Dipanjan Das, Nathan Schneider,
Desai Chen, Noah A. Smith

Probabilistic Frame-Semantic Parsing

- Introduction
- Recources and Task
- Target Identification
- Frame Identification
- Argument Identification
- Conclusion

Introduction

This paper contributes a formalization of frame-semantic parsing as a structure prediction problem and describes an implemented parser that <u>transforms an English sentence into a frame-semantic representation</u>. It finds words that evoke FrameNet frames, selects frames for them, and locates the arguments for each frame.

Resources and Task

- Resources
- Task
- Baseline

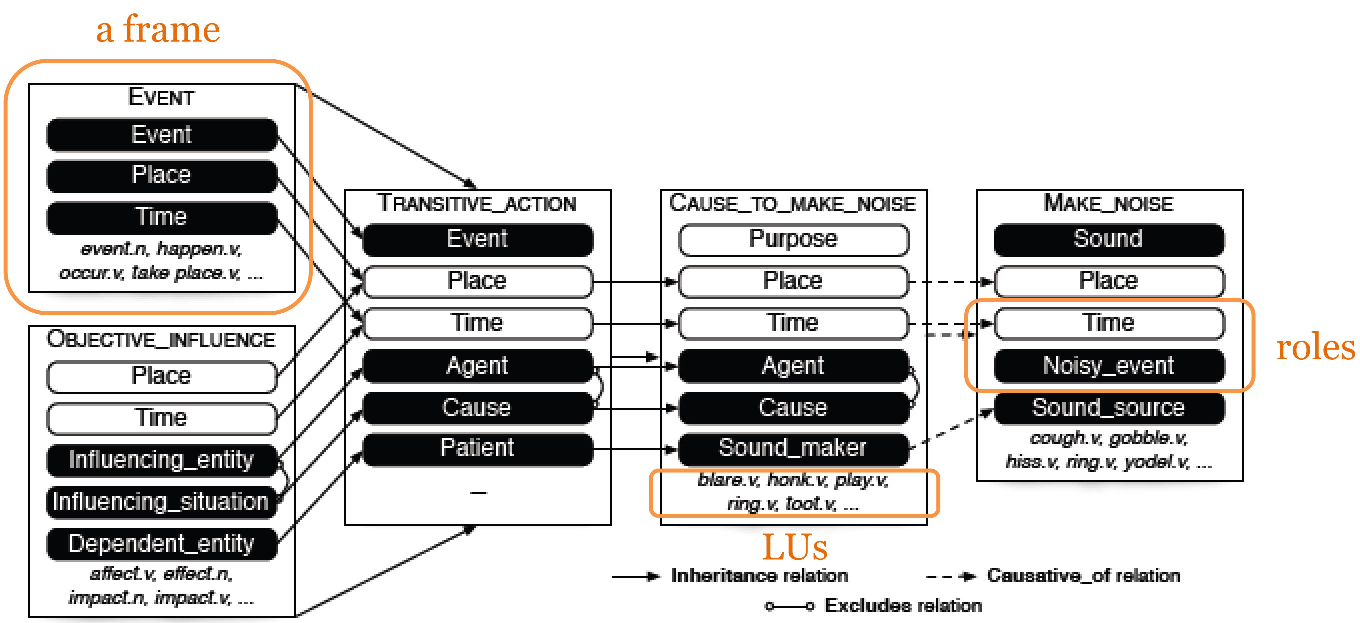
The Main Resource Used: FrameNet Lexicon

FrameNet Lexicon is a taxonomy of manually identified general-purpose frames for English. It contains:

