A Joint Model for Semantic Sequences: Frames, Entities, Sentiments

Haoruo Peng Snigdha Chaturvedi Dan Roth
University of Illinois, Urbana-Champaign
{hpeng7,snigdha,danr}@illinois.edu

Abstract

Understanding stories – sequences of events – is a crucial yet challenging natural language understanding task. These events typically carry multiple aspects of semantics including actions, entities and emotions. Not only does each individual aspect contribute to the meaning of the story, so does the interaction among these aspects. Building on this intuition, we propose to jointly model important aspects of semantic knowledge – frames, entities and sentiments – via a semantic language model. We achieve this by first representing these aspects' semantic units at an appropriate level of abstraction and then using the resulting vector representations for each semantic aspect to learn a joint representation via a neural language model. We show that the joint semantic language model is of high quality and can generate better semantic sequences than models that operate on the word level. We further demonstrate that our joint model can be applied to story cloze test and shallow discourse parsing tasks with improved performance and that each semantic aspect contributes to the model.

1 Introduction

Understanding a story requires understanding sequences of events. It is thus vital to model semantic sequences in text. This modeling process necessitates deep semantic knowledge about what can happen next. Since events involve actions, participants and emotions, semantic knowledge about these aspects must be captured and modeled.

Consider the examples in Figure 1. In Ex.1, we observe a sequence of actions (commit, arrest, charge, try), each corresponding to a predicate frame. Clearly, "convict" is more likely than "go" to follow such sequence. This semantic knowledge can be learned through modeling frame sequences observed in a large corpus. This phenomena has already been studied in script learning works (Chatman, 1980; Chambers and Jurafsky, 2008b; Ferraro and Van Durme, 2016; Pichotta and Mooney, 2016a; Peng and Roth, 2016). However, modeling actions is not sufficient; participants in actions and their emotions are also important. In Ex. 2, Opt.2 is not a plausible answer because the story is about "football", and it does not make sense to suddenly change the key en-

Ex.1 (Actions - Frames) Steven Avery *committed* murder. He was *arrested*, *charged* and *tried*.
Opt.1 Steven Avery was convicted of murder.
Opt.2 Steven went to the movies with friends.
Alter. Steven was held in jail during his trial.

Ex.2 (Participants - Entities) It was my first time ever playing *football* and I was so nervous. During the game, I got tackled and it did not hurt at all!
Opt.1 I then felt more confident playing football.
Opt.2 I realized playing baseball was a lot of fun.
Alter. However, I still love baseball more.

Ex.3 (Emotions - Sentiments) Joe wanted to become a professional plumber. So, he applied to a trade school. Fortunately, he was *accepted*.
Opt.1 It made Joe very happy.
Opt.2 It made Joe very sad.
Alter. However, Joe decided not to enroll because he did not have enough money to pay tuition.

Figure 1: **Examples of short stories requiring different aspects of semantic knowledge.** For all stories, Opt.1 is the correct follow-up, while Opt.2 is the contrastive wrong follow-up demonstrating the importance of each aspect. Alter. showcases an alternative correct follow-up, which requires considering different aspects of semantics jointly.

Models	Context Input	Generated Ending
4-gram	Steven Avery committed murder. He was arrested, charged and tried.	With law by the judge <UNK> ...
RNNLM	*same as above*	The information under terrorism ...
Seq2Seq	*same as above*	He decided for a case.
FC-SemLM	commit.01 arrest.01 charge.05 try.01	convict.01
FES-LM	PER[new]-commit.01-ARG[new](NEG) ARG[new]-arrest.01-PER[old](NEU) ARG[new]-charge.05-PER[old](NEU) ARG[new]-try.01-PER[old](NEG)	ARG[new]-convict.01-PER[old](NEG)

Table 1: **Comparison of generative ability for different models.** For each model, we provide Ex.1 as context and compare the generated ending. 4-gram and RNNLM models are trained on NYT news data while Seq2Seq model is trained on the story data (details see Sec. 5). These are models operated on the word level. We compare them with FC-SemLM (Peng and Roth, 2016), which works on frame abstractions, i.e. "predicate.sense". For the proposed FES-LM, we further assign the arguments (subject and object) of a predicate with NER types ("PER, LOC, ORG, MISC") or "ARG" if otherwise. Each argument is also associated with a "[new/old]" label indicating if it is first mentioned in the sequence (decided by entity co-reference). Additionally, the sentiment of a frame is represented as positive (POS), neural (NEU) or negative (NEG). FES-LM can generate better endings in terms of soundness and specificity. The FES-LM ending can be understood as "[Something] convict a person, who has been mentioned before (with an overall negative sentiment)", which can be instantiated as "Steven Avery was convicted." given current context.

tity to "baseball". In Ex.3, one needs understand that "being accepted" typically indicates a positive sentiment and that it applies to "Joe".

