Semantic Frame Forecast

Chieh-Yang Huang and Ting-Hao (Kenneth) Huang
Pennsylvania State University, University Park, PA 16802, USA
{chiehyang, txh710}@psu.edu

Abstract

This paper introduces **semantic frame forecast**, a task that predicts the semantic frames that will occur in the next 10, 100, or even 1,000 sentences in a running story. Prior work focused on predicting the immediate future of a story, such as one to a few sentences ahead. However, when novelists write long stories, generating a few sentences is not enough to help them gain high-level insight to develop the follow-up story. In this paper, we formulate a long story as a sequence of "story blocks," where each block contains a fixed number of sentences (*e.g.,* 10, 100, or 200). This formulation allows us to predict the follow-up story arc beyond the scope of a few sentences. We represent a story block using the term frequencies (TF) of **semantic frames** in it, normalized by each frame's inverse document frequency (IDF). We conduct semantic frame forecast experiments on 4,794 books from the Bookcorpus and 7,962 scientific abstracts from CODA-19, with block sizes ranging from 5 to 1,000 sentences. The results show that automated models can forecast the follow-up story blocks better than the random, prior, and replay baselines, indicating the task's feasibility. We also learn that the models using the frame representation as features outperform all the existing approaches when the block size is over 150 sentences. The human evaluation also shows that the proposed frame representation, when visualized as word clouds, is comprehensible, representative, and specific to humans. Our code is available at: https://github.com/appleternity/FrameForecasting.

1 Introduction

Writing a good novel is hard. Creative writers can get stuck in the middle of their drafts and struggle to develop follow-up scenes. Writing support systems, such as Heteroglossia (Huang et al., 2020a), generate paragraphs or ideas to help writers figure out the next part of the ongoing story. However, little literature focuses on plot prediction for **long stories**. Much prior work focused on predicting the immediate future of a story, *i.e.,* one to a few sentences later. For example, the Creative Help system used a recurrent neural network model to generate the next sentence to support writing (Roemmele and Gordon, 2015); the Scheherazade system uses crowdsourcing and artificial intelligence techniques to interactively construct the narrative sentence by sentence (Li and Riedl, 2015); Clark et al. (2018) study machine-in-the-loop story writing where the machine constantly generates a suggestion for the next sentence to stimulate writers; and Metaphoria (Gero and Chilton, 2019) generates metaphors, an even smaller unit, to inspire writers based on an input word by searching relations and ranking distances on ConceptNet (Liu and Singh, 2004).

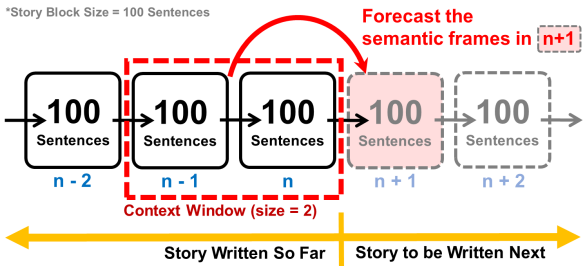

Figure 1: The semantic frame forecast is a task that predicts the semantic frames that will occur in the next part of a story based on the texts written so far.

Generating a coherent story across multiple sentences is challenging, even with cutting-edge pretrained models (See et al., 2019). To generate coherent stories, researchers often first generate a high-level representation of the story plots and then use it as a guide to generate a full story. For example, Martin et al. (2018) propose an event representation that uses an *SVO* tuple to generate story plots; Plan-and-write (Yao et al., 2019) uses the RAKE algorithm (Rose et al., 2010) to extract the keyword in each sentence to form a storyline and treat it as an intermediate representation; Fan et al. (2019)

use predicate-argument pairs annotated by semantic role labelers to model the structure of stories; and Zhang et al. (2020) take words with a certain part-of-speech tag as anchors and show that using anchors as the intermediate representation can improve the story quality. However, these projects all focused on short stories: The event representation is developed on a Wikipedia movie plot summary dataset (Bamman et al., 2013), where a summary has an average of 14.52 sentences; Plan-and-write uses the ROCStories dataset (Mostafazadeh et al., 2016), where each story has only 5 sentences; Fan *et al.* test their algorithm on the WritingPrompts dataset (Fan et al., 2018), where stories have 734 words (around 42 sentences) on average; and Zhange *et al.*'s anchor representation is developed on the VIST dataset (Huang et al., 2016), where a story has 5 sentences.

All the existing intermediate representations are generated on a sentence basis, meaning that the length of the representations increases along with the story length. That is, when applying these representations to **novels that usually have more than 50,000 words** (as defined by the National Novel Writing Month (wik, 2020)), it is not likely that such representations can still work. We thus introduce a new **Frame Representation** that compiles semantic frames into a fixed-length TF-IDF vector and a **Semantic Frame Forecast** task that aims to predict the next frame representation using the information in the current story block (see Figure 1). Two different datasets are built to examine the effectiveness of the proposed frame representation: one from Bookcorpus (Zhu et al., 2015), a fiction dataset; and one from CODA-19 (Huang et al., 2020b), a scientific abstract dataset. We establish several baselines and test them on different story block sizes, up to 1,000 sentences. The result shows that the proposed frame representation successfully captures the story plot information and helps the semantic frame forecast task, especially for story blocks with more than 150 sentences. To enable humans to perceive and comprehend frame representations, we further propose a process that visualizes a vector-based frame representation as word clouds. Human evaluations show that word clouds represent a story block with reasonable specificity, and our proposed model produces word clouds that are more representative than that of BERT.

