP tasks.

In future work, we would like to expand our neurosymbolic model to use other sources of linguistic knowledge that proved useful in LEXIS ablation tests. It would also be interesting to assess the success of this approach in other NLU tasks, such as causal inference and textual entailment, where event semantic knowledge is again theoretically important.

Limitations

Since our methodology relies heavily on the VerbNet lexicon and parser, the inherent limitations and shortcomings of them percolate into our model as well. VerbNet classes are designed to generalize over and abstract away from some semantic aspects of verbs in order to achieve meaningful classes. Therefore, we can rely on VerbNet only when the type of semantic knowledge we intend to obtain is included in existing VerbNet semantic predicates. For example, ProPara state change types have counterparts in VerbNet semantic predicates, while the Recipes dataset state change types do not. As explained in 5.2, we resorted to verb-specific features to obtain the type of semantic knowledge needed to predict state changes for Recipes.

VerbNet's coverage imposes a second limitation. Some verbs are missing from the lexicon (e.g. 'migrate'), leading to empty parses. Some other verbs may exist in the lexicon but a certain sense of them is missing. For example, at the time of developing the VerbNet labeled data, the locative sense of the verb 'be' was missing from the lexicon, and by extension from the labeled data. In such cases, the parser assigns that verb to an alternative class with a different sense of the same verb lemma (in this case to `seem-109-1-1`).

Finally, the amount of VerbNet training data is relatively small (compared to PropBank (Kingsbury and Palmer, 2002) or AMR (Banarescu et al., 2013)), leading to misclassifications due to sparse data. All of these limitations can be improved by expanding the coverage of the VerbNet lexicon, and expanding and updating the VerbNet labeled data accordingly.

Acknowledgements

We gratefully acknowledge the support of DARPA KAIROS Program No. FA8750-19-2-1004. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA or the U.S. government.

References

Aida Amini, Antoine Bosselut, Bhavana Dalvi Mishra, Yejin Choi, and Hannaneh Hajishirzi. 2020. Procedural reading comprehension with attribute-aware context flow. In *Automated Knowledge Base Construction*.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

Antoine Bosselut, Omer Levy, Ari Holtzman, Corin Ennis, Dieter Fox, and Yejin Choi. 2017. Simulating action dynamics with neural process networks. *arXiv preprint arXiv:1711.05313*.

Susan Windisch Brown, Julia Bonn, James Gung, Annie Zaenen, James Pustejovsky, and Martha Palmer. 2019. Verbnet representations: Subevent semantics for transfer verbs. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 154–163.

Susan Windisch Brown, Julia Bonn, Ghazaleh Kazeminejad, Annie Zaenen, James Pustejovsky, and Martha Palmer. 2022. Semantic representations for nlp using verbnet and the generative lexicon. *Frontiers in artificial intelligence*, 5.

Susan Windisch Brown, James Pustejovsky, Annie Zaenen, and Martha Palmer. 2018. Integrating generative lexicon event structures into verbnet. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

William Croft. 2015. Force dynamics and directed change in event lexicalization and argument realization. In *Cognitive science perspectives on verb representation and processing*, pages 103–129. Springer.

William Croft. 2017. The structure of events and the structure of language. In *The new psychology of language*, pages 67–92. Routledge.

Rajarshi Das, Tsendsuren Munkhdalai, Xingdi Yuan, Adam Trischler, and Andrew McCallum. 2018. Building dynamic knowledge graphs from text using machine reading comprehension. *arXiv preprint arXiv:1810.05682*.

Walter De Mulder. 2021. Force dynamics. In *The Routledge Handbook of Cognitive Linguistics*, pages 228–241. Routledge.

Xinya Du, Bhavana Dalvi, Niket Tandon, Antoine Bosselut, Wen-tau Yih, Peter Clark, and Claire Cardie. 2019. Be consistent! improving procedural text comprehension using label consistency. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2347–2356.

Hossein Rajaby Faghihi and Parisa Kordjamshidi. 2021. Time-stamped language model: Teaching language models to understand the flow of events. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4560–4570.

Artur d'Avila Garcez, Marco Gori, Luis C Lamb, Luciano Serafini, Michael Spranger, and Son N Tran. 2019. Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning. *arXiv preprint arXiv:1905.06088*.

James Gung. 2020. *Abstraction, Sense Distinctions and Syntax in Neural Semantic Role Labeling*. Ph.D. thesis, University of Colorado at Boulder.