As importantly, we believe that modeling these semantic aspects should be done jointly; otherwise, it may not convey the complete intended meaning. Consider the alternative follow-ups in Figure 1: in Ex.1, the entity "jail" gives strong indication that it follows the storyline that mentions "murder"; in Ex.2, even though "football" is not explicitly mentioned, there is a comparison between "baseball" and "football" that makes this continuation coherent; in Ex.3, "decided not to enroll" is a reasonable action after "being accepted", although the general sentiment of the sentence is negative. These examples show that in order to model semantics in a more complete way, we need to consider interactions between frames, entities and sentiments.

In this paper, we propose a joint semantic language model, FES-LM, for semantic sequences, which captures **F**rames, **E**ntities and **S**entiment information. Just as "standard" language models built on top of words, we construct FES-LM by building language models on top of joint semantic representations. This joint semantic representation is a mixture of representations corresponding to different semantic aspects. For each aspect, we capture semantics via abstracting over and disambiguating text surface forms, i.e. semantic frames for predicates, entity types for semantic arguments, and sentiment labels for the overall context. These abstractions provide the basic vocabulary for FES-LM and are essential for capturing the underlying semantics of a story. In Table 1, we provide Ex.1 as context input (although FC-SemLM and FES-LM automatically generate a more abstract representation of this input) and examine the ability of different models to generate an ending. 4-gram, RNNLM and Seq2Seq models operate on the word level, and the generated endings are not satisfactory. FC-SemLM (Peng and Roth, 2016) works on basic frame abstractions and the proposed FES-LM model adds abstracted entity and sentiment information into frames. The results show that FES-LM produces the best ending among all compared models in terms of semantic soundness and specificity.

We build the joint language model from plain text corpus with automatic annotation tools, requiring no human effort. In the empirical study, FES-LM is first built on news documents. We provide perplexity analysis of different variants of FES-LM as well as for the narrative cloze test,

where we test the system's ability to recover a randomly dropped frame. We further show that FES-LM improves the performance of sense disambiguation for shallow discourse parsing. We then re-train the model on short commonsense stories (with the model trained on news as initialization). We perform story cloze test (Mostafazadeh et al., 2017), i.e. given a four-sentence story, choose the fifth sentence from two provided options. Our joint model achieves the best known results in the unsupervised setting. In all cases, our ablation study demonstrates that each aspect of FES-LM contributes to the model.

The main contributions of our work are: 1) the design of a joint neural language model for semantic sequences built from frames, entities and sentiments; 2) showing that FES-LM trained on news is of high quality and can help to improve shallow discourse parsing; 3) achieving the state-of-the-art result on story cloze test in an unsupervised setting with the FES-LM tuned on stories.

2 Semantic Aspect Modeling

This section describes how we capture different aspects of the semantic information in a text snippet via semantic frames, entities and sentiments.

2.1 Semantic Frames

Semantic frame is defined by Fillmore (1976): *frames are certain schemata or frameworks of concepts or terms which link together as a system, which impose structure or coherence on some aspect of human experience, and which may contain elements which are simultaneously parts of other such frameworks.* In this work, we simplify it by defining a semantic frame as a composition of a predicate and its corresponding argument participants. The design of PropBank frames (Kingsbury and Palmer, 2002) and FrameNet frames (Baker et al., 1998) perfectly fits our needs. Here we require the predicate to be disambiguated to a specific sense, thus each frame can be uniquely represented by its predicate sense. These frames provide a good level of generalization as each frame can be instantiated into various surface forms in natural texts. For example, in Ex.1, the semantic frame in Opt.1 would be abstracted as "convict.01". We associate each of these frames with an embedding. The arguments of the frames are modeled as entities, as described next.

Additionally, in accordance with the idea proposed by Peng and Roth (2016), we also extend the frame representations to include discourse markers since they model relationships between frames. In this work, we only consider explicit discourse markers between abstracted frames. We use surface forms to represent discourse markers because there is only a limited set. We also assign an embedding with the same dimension as frames to each discourse marker.

Ex.4 The doctor told Susan that *she* was busy.
The doctor told Susan that *she* had cancer.
Mary told Susan that *she* had cancer.

Figure 2: **Examples of the need for different levels of entity abstraction.** For each sentence, one wants to understand what the pronoun "she" refers to, which requires different abstractions for two underlined entity choices depending on context.

To unify the representation, we formally use e_f to represent an embedding of a disambiguated frame/discourse marker. Such embedding would later be learned during language model training.

2.2 Entities

We consider the subject and object of a predicate as the essential entity information for modeling semantics. To achieve a higher level of abstraction, we model entity types instead of entity surface forms. We choose to assign entities with labels produced by Named Entity Recognition (NER), as NER typing is reliable.[1]

In fact, it is difficult to abstract each entity into an appropriate level since the decision is largely affected by context. Consider the examples shown in Figure 2. For the first sentence, to correctly understand what "she" refers to, it is enough to just abstract both entities "the doctor" and "Susan" to the NER type "person", i.e. the semantic knowledge being *person A told person B that person A was busy*. However, when we change the context in the second sentence, the "person" abstraction becomes too broad as it loses key information for this "doctor - patient" situation. The ideal semantic abstraction would be "a doctor told a patient that the patient had a disease". For the third sentence, it is ambiguous without further context from other sentences. Thus, entity abstraction is a delicate balance between specificity and correctness.