2 Related Work

Automated Story Generation. Classic story generation focuses on generating logically coherent stories, plot planning (Riedl and Young, 2010; Li et al., 2013), and case-based reasoning (Gervás et al., 2004). Recently, several neural story generation models have been proposed (Peng et al., 2018; Fan et al., 2018), even including massive pretrained models (Radford et al., 2019; Keskar et al., 2019). However, researchers realize that word-by-word generation models cannot efficiently model the long dependency across sentences (See et al., 2019). Models using intermediate representations as guidance to generate stories are then proposed (Yao et al., 2019; Martin et al., 2018; Ammanabrolu et al., 2020; Fan et al., 2019; Zhang et al., 2020). These works are developed toward short stories and thus are insufficient for modeling novels (See Section 1).

Automated Story Understanding. Story understanding is a longstanding goal of AI (Roemmele and Gordon, 2018). Several tests were proposed to evaluate AI models' ability to reason the event sequence in a story. Roemmele *et al.* (2011) proposed the Choice of Plausible Alternatives (COPA) task, focusing on commonsense knowledge related to identifying causal relations between sequences. Mostafazadeh *et al.* (2016) proposed the Story Cloze Test, in which the model is required to select which of two given sentences best completes a particular story. Ippolito *et al.* (2019) proposed the Story Infilling task, which aims to generate the middle span of a story that is coherent with the foregoing context and will reasonably lead to the subsequent plots. Under the broader umbrella of story understanding, some prior work aimed to predict the next event in a story (Granroth-Wilding and Clark, 2016) or to identify the right follow-up line in dialogues (Lowe et al., 2016).

3 Semantic Frame Forecast

As shown in Figure 1, we formulate a long story as a sequence of fixed-length story blocks. Each story block (Figure 2 (1)) has a set of semantic frames (Figure 2 (2)) (Baker et al., 1998). We convert a story block into the **Frame Representation** (Figure 2 (3)), a TF-IDF vector over semantic frames, by computing the term frequency in that story block and the inverse document frequency over all the story blocks in the corpus. FrameNet (Baker et al.,

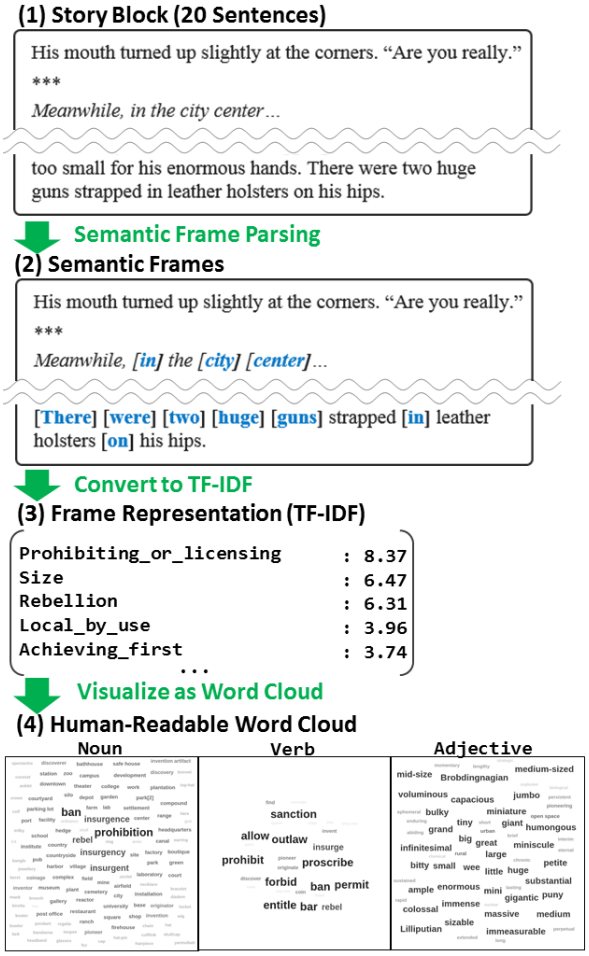

Figure 2: The steps to generate the frame representation for story blocks. The human-readable word clouds are generated to illustrate the conceptual meaning of the frame representation.

1998) defined a total of 1,221 different semantic frames, so the generated TF-IDF has 1,221 dimensions. The **Semantic Frame Forecast** is then defined as a task to predict the frame representation of the $n+1$-th story block using the foregoing content, namely the n-th story block.

Evaluation Metric. We use Cosine Similarity between the predicted vector and the gold-standard vector (complied from the human-written story block) for evaluation. Many other metrics, such as Mean-Squared Error (MSE), also exist to measure the distance between two vectors.

4 Data

We build the dataset from the existing Bookcorpus dataset (Zhu et al., 2015) and CODA-19 dataset (Huang et al., 2020b). This section describes how we preprocess the data, remove undesired content, and build the final dataset.