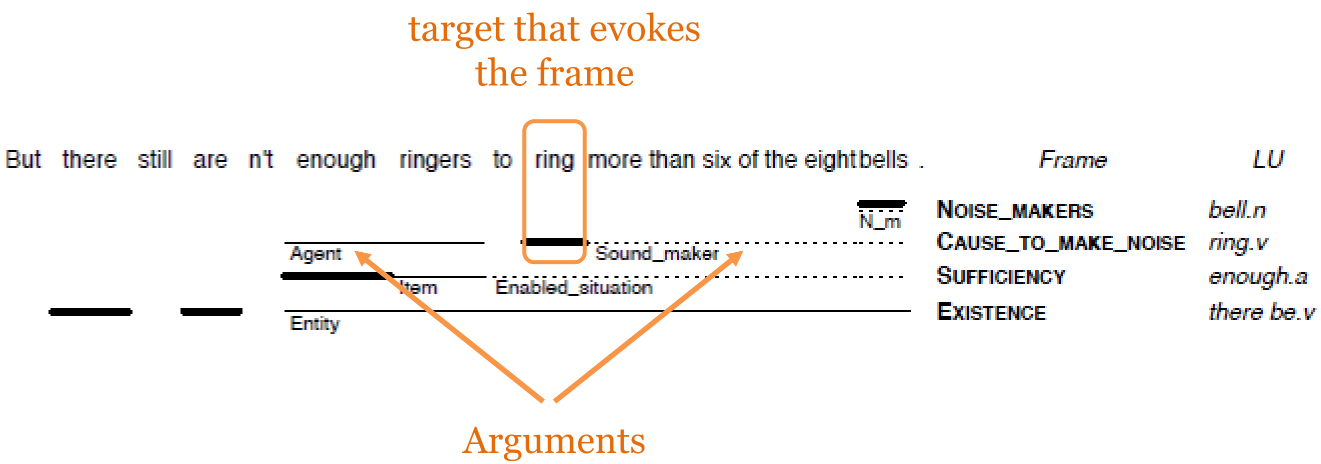
- **Lexical Units (LUs)**: lemmas listed in lexicons with frames that can denote the frame or some aspect of it.
- **Targets**: words or phrase tokens that evoke a frame.
- **Roles**: frame elements corresponding to different aspects of the concept represented by the frame.
- **Arguments**: sequences of word tokens annotated as filling frame roles.
- Information about relations between frames and between roles.

In most cases, the definition of a frame is accompanied with a set of **exemplar sentences** in the FrameNet.

FrameNet Lexicon: Examples



FrameNet Lexicon: Examples



FrameNet Lexicon: Snapshot

Framenet Lexicon v. 1.3		
lexical entries	**exemplars**	
	counts	**coverage**
8379 LUs	139K sentences, 3.1M words	70% LUs
795 frames	1 frame annotation / sentence	63% frames
7124 roles	285K overt arguments	56% roles

Snapshot of lexicon entries and exemplar sentences. Coverage indicates the fraction of types attested in at least one exemplar. The lexicon associates an average of 12.8 LUs with a frame, and 66% of those LUs are attested for that frame. The average ambiguity of an LU is 1.2 frames

Data from SemEval'07

Full-text Annotations	*SemEval'07 data* train			dev			test		
Size	*(words sentences documents)*								
all	43.3K	1.7K	22	6.3K	251	4	2.8K	120	3
ANC (travel)	3.9K	154	2	.8K	32	1	1.3K	67	1
NTI (bureaucratic)	32.2K	1.2K	15	5.5K	219	3	1.5K	53	2
PropBank (news)	7.3K	325	5	0	0	0	0	0	0
Annotations	*(frames/word overt arguments/word)*								
all	0.23	0.39		0.22	0.37		0.37	0.65	
ANC	0.22	0.38		0.15	0.29		0.37	0.60	
NTI	0.23	0.40		0.23	0.37		0.38	0.69	
PropBank	0.22	0.37							
Coverage of lexicon	*(% frames % roles % LUs)*								
all	64.1	27.4	21.0	34.0	10.2	7.3	29.3	7.7	4.9
ANC	26.4	7.4	4.8	8.9	2.0	1.1	17.5	3.9	2.3
NTI	52.4	21.1	14.9	31.5	9.2	6.7	19.0	5.0	3.0
PropBank	40.8	12.0	8.4						
Out-of-lexicon types	*(frames roles LUs)*								
all	14	69	71	2	4	2	39	99	189
ANC	12	39	41	0	0	2	26	63	123
NTI	6	32	33	2	4	0	19	45	70
PropBank	3	11	3						
Out-of-lexicon tokens	*(% frames % roles % LUs)*								
all	0.7	0.9	1.1	1.0	0.4	0.2	9.8	11.2	25.3
ANC	3.2	4.2	7.6	0.0	0.0	1.8	11.5	13.5	34.8
NTI	0.6	0.6	0.5	1.1	0.4	0.0	8.5	9.4	17.4
PropBank	0.3	0.4	0.2						

Data Preprocessing

- POS tags from MXPOST (Ratnaparkhi, 1996)
- Dependency parses from the MST parser (McDonald et al., 2005)
- WordNet for lemmantization
- Code from the SRL system (Johansson and Nugues, 2008) for verb labelling

Task

Annotations of frame-semantic structure:

- Targets, the words or phrases that evoke frames;
  - *target identification*
- The frame type, defined in the lexicon and evoked by each target;
  - *frame identification*
- Arguments (or spans of words), serve to fill roles defined by each evoked frame.
  - *argument identification*

Baseline

- Known frame-evoking words: use an SVM classifier to disambiguate frames.
- Unknown words: use WordNet synsets to extend the vocabulary of frame-evoking words , and then used a collection of separate SVM classifiers to predict a single evoked frame for each occurrence of a word in the extended set.