[1] Though there are a number works on fine-grained entity typing (Yogatama et al., 2015; Ren et al., 2016), their performances are between 65% and 75%, much lower than NER.

Besides type information, Ex.2 in Figure 1 shows the necessity of providing *new entity* information, i.e whether or not an entity is appeared for the first time in the whole semantic sequence. This corresponds well with the definition of *anaphrocity* in co-reference resolution, i.e. whether or not the mention starts a co-reference chain. Thus, we can encode this binary information as an additional dimension in the entity representation.

Thus, we formally define r_e as the entity representation. It is the concatenation of two entity vectors r_{sub} and r_{obj} for subject entity and object entity respectively. Both r_{sub} and r_{obj} are constructed as a one hot vector[2] to represent an entity type, plus an additional dimension indicating whether or not it is a *new entity* (1 if it is new).

2.3 Sentiments

For a piece of text, we can assign a sentiment value to it. It can either be positive, negative, or neutral. In order to decide which one is most appropriate, we first use a look-up table from word lexicons to sentiment, and then count the number of words which corresponds to positive (n_{pos}) and negative (n_{neg}) sentiment respectively. If $n_{pos} > n_{neg}$, we determine the text as positive; and if $n_{pos} < n_{neg}$, we assign the negative label; and if the two numbers equal, we deem the text as neutral. We use one hot vector for three sentiment choices, and define sentiment representation as r_s.

3 FES-LM - Joint Modeling

We present our joint model FES-LM and the neural language model implementation in this section. The joint model considers frames, entities and sentiments together to construct FES representations in order to model semantics more completely. Moreover, we build language models on top of such representations to reflect the sequential nature of semantics.

3.1 FES Representation

We propose FES-LM as a joint model to embed frame, entity and sentiment information together. Thus for each sentence/clause (specific to a frame), we can get individual representations for the frame (i.e. e_f), entity types and new entity information corresponds to subject and object of the frame (i.e. r_e), and sentiment information (i.e. r_s). Thus, we construct the FES representation as:

$$r_{\text{FES}} = e_f + W_e r_e + W_s r_s.$$

W_e, W_s are two matrices transforming entity and sentiment representations into the frame embedding space, which are added to the corresponding frame embedding. These two parameters are shared across all FES representations. During language model training, we learn frame embeddings e_f as well as W_e and W_s. An overview of the FES representation in a semantic sequence is shown in Figure 3. Note that if the frame embedding represents a discourse marker, we set the corresponding entity and sentiment representations as zero vectors since no entity/sentiment is matched to a discourse marker. It is our design choice to add the entity and sentiment vectors to the frame embeddings, which creates a unified semantic space. During training, the interactions between different semantic aspects are captured by optimizing the loss on the joint FES representations.[3]

3.2 Neural Language Model

To model semantic sequences and train FES representations, we build neural language models. Theoretically, we can utilize any existing neural language model. We choose to implement the log-bilinear language model (LBL) (Mnih and Hinton, 2007) as our main method since previous works have reported best performance using it (Rudinger et al., 2015; Peng and Roth, 2016).

For ease of explanation, we assume that a semantic sequence of FES representations is $[\text{FES}_1, \text{FES}_2, \text{FES}_3, \ldots, \text{FES}_k]$, with FES_i being the i_{th} FES representation in the sequence. It assigns each token (i.e. FES representation) with three components: a target vector $v(\text{FES})$, a context vector $v'(\text{FES})$ and a bias $b(\text{FES})$. Thus, we model the conditional probability of a token FES_t given its context $c(\text{FES}_t)$:

$$p(\text{FES}_t | c(\text{FES}_t)) = \frac{\exp(v(\text{FES}_t)^\intercal u(c(\text{FES}_t)) + b(\text{FES}_t))}{\sum_{\text{FES} \in \mathcal{V}} \exp(v(\text{FES})^\intercal u(c(\text{FES}_t)) + b(\text{FES}))}.$$

Here, $\mathcal{V}$ denotes the vocabulary (all possible FES representations) and we define

[2] Each dimension of the vector indicates an entity type (binary 0/1), and the vector contains exactly one element of 1.

[3] An alternative design choice is to concatenate the vector representations from different semantic aspects together, but we did not get better empirical results compared to our current design.