Bookcorpus Dataset. We obtain a total of 15,605 raw books and their corresponding meta data. To get high-quality fictional content, we remove books using the following heuristic rules: (i) short books whose size is less than 10KB; (ii) books that contain HTML code; (iii) books that are in the epub format (an e-book file format); (iv) books that are not in English; (v) books that are in the "Non-Fiction" genre; (vi) books that are in the "Anthologies" genre; (vii) books that are in the "Graphic Novels & Comics" genre. Since most books contain book information, author information, and some nonfictional content at the beginning and end of the book, we use regular expressions to match the term "Chapter" to locate the chapter title. Only the contents between the first chapter title and the last chapter title are kept. The last chapter is also removed as there are no certain boundaries to identify the story ending. Books whose chapter titles are unlocatable are also removed. After removing all the unqualified books, a total of 4,794 books were used in our dataset. We transliterate all non-ASCII characters into ASCII characters using Unidecode (https://pypi.org/project/Unidecode/) to fulfill the requirement of Open-SESAME (Swayamdipta et al., 2017). Open-SESAME is then used to parse the semantic frames for each sentence.

The books are split into training/validation/test sets following a 70/10/20 split, resulting in 3,357, 479, and 958 books, respectively. To measure the effect of frame representation for different context lengths, we vary the story block length, using 5, 10, 20, 50, 100, 150, 200, 300, 500, and 1,000 sentences. When creating instances, we first split a book into story blocks with the specified length and extract all the consecutive two story blocks as instances when context window size (see Figure 1) is set to 1. The IDF of the semantic frame is then computed over the story blocks using all the training sets. Combining with the TF value in each story block, we convert story blocks into frame representations. We use scikit-learn's implementation (Pedregosa et al., 2011) of TF-IDF but with a slight modification on IDF: Scikit-learn uses $idf(t) = log(\frac{n}{df(t)+1})$ to compute a smoothing IDF, but we use $idf(t) = log(\frac{n}{df(t)})$. The detailed statistic information is shown in Table 1.

CODA-19 Dataset. We envision a broader definition of "creativity" in writing and attempt to apply story arc prediction technologies to the do-

Block Size	5	10	20	50	100	150	200	300	500	1000
# Words Mean	71.7	143.5	286.9	717.2	1433.9	2149.8	2865.3	4293.7	7142.5	14212.3
# Frames Mean	17.5	35.0	69.9	174.5	348.6	522.1	695.4	1040.7	1727.3	3417.2
# Events Mean	10.0	20.0	39.9	99.8	199.4	298.9	398.2	596.4	991.2	1967.1
# Train	3,744,948	1,869,947	932,464	369,941	182,479	119,967	88,720	57,455	32,469	13,749
# Valid	574,840	287,054	143,166	56,838	28,073	18,466	13,672	8,881	5,035	2,166
# Test	1,054,816	526,687	262,625	104,198	51,396	33,776	24,987	16,178	9,138	3,861

Table 1: The statistic of Bookcorpus dataset in ten different story block lengths. We use Open-Sesame to parse the semantic frame for each sentence. The *Events* represents the SVO tuples (Martin et al., 2018).

Block Size	1	3	5
# Words Mean	26.3	77.3	124.7
# Frames Mean	6.0	17.5	27.6
# Events Mean	1.2	3.5	5.6
# Train	48,489	9,858	2,739
# Valid	5,615	1,146	334
# Test	5,238	1,047	287

Table 2: The statistic of CODA-19 dataset in three different story block lengths. We use Open-Sesame to parse the semantic frame for each sentence. The *Events* represents the SVO tuples (Martin et al., 2018).

mains outside novels, for example, scholarly articles. As an earlier exploration, we choose to use a smaller set of human-annotated abstracts (CODA-19 (Huang et al., 2020b)) rather than machine-extracted full text (CORD-19 (Wang et al., 2020a)) in our proof-of-concept study, avoiding formatting issues (*e.g.*, reference format, parsing errors) and intensive data cleaning effort. The original CODA-19 dataset contains $10,966$ human-annotated English abstracts for five different aspects: Background, Purpose, Method, Finding/Contribution, and Other. We remove sentences that are annotated as "Other," an aspect for sentences that are not directly related to the content (*e.g.,* terminology definitions or copyright notices.) Abstracts that contain Unicode characters are also removed. A total of $7,962$ abstracts are used in our dataset. We then use Open-SESAME to parse the semantic frames for each sentence. We adopt CODA-19's original split, where the training set, validation set, and testing set have $6,509$, 737, and 716 abstracts, respectively. Three different lengths of story block are used: 1, 3, and 5. We then create instances and compute TF-IDF as described above. Table 2 shows the details.

5 Models

We implement two naive baselines, an information retrieval baseline, two machine learning baselines, two deep learning baselines, an existing model and a text generation baseline.

Replay Model. For each instance, the replay model takes the frame representation in the n-th story block as the prediction, *i.e.*, the same frames will occur again.

Prior Model. The prior model computes the mean of the frame representation over the training set and uses it as the prediction for all the testing instances.

Information Retrieval with Frame Representation. For each instance, the information retrieval model searches for the most similar story block in the training set and takes the frame representation from its next story block as the prediction. In this setting, we adopt the cosine similarity on frame representations to measure the story similarity. For block size 5 in the Bookcorpus dataset, there are around 3.7 million instances in the training set, which is infeasible to finish.

Random Forest with Frame Representation. The foregoing story block's frame representation is used as the feature for prediction. We use scikit-learn's implementation of Random Forest Regressor (Pedregosa et al., 2011) with a max depth of 3 and 20 estimators. For block sizes that have more than one million training instances (5 and 10 in the Bookcorpus dataset), we randomly sample one million instances to train the model.