James Gung and Martha Palmer. 2021. Predicate representations and polysemy in verbnet semantic parsing. In *Proceedings of the 14th International Conference on Computational Semantics (IWCS)*, pages 51–62.

Aditya Gupta and Greg Durrett. 2019. Tracking discrete and continuous entity state for process understanding. In *Proceedings of the Third Workshop on Structured Prediction for NLP*, pages 7–12.

Kyle Hamilton, Aparna Nayak, Bojan Božić, and Luca Longo. 2022. Is neuro-symbolic ai meeting its promise in natural language processing? a structured review. *arXiv preprint arXiv:2202.12205*.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Hao Huang, Xiubo Geng, Jian Pei, Guodong Long, and Daxin Jiang. 2021. Reasoning over entity-action-location graph for procedural text understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5100–5109.

Ghazaleh Kazeminejad. 2023. *Computational Lexical Resources for Explainable Natural Language Understanding*. Ph.D. thesis, University of Colorado at Boulder.

Ghazaleh Kazeminejad, Martha Palmer, Susan Windisch Brown, and James Pustejovsky. 2022. Componential analysis of english verbs. *Frontiers in Artificial Intelligence*, 5.

Ghazaleh Kazeminejad, Martha Palmer, Tao Li, and Vivek Srikumar. 2021a. Automatic entity state annotation using the verbnet semantic parser. In *Proceedings of The Joint 15th Linguistic Annotation Workshop (LAW) and 3rd Designing Meaning Representations (DMR) Workshop*, pages 123–132.

Ghazaleh Kazeminejad, Martha Palmer, Tao Li, and Vivek Srikumar. 2021b. Automatic entity state annotation using the VerbNet semantic parser. In *Proceedings of the Joint 15th Linguistic Annotation Workshop (LAW) and 3rd Designing Meaning Representations (DMR) Workshop*, pages 123–132, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Paul R Kingsbury and Martha Palmer. 2002. From treebank to propbank. In *LREC*, pages 1989–1993.

Inderjeet Mani and James Pustejovsky. 2012. *Interpreting motion: Grounded representations for spatial language*. 5. Oxford University Press.

Bhavana Dalvi Mishra, Lifu Huang, Niket Tandon, Wen-tau Yih, and Peter Clark. 2018. Tracking state changes in procedural text: a challenge dataset and models for process paragraph comprehension. *arXiv preprint arXiv:1805.06975*.

Bhavana Dalvi Mishra, Niket Tandon, Antoine Bosselut, Wen-tau Yih, and Peter Clark. 2019. Everything happens for a reason: Discovering the purpose of actions in procedural text. *arXiv preprint arXiv:1909.04745*.

James Pustejovsky and Jessica L Moszkowicz. 2011. The qualitative spatial dynamics of motion in language. *Spatial Cognition & Computation*, 11(1):15–44.

Danilo Ribeiro, Thomas Hinrichs, Maxwell Crouse, Kenneth Forbus, Maria Chang, and Michael Witbrock. 2019. Predicting state changes in procedural text using analogical question answering. In *7th Annual Conference on Advances in Cognitive Systems*.

Karin Kipper Schuler. 2005. *VerbNet: A broad-coverage, comprehensive verb lexicon*. University of Pennsylvania.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI conference on artificial intelligence*.

Leonard Talmy. 1988. Force dynamics in language and cognition. *Cognitive science*, 12(1):49–100.

Niket Tandon, Bhavana Dalvi, Keisuke Sakaguchi, Peter Clark, and Antoine Bosselut. 2019. Wiqa: A dataset for “what if...” reasoning over procedural text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6076–6085.

Jialong Tang, Hongyu Lin, Meng Liao, Yaojie Lu, Xianpei Han, Le Sun, Weijian Xie, and Jin Xu. 2022. Procedural text understanding via scene-wise evolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11367–11375.

Jizhi Tang, Yansong Feng, and Dongyan Zhao. 2020. Understanding procedural text using interactive entity networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7281–7290.

Shubham Toshniwal, Patrick Xia, Sam Wiseman, Karen Livescu, and Kevin Gimpel. 2021. On generalization in coreference resolution. In *Proceedings of the Fourth Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 111–120.

Zhihan Zhang, Xiubo Geng, Tao Qin, Yunfang Wu, and Daxin Jiang. 2021. Knowledge-aware procedural text understanding with multi-stage training. In *Proceedings of the Web Conference 2021*, pages 3512–3523.