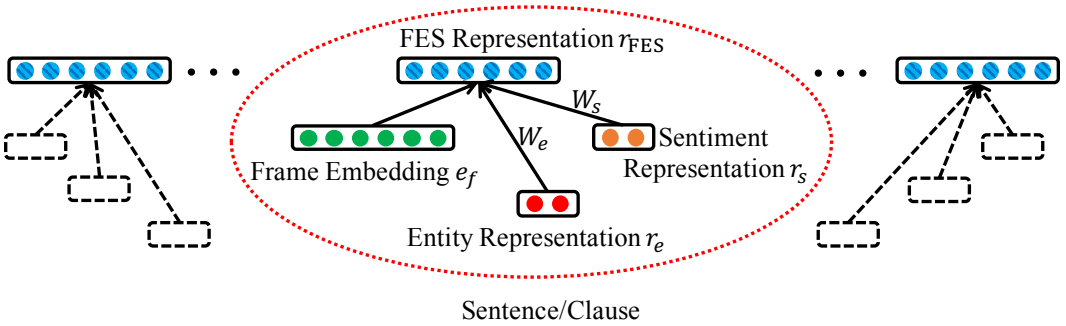

Figure 3: **An overview of the FES representation in a semantic sequence.** Semantic frames are represented by vector r_f. The entity representation r_e is the concatenation of r_{sub} and r_{obj}, both consist of two parts: an one-hot vector for entity type plus an additional dimension to indicate whether or not it is a *new entity*. The sentiment representation r_s is also one-hot.

$$u(c(\text{FES}_t)) = \sum_{c_i \in c(\text{FES}_t)} q_i \odot v'(c_i).$$

Note that $\odot$ represents element-wise multiplication and q_i is a vector that depends only on the position of an FES representation in context, which is also a model parameter. For language model training, we maximize the overall sequence probability $\prod_{t=1}^{k} p(\text{FES}_t | c(\text{FES}_t))$.

4 Building FES-LM

In this section, we explain how we build FES-LM from un-annotated plain text.

4.1 Dataset and Preprocessing

Dataset We first use the New York Times (NYT) Corpus[4] (from year 1987 to 2007) to train FES-LM. It contains over 1.8M documents in total. To fine tune the model on short stories, we re-train FES-LM on the ROCStories dataset (Mostafazadeh et al., 2017) with the model trained on NYT as initialization. We use the train set of ROCStories, which contains around 100K short stories (each consists of five sentences) [5].

Preprocessing We pre-process all documents with Semantic Role Labeling (SRL) (Punyakanok et al., 2004) and Part-of-Speech (POS) tagger (Roth and Zelenko, 1998). We also implement the explicit discourse connective identification module of a shallow discourse parser (Song et al., 2015). Additionally, we utilize within document entity co-reference (Peng et al., 2015a) to produce co-reference chains to get the *new entity* information. To obtain all annotations, we employ the Illinois NLP tools[6].

4.2 FES Representation Generation

As shown in Sec. 3, each FES representation is built from basic semantic units: frame / entity / sentiment. We describe our implementation details on how we extract these units from text and how we further construct their vector representations respectively.

Frame Abstraction and Enrichment We directly derive semantic frames from semantic role labeling annotations. As the Illinois SRL package is built upon PropBank frames, we map them to FrameNet frames via VerbNet senses to achieve a higher level of abstraction. The mapping is deterministic and partial[7]. For unmapped PropBank frames, we retain their original PropBank forms. We then enrich the frames by augmenting them to verb phrases. We apply three heuristic rules: 1) if a preposition immediately follows a predicate, we append the preposition e.g. *"take over"*; 2) if we encounter the role label AM-PRD which indicates a secondary predicate, we append it to the main predicate e.g. *"be happy"*; 3) if we see the semantic role label AM-NEG which indicates negation, we append "not" e.g. *"not like"*. We further connect compound verbs together as they represent a unified semantic meaning. For this, we apply a rule that if the gap between two predicates is less than two tokens, we treat them as a unified semantic frame defined by the conjunction of the two (augmented) semantic frames, e.g. *"decide to*

[4] Available at https://catalog.ldc.upenn.edu/LDC2008T19

[5] Available at http://cs.rochester.edu/nlp/rocstories/

[6] Available at http://cogcomp.org/page/software/

[7] We use the mapping file http://verbs.colorado.edu/verb-index/fn/vn-fn.xml to do it. For example, "place" and "put" with the same VerbNet sense id "9.1-2" are both mapped to the FrameNet frame "Placing".

	Vocabulary Size				Sequence Size	
	FES	F	E	S	#seq	#token
NYT	4M	15K	100	7	1.2M	25.4M
ROCStories	200K	1K	98	7	100K	630K

Table 2: **Statistics on FES-LM vocabularies and sequences.** We compare FES-LM trained on NYT vs. ROCStories; "FES" stands for unique FES representations while "F" for frame embeddings, "E" for entity representations, and "S" for sentiment representations. "#seq" is the number of sequences, and "#token" is the total number of tokens (FES representations) used for training.

buy" being represented by "decide.01-buy.01".

To sum up, we employ the same techniques to deal with frames as discussed in Peng and Roth (2016), which allows us to model more fine-grained semantic frames. As an example of this processing step, *"He didn't want to give up."* is represented as "(not)want.01-give.01[up]". Each semantic frame (here, including discourse markers) is represented by a 200-dimensional vector e_f.

Entity Label Assignment For each entity (here we refer to subject and object of the predicate), we first extract its syntactic head using Collins' Head Rule. To assign entity types, we then check if the head is inside a named entity generated by NER. If so, we directly assign the NER label to this entity. Otherwise, we check if the entity is a pronoun that refers to a person i.e. *I, me, we, you, he, him, she, her, they, them*; in which case, we assign "PER" label to it. For all other cases, we simply assign "ARG" label to indicate the type is unknown.