LGBM with Frame Representation. This is the same as the previous setup but trained using the LGBM Regressor model (Ke et al., 2017) with the max depth 5, the number of leaves 5, and the number of estimators 100. For block sizes that have more than one million training instances (5 and 10 in the Bookcorpus dataset), we randomly sample one million instances to train the model.

DAE with Frame Representation. This is the same as the previous setting but trained with the

Denoising Autoencoder architecture (Bengio et al., 2013). We feed in the foregoing story block's frame representation and output the frame representation for the follow-up story block. Thirty percent of the input is dropped randomly. The model is optimized using the cosine distance ($1 - cosine\ similarity$). Both the encoder and decoder are created via five dense layers with a hidden size of 512. We use a learning rate of 1e-5 and a batch size of 512 and train the model with the early stopping criteria of no improvement for 20 epochs. The best model on the validation set is kept for testing.

Event Representation Model (Event-Rep). We use Martin *et al.*'s event representation (2018) on the foregoing story block as the feature. An event tuple is defined as $\langle s, v, o, m \rangle$, where s is the subject, v is the verb, o is the object, and m is the verb modifier. We extract the dependency relation using the Stanza parser (Qi et al., 2020). Unlike Martin *el al.*'s implementation, where the empty placeholder $\varnothing$ only replaces unidentified objects and modifiers, we find that the subjects can also be frequently missing in fiction books. For example, in "*"**Come** out?" Zack asked. "**Come** out of where?"*". In both cases here, the verb "come" does not have a subject. In "*Fine, **follow** me.*", "follow" has an object but does not have a subject. Therefore, we allow s to have a $\varnothing$ placeholder in our implementation. All words are stemmed by NLTK (Loper and Bird, 2002).

After extracting the event representation, the sequence of event tuples in the foregoing story block is fed into a five-layer LSTM model (Hochreiter and Schmidhuber, 1997) to predict its follow-up frame representation. Note that the length of the event tuple sequence changes along with the block size. We thus set the maximum length of the sequence to the 95th percentile of the length in the training set. Sequences longer than the maximum length are left-truncated. The model is trained with a hidden size of 512, a learning rate of 3e-5, a dropout rate of 0.05, and a batch size of 64. We optimize the model using the cosine distance and apply the early stopping criteria of no improvement for three epochs. The best model on the validation set is kept for testing.

BERT. We take the pure text in the foregoing story block as the feature and apply the pretrained BERT model (Devlin et al., 2019). BERT has a token length limitation, so we set the maximum length of tokens to 500 for Bookcorpus and 300 for CODA-19. Sentences with more than 500 tokens are truncated from the left. We take the `[CLS]` token representation from the last layer and add a dense layer on top of it to predict the follow-up frame representation. The model is trained with a learning rate of 1e-5 and a batch size of 32. We optimize the model using the cosine distance and apply the early stopping when no improvement for five epochs. The model with the best score on the validation set is kept for testing.

SciBERT (For CODA-19 Only). This is the same as the previous setting but is trained using the pretrained SciBERT model (Beltagy et al., 2019). We only test this approach on the CODA-19 dataset since it is from the scientific domain.

GPT-2 (For Bookcorpus Only). We also include a text generation model, GPT-2 (gpt2-xl) (Radford et al., 2019) with block sizes of 5, 10, 20, and 50. Since GPT-2 is computationally expensive, we conduct the experiment on a subset of the dataset, where 1,000 instances are randomly selected. We feed the text in the latest story block (n) into GPT-2 and generate 70, 150, 300, and 700 words for block sizes 5, 10, 20, and 50, respectively (5 sentences $\approx$ 70 words; 10 sentences $\approx$ 150 words in Bookcourpus, etc). For stories that exceed the GPT-2's word limit, we truncate the text from the left. Stories with block size larger than 100 would have more than 1400 words which by itself exceed the GPT-2's word limit. Generated stories are then parsed by Open-SESAME to extract the semantic frames and turned into frame representations as the predictions.

6 Experimental Results and Analysis

Table 3 and Table 4 show the experimental results. In this section, we summarize the main findings.

Predicting forthcoming semantic frames is remarkably challenging yet possible. Machine-learning models outperform the two naive baselines for different story lengths. In the Bookcorpus dataset, BERT performs the best for story blocks under 100 sentences, while LGBM performs the best for story blocks over 150 sentences. In the CODA-19 dataset, SciBERT performs the best for block sizes of 1 and 3, while DAE performs the best for a block size of 5. While the task is very challenging, these results shed light on the semantic frame forecast task. However, the improvement

Feature	Model	Block Size									
		5	**10**	**20**	**50**	**100**	**150**	**200**	**300**	**500**	**1000**
-	Replay Baseline	.0654	.0915	.1237	.1737	.2163	.2448	.2665	.3000	.3462	.4155
-	Prior Baseline	.2029	.2435	.2857	.3389	.3754	.3962	.4105	.4302	.4528	.4776
Frame	IR Baseline	-	.0631	.0851	.1290	.1841	.2085	.2262	.2536	.2859	.3321
Frame	Random Forest	.2037	.2448	.2881	.3427	.3807	.4025	.4184	.4402	.4659	.4966
Frame	LGBM	.2072	.2506	.2967	.3564	.3995	**.4255**	**.4441**	**.4711**	**.5048**	**.5510**
Frame	DAE	.2082	.2515	.2966	.3547	.3976	.4223	.4400	.4598	.4898	.5280
Event	Event-Rep	.2111	.2541	.2994	.3532	.3929	.4126	.4280	.4453	.4626	.4792
Text	BERT	**.2172**	**.2611**	**.3073**	**.3637**	**.4012**	.4229	.4371	.4559	.4779	.5057
Text	GPT-2	.0519	.0739	.0990	.1402	-	-	-	-	-	-
	DELTA	.0142	.0176	.0216	.0249	.0257	.0293	.0336	.0409	.0520	.0734