Target Identification

- Purpose
- Choosing Candidates
- Results

Purpose

To solve the problem of deciding which word tokens (or word token sequences) evoke frames in a given sentence.

Problems and Solution

Problems:

- In frame semantics, verbs, nouns, adjectives, prepositions, etc. can all evoke frames, making simple POS criteria un-useful.
- Because the test set is more completely annotated *(which means more frames per token in test set)*, learned models did not generalize well and achieved poor test recall.

Solution: A small set of rules is used to identify targets.

Choosing Candidates

- For a span to be a candidate target, it must appear as a target in the training data or the lexicon.
- Considering multiword targets.
- Pruning all prepositions, but keeping support verbs.

This is a conservative approach that will never propose a target that was not seen in the training data or FrameNet.

Results

Target Identification	Precision	Recall	*F1*
This technique	89.92	70.79	79.21
Baseline: J&N'07	87.87	67.11	76.10

The Next Step

Given targets, the parser next identifies their frames.

Frame Identification

- Lexical Units
- Model
- Training
- Results

Lexical Units

Frame-evoking lexical units, the words and phrases listed in the lexicon as referring to specific frames, are used for frame identification.

Words in frame-evoking LUs are lemmatized and POS-tagged.

All targets in the exemplar sentences, and most in our training and test data, correspond to known LUs.

Via a latent variable, the frame identification model will incorporate features based on exemplar and training targets rather than LUs, to incorporate frame-evoking expressions found in the training data and to avoid lemmatization errors.

Sets of Targets: $\mathcal{L}$, $\mathcal{L}_f$, $\mathcal{L}^l$ and $\mathcal{L}_f^l$

$\mathcal{L}$: the set of (unlemmatized and POS-tagged) targets found in the exemplar sentences of the lexicon and/or the sentences in our training set.

$\mathcal{L}_f$: the subset ($\mathcal{L}_f \subset \mathcal{L}$) of these targets annotated as evoking a particular frame f.

$\mathcal{L}^l$ and $\mathcal{L}_f^l$ denote the lemmatized versions of $\mathcal{L}$ and $\mathcal{L}_f$.

A Probabilistic Model

To identify candidate frames, a prediction rule is firstly needed. It requires a probabilistic model over frames for a target:

$$f_i \leftarrow \underset{f \in \mathcal{F}_i}{\mathrm{argmax}} \sum_{\ell \in \mathcal{L}_f} p(f, \ell \mid t_i, \mathrm{x})$$

A Probabilistic Model

$$f_i \leftarrow \underset{f \in \mathcal{F}_i}{\operatorname{argmax}} \sum_{\ell \in \mathcal{L}_f} p(f, \ell \mid t_i, \mathrm{x})$$

x: given sentence

t_i: the ith frame-evoking target (its lemma is denoted as t_i^l)

f: a particular frame (which is being scored)

$\mathcal{F}_i$: the set of candidate frames for the ith target in x

ℓ: a prototype consisting of the words and POS tags of a target seen in an exemplar or training sentence as evoking f. $\ell = \langle w_\ell, \pi_\ell \rangle$ (w_ℓ: the sequence of words; π_ℓ: their part-of-speech tags)

A Probabilistic Model

Using this model, we seek a list $\mathrm{f} = \langle f_1, \ldots, f_m \rangle$ of frames, one per target. In this model, the set of candidate frames for t_i is defined to include every frame f such that $t_i^l \in \mathcal{L}_f^l$ — or if $t_i^l \notin \mathcal{L}^l$, then every known frame.

A Conditional Log-linear Model

We adopt a conditional log-linear model:

for $f \in \mathcal{F}_i$ and $\ell \in \mathcal{L}_f$,

$$p_\theta(f, \ell \mid t_i, \mathrm{x}) = \frac{\exp \theta^\top g(f, \ell, t_i, \mathrm{x})}{\sum_{f' \in \mathcal{F}_i} \sum_{\ell' \in \mathcal{L}_{f'}} \exp \theta^\top g(f', \ell', t_i, \mathrm{x})}$$

θ: the model weights

g: a vector-valued feature function

This discriminative formulation is very flexible, allowing for a variety of (possibly overlapping) features.

Training

Given the training data (*which is of the form* $\langle\langle \mathrm{x}^{(\mathrm{j})}, t^{(\mathrm{j})}, f^{(\mathrm{j})}, \mathcal{A}^{(\mathrm{j})}\rangle\rangle_{\mathrm{j}=1}^{\mathrm{N}}$ *(N is the number of sentences)*), the frame identification model is trained discriminatively by maximizing the following log-likelihood:

$$\max_{\theta} \sum_{j=1}^{N} \sum_{i=1}^{m_j} \log \sum_{\ell \in \mathcal{L}_{f_i^{(j)}}} p_\theta(f_i^{(j)}, \ell \mid t_i^{(j)}, \mathrm{x}^{(j)})$$

Results

Given gold-standard targets, the model is able to predict frames for lemmas not seen in training.