In order to assign "new entity" labels, we check if the head is inside a mention identified by the co-reference system to start a new co-reference chain. If so, we assign 1; otherwise, we assign 0. On ROCStories dataset, we add an additional rule that all pronouns indicating a person will not be "new entities". This makes the co-reference decisions more robust on short stories.[8]

The entity representation r_e is eventually constructed as a one-hot vector for types of 5 dimensions and an additional dimension for "new entity" information. As we consider both subjects and objects of a frame, r_e is of 12 dimensions in total. If either one of the entities within a frame is missing from SRL annotations, we set its corresponding 6 dimensions as zeros.

Sentiment Representation Generation We first determine the polarity of a word by a look-up table from two pre-trained sentiment lexicons (Liu et al., 2005; Wilson et al., 2005). We then count the number of positive words versus negative words to decide the sentiment of a piece of text as detailed in Sec. 2. This process is done on text corresponding to each frame, i.e. a sentence or a clause. Since we have two different lexicons, we get two separate one-hot sentiment vectors, each with a dimension of 3. Thus, the sentiment representation is the concatenation of the two vectors, a total dimension of 6.

4.3 Neural Language Model Training

For the NYT corpus, we treat each document as a single semantic sequence while on ROCStories, we see each story as a semantic sequence. Additionally, we filter out rare frames which appear less than 20 times in the NYT corpus. Statistics on the eventual FES-LM vocabularies (unique FES representations) and semantic sequences in both datasets are shown in Table 2. Note that the number of unique FES representations reflects the richness of the semantic space that we model. On both datasets, it is about 200 times over what is modeled by only frame representations. At the same time, we do not incur burden on language model training. It is because we do not model unique FES representations directly, and instead we are still operating in the frame embedding space.[9]

We use the OxLM toolkit (Baltescu et al., 2014) with Noise-Constrastive Estimation (Gutmann and Hyvarinen, 2010) to implement the LBL model. We set the context window size to 5 and produce 200-dimension embeddings for FES representations. In addition to learning language model parameters, we also learn frame embeddings e_f along with parameters for W_e (12x200 matrix) and W_s (6x200 matrix).

5 Evaluation

We first show that our proposed FES-LM is of high quality in terms of language modeling ability. We then evaluate FES-LM for shallow discourse parsing on news data as well as application for story cloze test on short common sense stories. In all studies, we verify that each semantic aspect contributes to the joint model.

[8]The same rule is not applied on news, since pronouns indicating a person can start a co-reference chain in news.

[9]The FES representation space can be seen as entity and sentiment infused frame embedding space.

	CBOW	SG	LBL
Perplexity			
FES-LM	133.8	135.8	**126.0**
Narrative Cloze Test (Recall@30)			
FES-LM	38.9	37.3	**43.2**
FES-LM - Entity	35.3	33.1	38.4
FES-LM - Sentiment	34.9	32.8	36.3

Table 3: **Quality comparison of neural language models.** We report results for perplexity and narrative cloze test. Both evaluations are done on the gold PropBank data (annotated with gold frames). LBL outperforms CBOW and SG on both tests. We carry out ablation studies for narrative cloze test for FES-LM without entity and sentiment aspects respectively.

5.1 Quality of FES-LM

To evaluate the modeling ability of different neural language models, we train each variant of FES-LM on NYT corpus and report perplexity and narrative cloze test results. Here, we choose the Skip-Gram (SG) model (Mikolov et al., 2013b) and Continuous-Bag-of-Words (CBOW) model (Mikolov et al., 2013a) for comparison with the LBL model. We utilize the word2vec package to implement both SG and CBOW. We set the context window size to be 10 for SG and 5 for CBOW.

We employ the same experimental setting as detailed in Peng and Roth (2016). Results are shown in Table 3. They confirm that LBL model performs the best with the lowest perplexity and highest recall for narrative cloze test.[10] Note that the numbers reported are not directly comparable with those in literature (Rudinger et al., 2015; Peng and Roth, 2016), as we model much richer semantics even though the numbers seem inferior. We further carry out ablation studies for narrative cloze test for FES-LM without entity and sentiment aspects respectively[11]. The results show that sentiment contributes more than entity information.

5.2 Application on News

We choose shallow discourse parsing as the task to show FES-LM's applicability on news. In particular, we evaluate on identifying the correct sense of discourse connectives (both explicit and implicit ones). We choose Song et al. (2015), which uses a supervised pipeline approach, as our base system. We follow the same experimental setting as described in Peng and Roth (2016), i.e. we add additional conditional probability features generated from FES-LM into the base system. We evaluate on CoNLL16 (Xue et al., 2016) test and blind sets, following the train and development split from the Shared Task, and report F1 using the official shared task scorer.

Table 4 shows the results for shallow discourse parsing with added FES-LM features. We get significant improvement over the base system(*) (based on McNemar's Test) and outperform SemLM, which only utilizes frame information in the semantic sequences. We also rival the top system (Mihaylov and Frank, 2016) in the CoNLL16 Shared Task (connective sense classification subtask). Note that the FES-LM used here is trained on NYT corpus. The ablation study shows that entity aspect contributes less than sentiment aspect in this application.