Table 3: Baseline result for Bookcorpus dataset. BERT and Event-Rep work better in smaller block sizes, while models using frame representation perform better in larger block sizes. DELTA represents the difference between the best model and the prior baseline — an extremely simple but strong baseline — in that specific block size. The small value of DELTA shows that semantic frame forecast is challenging yet possible.

Feature	Model	Block Size		
		1	**3**	**5**
-	Replay Baseline	.0524	.0971	.1363
-	Prior Baseline	.1573	.2067	.2288
Frame	IR Baseline	.0315	.0601	.0752
Frame	Random Forest	.1581	.2081	.2278
Frame	LGBM	.1561	.2024	.2094
Frame	DAE	.1611	.2155	**.2380**
Event	Event-Rep	.1595	.2118	.2332
Text	BERT	.1660	.2202	.2353
Text	SciBERT	**.1675**	**.2219**	.2339
	DELTA	.0102	.0152	.0092

Table 4: Baseline result for CODA-19 dataset. SciBERT performs the best in block size 1 and 3. Using the frame representation as the feature, DAE performs the best for block size 5. DELTA shows the difference between the best model and the prior baseline in that specific block size. The small value of DELTA shows that semantic frame forecast is challenging yet possible.

is not big, as shown in the DELTA row, suggesting that semantic frame forecast requires more investigation and understanding.

"Prior" is a robust and strong baseline. In both the Bookcorpus dataset and the CODA-19 dataset, the prior baseline is strong. As the story gets longer, the performance also increases. This suggests that when the story block gets bigger, more and more frames will constantly occur.

Replay baseline shows the relation of consecutive story blocks. The replay baseline assumes that the events that happen now will likely happen again shortly. The results in Table 3 and Table 4 partially confirm this assumption. To understand more about the assumption, we use the replay baseline to predict the n+i-th story block from the n-th story block in the Bookcorpus dataset. Figure 3

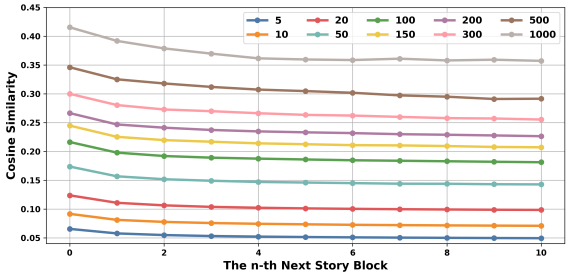

Figure 3: Using the replay baseline to predict the n+i-th story block from the n-th story block (story block size = 5, 10, $\cdots$, 1000.) Things that happen in the current story block are more likely to happen again shortly.

shows the results. We can see that things that happen now will be more likely to happen in the near future compared to story blocks farther from the current one.

Event-Rep works better in short stories. In the Bookcorpus dataset, event representation works better than the frame representation in small block sizes (5, 10, and 20). However, starting from a block size of 50, the model cannot perform as well as the other models. We thus conclude that event representation works better in short stories. The main reason is that event representations are generated on a sentence-by-sentence basis and will create overwhelming information on long stories. The existing intermediate representations (see Section 1) are mostly generated from sentences and will likely have the same issue as the event representation. Compared to the existing works, the proposed frame representation encodes a story block, no matter how long it is, into a fixed-length vector and therefore performs better on longer stories.

Feature	Model	Block Size						
		5	10	20	50	100	150	200
-	Prior	.2029	.2435	.2856	.3388	.3754	.3962	.4105
Frame	IR	.0401	.0615	.0900	.1368	.1775	.2051	.2262
Frame	RF	.2030	.2440	.2871	.3418	.3801	.4025	.4184
Frame	LGBM	.2033	.2472	.2935	.3540	**.3980**	**.4248**	**.4441**
Frame	DAE	.2058	.2482	.2929	.3507	.3926	.4178	.4400
Event	Event-Rep	.2046	.2470	.2905	.3454	.3799	.4069	.4171
Text	BERT	**.2088**	**.2529**	**.2981**	**.3550**	.3949	.4178	.4371

Table 5: Result of the downsampling experiment. Although all the performance drops, the observations we find are still true. Therefore, the conclusions are not merely caused by the effect of data size.

BERT performs very well in short stories. The results of BERT and SciBERT in Table 3 and Table 4 show that textual information is helpful in predicting story blocks. BERT performs better when the block size is under 100 in the Bookcorpus dataset and below 3 in CODA-19. However, handling long texts remain challenging for BERT, as its computational complexity scales with the square of the token length. Researchers started reducing the computation complexity for transformer-based models to allow modeling on long texts such as Linformer (Wang et al., 2020b), Longformer (Beltagy et al., 2020), Reformer (Kitaev et al., 2020), and BigBird (Zaheer et al., 2020). However, these models still require a lot of computation power and are not yet ready for general use.