5.3 Application on Stories

For the story cloze test on the ROCStories dataset. We evaluate in an unsupervised setting, where we disregard the labeled development set and directly test on the test set[12]. We believe this is a better setting to reflect a system's ability to model semantic sequences compared to the supervised setting where we simply treat the task as a binary classification problem with a development set to tune.

We first generate a set of conditional probability features from FES-LM. For each story, we extract semantic aspect information as described in Sec. 2 and construct the joint FES representation according to the learned FES-LM. We then utilize the conditional probability of the fifth sentence s_5 given previous context sentences C as features. Suppose the semantic information in the fifth sentence can be represented by $r_{\text{FES\_k}}$, we can then define the features as $p(s_5|C) = p(r_{\text{FES\_k}}|r_{\text{FES\_(k-1)}}, r_{\text{FES\_(k-2)}}, \cdots, r_{\text{FES\_(k-t)}}), \quad t = 1, 2, \cdots, k$. We get multiple features depending on how long we go back in the context in terms of FES representations. Note that one sentence can contain multiple FES representations depending on how many semantic frames it has. For simplicity, we assume a single FES representation $r_{\text{FES\_k}}$

[10] We also tried Neural-LSTM (Pichotta and Mooney, 2016a) and context2vec (Melamud et al., 2016) model, but we cannot get better results.

[11] The ablation study is not done for perplexity test because FES-LM with less semantic aspects yields smaller vocabulary, which naturally leads to lower perplexity.

[12] The test set contains $1,871$ four-sentences long stories with two fifth sentence options for each, of which only one is correct; and we report the accuracy.

	CoNLL16 Test			CoNLL16 Blind		
	Explicit	Implicit	Overall	Explicit	Implicit	Overall
Base (Song et al., 2015)*	89.8	35.6	60.4	75.8	31.9	52.3
SemLM (Peng and Roth, 2016)	**91.1**	36.3	61.4	77.3	33.2	53.8
Top (Mihaylov and Frank, 2016)	89.8	**39.2**	**63.3**	78.2	**34.5**	**54.6**
FES-LM (this work)	91.0	37.5	61.8	**78.3**	34.4	54.5
FES-LM - Entity	90.8	37.1	61.6	77.9	34.0	54.1
FES-LM - Sentiment	90.5	36.9	61.3	77.3	33.8	53.9

Table 4: **Shallow discourse parsing results.** With added FES-LM features, we get significant improvement (based on McNemar's Test) over the base system(*) and outperform SemLM, which only models frame information. We also rival the top system (Mihaylov and Frank, 2016) in the CoNLL16 Shared Task (connective sense classification subtask).

Baselines		
Seq2Seq		58.0%
DSSM (Mostafazadeh et al., 2016)		58.5%
Seq2Seq with attention		**59.1%**
Individual Aspect	S.	M.V.
F-LM	57.8%	56.3%
E-LM	52.1%	52.6%
S-LM	54.2%	54.9%
Joint Model	S.	M.V.
FES-LM (this work)	**62.3%**	61.6%
FES-LM - Entity	61.5%	61.7%
FES-LM - Sentiment	61.1%	60.9%

Table 5: **Accuracy results for story cloze text in the unsupervised setting.** "S." represents the inference method with the single most informative feature while "M.V." means majority voting. FES-LM outperforms the strongest baseline (Seq2Seq with attention) by 3 points. The difference is statistically significant based on McNemar's Test. Additional ablation studies show that each semantic aspect contributes to the joint model.

for s_5. In practice, we get at most 12 FES representations as context. We align the features by t, indicating how long we consider the story context. Thus, for each story, we generate at most 12 pairs of conditional probability features. Evey pair of such features can yield a decision on which ending is more probable. Here, we test two different inference methods: a single most informative feature (where we go with the decision made by the pair of features which have the highest ratio) or majority voting based on all feature pairs. Note that we need to re-train FES-LM on the stories (train set of ROCStories, 5-sentence stories, no negative examples provided)[13].

We compare FES-LM with Seq2Seq baselines (Sutskever et al., 2014). We also train the Seq2Seq model on the train set of ROCStories, where we set input as the 4-sentence context and the output as the 5th ending sentence for each story. At test time, we get probability of each option ending from the soft-max layer and choose the higher one as the answer. We use an LSTM encoder (300 hidden units) and decode with an LSTM of the same size. Since it is operated on the word level, we use pre-trained 300-dimensional GloVe embeddings (Pennington et al., 2014) and keep them fixed during training. In addition, we add an attention mechanism (Bahdanau et al., 2014) to make the Seq2Seq baseline stronger. We also report DSSM from Mostafazadeh et al. (2016) as the previously best reported result[14]. To study how each individual aspect affects the performance, we develop neural language models on frames (F-LM), entities (E-LM) and sentiments (S-LM) as additional baseline models separately. We use the same language model training and feature generation techniques as FES-LM. Particularly, for F-LM, it is the same model as FC-SemLM defined in Peng and Roth (2016). Note that individual aspects cannot capture the semantic difference between two given options for all instances. For those instances that the baseline model fails to handle, we set the accuracy as 50% (expectation of random guesses).

The accuracy results are shown in Table 5. The best result we achieve (62.3%) outperforms the strongest baseline (Seq2Seq with attention, 59.1%). It is statistically significant based on McNemar's Test ($\alpha = 0.01$), illustrating the superior

[13]It is because of domain difference, e.g. average length of semantic sequence is different (stories are shorter while news are longer, see in Table 2).

[14]DSSM's model parameters are trained on the ROCStories corpus while hyper parameters are determined on the development set.

semantic modeling ability of FES-LM. Results are mixed comparing the two inference methods. The ablation study further confirms that each semantic aspect has its worth in the joint model.

6 Related Work

Our work is built upon the previous work (Peng and Roth, 2016). It generated a probabilistic model on semantic frames while taking into account discourse information, and showed applications to both co-reference resolution and shallow discourse parsing. This line of work is in general inspired by script learning. Early works (Schank and Abelson, 1977; Mooney and DeJong, 1985) tried to learn scripts via construction of knowledge bases from text. More recently, researchers focused on utilizing statistical models to extract high-quality scripts from large amounts of data (Chambers and Jurafsky, 2008a; Bejan, 2008; Jans et al., 2012; Pichotta and Mooney, 2014; Granroth-Wilding et al., 2015; Rudinger et al., 2015; Pichotta and Mooney, 2016c,a,b). Other works aimed at learning a collection of structured events (Chambers, 2013; Cheung et al., 2013; Balasubramanian et al., 2013; Bamman and Smith, 2014; Nguyen et al., 2015; Inoue et al., 2016). In particular, Ferraro and Van Durme (2016) presented a unified probabilistic model of syntactic and semantic frames while also demonstrating improved coherence. Several works have employed neural embeddings (Modi and Titov, 2014b,a; Frermann et al., 2014; Titov and Khoddam, 2015). Some prior works have used scripts-related ideas to help improve NLP tasks (Irwin et al., 2011; Rahman and Ng, 2011; Peng et al., 2015b). Most recently, Mostafazadeh et al. (2016, 2017) proposed story cloze test as a standard way to test a system's ability to model semantics. They released ROCStories dataset, and organized a shared task for LSDSem'17; which yields many interesting works on this task. Cai et al. (2017) developed a model that uses hierarchical recurrent networks with attention to encode sentences and produced a strong baseline.

7 Conclusion

This paper proposes FES-LM, a joint neural language model for semantic sequences built upon frames, entities and sentiments. Abstractions on these semantic aspects enable FES-LM to generate better semantic sequences than models working on the word level. Evaluations show that the joint model helps to improve shallow discourse parsing and achieves the best result for story cloze test in the unsupervised setting. In future work, we plan to extend FES-LM to capture more semantic aspects and work towards building a general semantic language model.

Acknowledgments

This work is supported by the US Defense Advanced Research Projects Agency (DARPA) under contract HR0011-15-2-0025, and by the Army Research Laboratory (ARL) under agreement W911NF-09-2-0053, and also by IBM-ILLINOIS Center for Cognitive Computing Systems Research (C3SR) - a research collaboration as part of the IBM Cognitive Horizon Network. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

References

D. Bahdanau, K. Cho, and Y. Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* .

C. F. Baker, C. J. Fillmore, and J. B. Lowe. 1998. The berkeley framenet project. In *COLING/ACL*.

N. Balasubramanian, S. Soderland, O. E. Mausam, and O. Etzioni. 2013. Generating coherent event schemas at scale. In *EMNLP*.

P. Baltescu, P. Blunsom, and H. Hoang. 2014. Oxlm: A neural language modelling framework for machine translation. *The Prague Bulletin of Mathematical Linguistics* .

D. Bamman and N. A. Smith. 2014. Unsupervised discovery of biographical structure from text. *TACL* .

C. A. Bejan. 2008. Unsupervised discovery of event scenarios from texts. In *FLAIRS Conference*.

Z. Cai, L. Tu, and K. Gimpel. 2017. Pay attention to the ending: Strong neural baselines for the ROC story cloze task. In *ACL*.

N. Chambers. 2013. Event schema induction with a probabilistic entity-driven model. In *EMNLP*.

N. Chambers and D. Jurafsky. 2008a. Jointly combining implicit constraints improves temporal ordering. In *EMNLP*.

N. Chambers and D. Jurafsky. 2008b. Unsupervised learning of narrative event chains. In *ACL*.

S. B. Chatman. 1980. *Story and discourse: Narrative structure in fiction and film*. Cornell University Press.

J. C. K. Cheung, H. Poon, and L. Vanderwende. 2013. Probabilistic frame induction. *arXiv:1302.4813* .

F. Ferraro and B. Van Durme. 2016. A unified bayesian model of scripts, frames and language. In *AAAI*.

C. J. Fillmore. 1976. Frame semantics and the nature of language. *Annals of the New York Academy of Sciences* .

L. Frermann, I. Titov, and Pinkal. M. 2014. A hierarchical bayesian model for unsupervised induction of script knowledge. In *EACL*.