The good performance does not merely come from the number of instances. Deep learning methods often require more instances for training. To show that the result in Table 3 is not mainly caused by the number of instances, we conduct the same experiment in Bookcorpus dataset using $88,720$ training instances for block sizes ranging from 5 to 200. Table 5 shows the results. The performance is affected, but the conclusions we make above still stand, showing that the number of instances is not the main factor for our observations. Meanwhile, we find that BERT is affected more than LGBM. In Table 5 the performance of BERT drops by -0.0092 to -0.0051 compared to Table 3, but LGBM only drops -0.0039 to -0.0007. Although this suggests that the number of instances can cause the difference, it also shows that the frame representation can be used with fewer instances.

GPT-2 is not effective. GPT-2 is not effective in predicting the story flow even though it can generate reasonable sentences. Even the naive Replay baseline outperforms the GPT-2 baseline in predicting the story block. We hypothesize that GPT-2 is not good at maintaining the coherence among sentences or events, especially in the creative writing domain. Similar phenomenons are also observed by others and used to motivate the need for guided generation models or progressive generation models (Wang et al., 2020c; Tan et al., 2020).

window	Feature	Model	Block Size		
			20	50	100
2	Frame	LGBM	.2989	.3590	**.4029**
	Text	BERT	**.3081**	**.3625**	.4002
5	Frame	LGBM	.2989	.3617	**.4065**
	Text	BERT	**.3082**	**.3618**	.3985

Table 6: Results of using 2 or 5 foregoing story blocks to predict the *n+1*-th story block. LGBM improves further when using more context but BERT fails to model the longer context, and its performance even gets hurt.

Frame	Lexical Units
Most Important Frames (Out of 50)	
Kinship	father, mother, son, daughter
Biological_urge	tired, sleepy, randy, hungry
Connectors	ribbon, rope, thread, string
Firefighting	fight, battle, control, tackle
Origin	Chinese, American, Vietnamese, origin
Least Important Frames (Out of 50)	
Proper_reference	proper, self
Cause_to_start	spark, generate, arouse, bring about
Friction	grate, squeal, scrunch, screech
Dominate_competitor	dominate, domination, dominant, strongman
State_continue	remain, stay, rest

Table 7: The most and least important five frames (from 50 random frames) identified in the ablation study.

6.1 Using a Larger Context Window

This paper focuses on using 1 story block to forecast the next one, *i.e.,* window size = 1 (see Figure 1.) As a proof of concept, we use 2 and 5 blocks (window size = 2 and 5) for prediction, respectively. We use two models: LGBM with frame representation, and BERT with text. For LGBM, we simply concatenate the frame representation from the input story blocks to create the input vector. For BERT, we put the event tuple and the text together as the input. Table 6 shows the results. While BERT does not benefit from using more contexts, LGBM's performance improves, suggesting the potentials of using a larger context window. More research is required to understand the effects.

6.2 Which Semantic Frames Affect the Follow-Up Story More?

Different frames may contribute differently to the prediction of the follow-up story. To understand which frame plays a more important role in the story, we conduct an ablation study by investigating the LGBM model on block 150. We obliterate one frame from the input frame representation and record the performance change, where a higher performance deduction means the frame removed is more important. A total of 50 frames are selected randomly for the ablation study. Table 7 shows the top and bottom five frames. We hypothesize that the more generic frames, such as "State_continue" and "Proper_reference," might be less important to the follow-up stories, but it will require more research to understand the impacts fully.

7 Human Evaluation

We further evaluate the proposed method with humans. We first visualize the vector of semantic frames into **word clouds** so that humans can perceive and comprehend it. We then use online crowd workers to test the *(i)* representativeness and *(ii)* the specificity of the produced word clouds.

Visualizing Semantic Frame Vectors into Word Clouds. Figure 4 shows the workflow of generating word clouds based on a frame representation (*i.e.,* a TF-IDF vector). In FrameNet, "lexical units" are the terms that can trigger a specific frame. Compared to showing the name and definition of a frame, lexical units are easier for people to read and comprehend. Therefore, we use the top 30 frames (ranked by their TF-IDF weights) and randomly select up to three lexical units for each frame to form a word cloud. The size and color of the lexical unit is computed according to the frame's TF-IDF weight, where a higher TF-IDF value will result in a larger font and darker color. Finally, we arrange the lexical units into three word clouds on nouns, verbs, and adjectives using their POS tags. All the word clouds are generated using `d3-cloud` (Davies, 2016).

7.1 Representativeness

This task evaluates which model can generate the most representative word cloud for a story block.

Task Setups. In this Human Intelligence Task (HIT), we show a story block ($n+1$) and two or three [noun, verb, adjective] word clouds ($n+1$) produced by different models based on the previous story block (n). The goal is to measure, from the users' perspective, how much the generated word clouds represent the actual human-written follow-up stories. We display the actual next story block ($n+1$) and the word clouds produced by different models based on the latest story block (n). The workers from Amazon Mechanical Turk (MTurk) are asked to read the story and select the word cloud that better represents the story block. In the worker interface, we set up a 3-minutes lock for submission and a reach-to-the-bottom lock for the story panel to make sure the workers read the story. Nine different workers are recruited for each task[1]. We empirically estimate the working time to be less than 6 minutes per HIT and set the price to \$0.99/HIT (hourly wage = \$10).