M. Granroth-Wilding, S. Clark, M. T. Llano, R. Hepworth, S. Colton, J. Gow, J. Charnley, N. Lavrač, M. Žnidaršič, and M. Perovšek. 2015. What happens next? event prediction using a compositional neural network model. In *AAAI*.

M. Gutmann and A. Hyvarinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*.

N. Inoue, Y. Matsubayashi, M. Ono, N. Okazaki, and K. Inui. 2016. Modeling context-sensitive selectional preference with distributed representations. In *COLING*.

J. Irwin, M. Komachi, and Y. Matsumoto. 2011. Narrative schema as world knowledge for coreference resolution. In *CoNLL Shared Task*.

B. Jans, S. Bethard, I. Vulić, and M. F. Moens. 2012. Skip n-grams and ranking functions for predicting script events. In *EACL*.

P. Kingsbury and M. Palmer. 2002. From Treebank to PropBank. In *Proceedings of LREC-2002*.

B. Liu, M. Hu, and J. Cheng. 2005. Opinion observer: Analyzing and comparing opinions on the web. In *WWW*.

O. Melamud, J. Goldberger, and I. Dagan. 2016. context2vec: Learning generic context embedding with bidirectional lstm. In *CoNLL*.

T. Mihaylov and A. Frank. 2016. Discourse relation sense classification using cross-argument semantic similarity based on word embeddings. In *CoNLL Shared Task*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* .

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *NAACL*.

A. Mnih and G. Hinton. 2007. Three new graphical models for statistical language modelling. In *ICML*.

A. Modi and I. Titov. 2014a. Inducing neural models of script knowledge. In *CoNLL*.

A. Modi and I. Titov. 2014b. Learning semantic script knowledge with event embeddings. In *ICLR Workshop*.

R. Mooney and G. DeJong. 1985. Learning schemata for natural language processing. In *IJCAI*.

N. Mostafazadeh, N. Chambers, X. He, D. Parikh, D. Batra, L. Vanderwende, P. Kohli, and J. Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *NAACL*.

N. Mostafazadeh, M. Roth, A. Louis, N. Chambers, and J. F. Allen. 2017. Lsdsem 2017 shared task: The story cloze test. In *LSDSEM workshop at EACL*.

K.-H. Nguyen, X. Tannier, O. Ferret, and R. Besançon. 2015. Generative event schema induction with entity disambiguation. In *ACL*.

H. Peng, K. Chang, and D. Roth. 2015a. A joint framework for coreference resolution and mention head detection. In *CoNLL*.

H. Peng, D. Khashabi, and D. Roth. 2015b. Solving hard coreference problems. In *NAACL*.

H. Peng and D. Roth. 2016. Two discourse driven language models for semantics. In *ACL*.

J. Pennington, R. Socher, and C. D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

K. Pichotta and R. J. Mooney. 2014. Statistical script learning with multi-argument events. In *EACL*.

K. Pichotta and R. J. Mooney. 2016a. Learning statistical scripts with lstm recurrent neural networks. In *AAAI*.

K. Pichotta and R. J. Mooney. 2016b. Statistical script learning with recurrent neural networks. In *EMNLP*.

K. Pichotta and R. J. Mooney. 2016c. Using sentence-level lstm language models for script inference. In *ACL*.

V. Punyakanok, D. Roth, W. Yih, and D. Zimak. 2004. Semantic role labeling via integer linear programming inference. In *COLING*.

A. Rahman and V. Ng. 2011. Coreference resolution with world knowledge. In *ACL*.

X. Ren, W. He, M. Qu, L. Huang, H. Ji, and J. Han. 2016. Afet: Automatic fine-grained entity typing by hierarchical partial-label embedding. In *EMNLP*.

D. Roth and D. Zelenko. 1998. Part of speech tagging using a network of linear separators. In *ACL-COLING*.

R. Rudinger, P. Rastogi, F. Ferraro, and B. Van Durme. 2015. Script induction as language modeling. In *EMNLP*.

R. C. Schank and R. P. Abelson. 1977. Scripts, plans, goals, and understanding: An inquiry into human knowledge structures. In *JMZ*.

Y. Song, H. Peng, P. Kordjamshidi, M. Sammons, and D. Roth. 2015. Improving a pipeline architecture for shallow discourse parsing. In *CoNLL Shared Task*.

I. Sutskever, O. Vinyals, and Q. V. Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*.

I. Titov and E. Khoddam. 2015. Unsupervised induction of semantic roles within a reconstruction-error minimization framework. In *NAACL*.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *EMNLP*.

N. Xue, H. T. Ng, A. Rutherford, B. Webber, C. Wang, and H. Wang. 2016. Conll 2016 shared task on multilingual shallow discourse parsing. *CoNLL* .

Dani Yogatama, Daniel Gillick, and Nevena Lazic. 2015. Embedding methods for fine grained entity type classification. In *ACL*.