We choose block size 150 to compare two models: LGBM with frame representation and BERT with text. Ground-truth word clouds are also added to some of the HITs to check the validity of the task. A total of 150 instances are randomly selected from Boocorpus testing set. For each instance, the foregoing story block is feed into LGBM and BERT to predict the frame representation of the follow-up story block. Out of 150 instances, 50 instances are conducted with ground truth, where a total of three word clouds are shown. Another 100 instances are used for comparing LGBM against BERT directly.

Results. Over the 50 HITs where ground truth is included, (ground truth, LGBM, BERT) wins (32, 15, 16) HITs, respectively (ties exist.) Nine assignments are recruited from 9 workers for each HIT. Regarding to the assignment voting, (ground truth, LGBM, BERT) gets (199, 131, 120) votes, respectively. The result suggests that humans can correctly perceive the word clouds' conceptual meaning as the ground truth is rated the best.

Over the 100 HITs where LGBM and BERT are compared directly, (LGBM, BERT) wins (59, 41) HITs. Regarding the assignment voting, (LGBM, BERT) gets (472, 428) votes, respectively. The result shows that LGBM is better than BERT in a block size of 150, which aligns with our automatic evaluation results using cosine similarity (see Section 6.)

[1]Four built-in worker qualifications are used: HIT Approval Rate ($\geq$98%), Number of Approved HITs ($\geq$ 3000), Locale (US Only), and Adult Content Qualification.

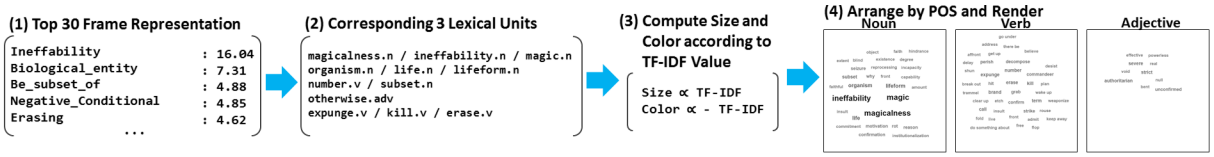

Figure 4: The workflow to visualize the word clouds from frame representation. The top semantic frames are used where each is illustrated by a maximum of three corresponding lexical units. The size and the color of the lexical units are computed according to the TF-IDF value.

7.2 Specificity

This task evaluates whether using the proposed word cloud to represent a story block is specific enough for humans to distinguish the correct story from the distractor.

Task Setups. In this HIT, we show two story blocks (n) and one set of [noun, verb, adjective] word clouds (n). Note that the current story block (n) and its ground-truth word cloud (n) are used to examine if humans can correctly perceive the semantic information from word cloud visualization. One story block is the answer that is referred to by the word clouds and the other one is a distractor. Workers are asked to read the two story blocks and select the story block that is referred to by the word clouds. Nine different workers are recruited for each HIT. We use the same worker interface design and built-in worker qualifications as that of Section 7.1. A HIT takes estimatedly 2.33 minutes and is priced at $0.38.

We choose block size 20 and use the ground-truth word clouds for this experiment. Fifty instances from 50 different books are randomly selected from Bookcorpus testing set. We also randomly select a 20-sentences story block from a different book as the distractor.

Results. Of the 450 assignments, 63.8% of the answers were correct. When aggregating the assignments using majority voting, 74% of 50 HITs were answered correctly. We thus believe that it is reasonably specific for humans to represent a story block using the proposed word clouds.

8 Conclusion

This paper proposes a semantic frame forecast task that aims to forecast the semantic frames in the next 10, 100, or even 1,000 sentences of a story. A long story is formulated as a sequence of story blocks that contain a fixed number of sentences. We further introduce a frame representation that can encode a story block into a fixed-length TF-IDF vector over semantic frames. Experiments on both the Bookcorpus dataset and CODA-19 dataset show that the proposed frame representation helps semantic frame forecast in large story blocks. By visualizing the frame representation as word clouds, we also show that it is comprehensible, representative, and specific to humans. In the future, we will introduce the frame representation into story generation models to ensure coherence when generating long stories. We will also explore the possibility of supporting writers to develop the next part of their stories by generating semantic frames as clues using semantic frame forecast.

Acknowledgments

We would like to thanks the Huck Institutes of the Life Sciences' Coronavirus Research Seed Fund (CRSF) and the College of IST COVID-19 Seed Fund at Penn State University who support the construction of CODA-19. We also thank Tiffany Knearem for the feedback for designing word cloud visualization and workers who participated the human evaluation study.

References

2020. National novel writing month.

Prithviraj Ammanabrolu, Ethan Tien, Wesley Cheung, Zhaochen Luo, William Ma, Lara J Martin, and Mark O Riedl. 2020. Story realization: Expanding plot events into sentences. In *AAAI*, pages 7375–7382.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, ACL '98/COLING '98, pages 86–90, Stroudsburg, PA, USA. Association for Computational Linguistics.

David Bamman, Brendan O'Connor, and Noah A Smith. 2013. Learning latent personas of film characters. In *Proceedings of the 51st Annual Meeting of*

the Association for Computational Linguistics (Volume 1: Long Papers), pages 352–361.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific 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 3606–3611.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150*.

Yoshua Bengio, Li Yao, Guillaume Alain, and Pascal Vincent. 2013. Generalized denoising auto-encoders as generative models. In *Advances in neural information processing systems*, pages 899–907.

Elizabeth Clark, Anne Spencer Ross, Chenhao Tan, Yangfeng Ji, and Noah A. Smith. 2018. Creative writing with a machine in the loop: Case studies on slogans and stories. In *23rd International Conference on Intelligent User Interfaces*, IUI '18, pages 329–340, New York, NY, USA. ACM.

J Davies. 2016. D3-cloud.(2016).

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

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann Dauphin. 2019. Strategies for structuring story generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2650–2660, Florence, Italy. Association for Computational Linguistics.

Katy Ilonka Gero and Lydia B. Chilton. 2019. Metaphoria: An algorithmic companion for metaphor creation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–12, New York, NY, USA. Association for Computing Machinery.

Pablo Gervás, Belén Díaz-Agudo, Federico Peinado, and Raquel Hervás. 2004. Story plot generation based on cbr. In *International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pages 33–46. Springer.

Mark Granroth-Wilding and Stephen Clark. 2016. What happens next? event prediction using a compositional neural network model. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Chieh-Yang Huang, Shih-Hong Huang, and Ting-Hao Kenneth Huang. 2020a. Heteroglossia: In-situ story ideation with the crowd. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–12.

Ting-Hao Huang, Chieh-Yang Huang, Chien-Kuang Cornelia Ding, Yen-Chia Hsu, and C Lee Giles. 2020b. Coda-19: Using a non-expert crowd to annotate research aspects on 10,000+ abstracts in the covid-19 open research dataset. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*.

Ting-Hao Kenneth Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra, Aishwarya Agrawal, Jacob Devlin, Ross Girshick, Xiaodong He, Pushmeet Kohli, Dhruv Batra, C. Lawrence Zitnick, Devi Parikh, Lucy Vanderwende, Michel Galley, and Margaret Mitchell. 2016. Visual storytelling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1233–1239, San Diego, California. Association for Computational Linguistics.

Daphne Ippolito, David Grangier, Chris Callison-Burch, and Douglas Eck. 2019. Unsupervised hierarchical story infilling. In *Proceedings of the First Workshop on Narrative Understanding*, pages 37–43.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in neural information processing systems*, pages 3146–3154.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. In *International Conference on Learning Representations*.

Boyang Li, Stephen Lee-Urban, George Johnston, and Mark Riedl. 2013. Story generation with crowdsourced plot graphs. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*.

Boyang Li and Mark Riedl. 2015. Scheherazade: Crowd-powered interactive narrative generation. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Hugo Liu and Push Singh. 2004. Conceptnet—a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ETMTNLP '02, page 63–70, USA. Association for Computational Linguistics.

Ryan Lowe, Iulian Vlad Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. On the evaluation of dialogue systems with next utterance classification. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 264–269.

Lara J Martin, Prithviraj Ammanabrolu, Xinyu Wang, William Hancock, Shruti Singh, Brent Harrison, and Mark O Riedl. 2018. Event representations for automated story generation with deep neural nets. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Nanyun Peng, Marjan Ghazvininejad, Jonathan May, and Kevin Knight. 2018. Towards controllable story generation. In *Proceedings of the First Workshop on Storytelling*, pages 43–49, New Orleans, Louisiana. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Mark O Riedl and Robert Michael Young. 2010. Narrative planning: Balancing plot and character. *Journal of Artificial Intelligence Research*, 39:217–268.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*.

Melissa Roemmele and Andrew Gordon. 2018. An encoder-decoder approach to predicting causal relations in stories. In *Proceedings of the First Workshop on Storytelling*, pages 50–59.

Melissa Roemmele and Andrew S Gordon. 2015. Creative help: a story writing assistant. In *International Conference on Interactive Digital Storytelling*, pages 81–92. Springer.

Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. *Automatic Keyword Extraction from Individual Documents*, chapter 1. John Wiley & Sons, Ltd.

Abigail See, Aneesh Pappu, Rohun Saxena, Akhila Yerukola, and Christopher D. Manning. 2019. Do massively pretrained language models make better storytellers? In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 843–861, Hong Kong, China. Association for Computational Linguistics.

Swabha Swayamdipta, Sam Thomson, Chris Dyer, and Noah A. Smith. 2017. Frame-Semantic Parsing with Softmax-Margin Segmental RNNs and a Syntactic Scaffold. *arXiv preprint arXiv:1706.09528*.

Bowen Tan, Zichao Yang, Maruan AI-Shedivat, Eric P Xing, and Zhiting Hu. 2020. Progressive generation of long text. *arXiv preprint arXiv:2006.15720*.

Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Doug Burdick, Darrin Eide, Kathryn Funk, Yannis Katsis, Rodney Michael Kinney, et al. 2020a. Cord-19: The covid-19 open research dataset. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*.

Sinong Wang, Belinda Li, Madian Khabsa, Han Fang, and Hao Ma. 2020b. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*.

Su Wang, Greg Durrett, and Katrin Erk. 2020c. Narrative interpolation for generating and understanding stories. *arXiv preprint arXiv:2008.07466*.

Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. 2019. Plan-and-write: Towards better automatic storytelling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7378–7385.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *arXiv preprint arXiv:2007.14062*.

Bowen Zhang, Hexiang Hu, and Fei Sha. 2020. Visual storytelling via predicting anchor word embeddings in the stories. *arXiv preprint arXiv:2001.04541*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE International Conference on Computer Vision (ICCV)*.