Together, the target and frame identification outperform the baseline by 4 F_1 points.

It is an improvement upon formal works because it requires only a single model.

Argument Identification

- Model
- Training
- Results

The Task

Given sentence x $= \langle x_1, \dots, x_n \rangle$, set of targets t $= \langle t_1, \dots, t_m \rangle$, list of evoked frames f $= \langle f_1, \dots, f_m \rangle$ corresponding to each target, the task is to choose which of each f_i's roles are filled, and by which parts of x.

Roles and Core Roles

Let $\mathcal{R}_{f_i} = \{r_1, \ldots, r_{|\mathcal{R}_{f_i}|}\}$ denote f_i's roles observed in an exemplar sentence and/or the training set.

Those roles conceptually and/or syntactically necessary for any use of the frame are marked as core roles.

$\mathcal{S}$: Set of Spans

We identify a set $\mathcal{S}$ of spans that are candidates for filling any role r $\in \mathcal{R}_{f_i}$. Here, it contains spans that:

- contain a single word,
- comprise a valid sub-tree of a word and its descendants in the dependency parse,
- is empty.

In training, if a labeled argument is not a valid sub-tree of the dependency parse, we add its span to $\mathcal{S}$.

Mapping of the Roles

Let $\mathcal{A}_i$ denote the mapping of roles in $\mathcal{R}_{f_i}$ to spans in $\mathcal{S}$.

The model makes a prediction for each $\mathcal{A}_i(r_k)$ (for all roles $r_k \in \mathcal{R}_{f_i}$) using

$$\mathcal{A}_i(r_k) \leftarrow \text{argmax}_{s \in \mathcal{S}}\, p(s \mid r_k, f_i, t_i, x)$$

Another Conditional Log-linear Model

This model is used over spans for each role of each evoked frame:

$$p_\psi(\mathcal{A}_i(r_k) = s \mid f_i, t_i, \mathrm{x}) = \frac{\exp \psi^\top \mathrm{h}(s, r_k, f_i, t_i, \mathrm{x})}{\sum_{s' \in \mathcal{S}} \exp \psi^\top \mathrm{h}(s', r_k, f_i, t_i, \mathrm{x})}$$

Features for this model depend on:

- the preprocessed sentence x;
- the target t;
- a role r of frame f;
- a candidate argument span $s \in \mathcal{S}$.

Another Conditional Log-linear Model

This model chooses the span for each role separately from the other roles and ignores all frames except the frame the role belongs to.

Training

Train the argument identification model by:

$$\max_{\psi} \sum_{j=1}^{N} \sum_{i=1}^{m_j} \sum_{k=1}^{\left|\mathcal{R}_{f_i^{(j)}}\right|} \log p_{\psi}(\mathcal{A}_i^{(j)}(r_k) \mid f_i^{(j)}, t_i^{(j)}, \mathrm{x}^{(j)})$$

Training this model until the argument identification F_1 score stops increasing on the development data.

Results

ARGUMENT IDENTIFICATION	targets	frames	spans	decoding	exact frame matching P	R	F_1	partial frame matching P	R	F_1
Argument identification (oracle spans)	*	*	*	naïve	86.61	75.11	80.45			
	*	*	*	beam §5.3	88.29	74.77	80.97			
Argument identification (full)	*	*	model §5	naïve	77.43	60.76	68.09			
	*	*	model §5	beam §5.3	78.71	60.57	68.46			
Parsing (oracle targets)	*	model §4	model §5	beam §5.3	49.68	42.82	46.00	57.85	49.86	53.56
Parsing (full)	auto §3	model §4	model §5	beam §5.3	**58.08**	**38.76**	**46.49**	**62.76**	**41.89**	**50.24**
Baseline: J&N'07	*auto*	*model*	*model*	*N/A*	*51.59*	*35.44*	*42.01*	*56.01*	*38.48*	*45.62*

- first 4 rows: the argument identification task
  Local model: 87% precision and 75% recall.

- When the heuristically-built candidate argument set replaces the set of true argument spans, the recall drops ~15 points, which seems to be largely due to syntactic parse errors.
  The decrease in precision indicates that the model has trouble discriminating between good and bad arguments.

- last two rows: comparing the full model with the baseline.

Conclusion

This paper offers a supervised model for rich frame-semantic parsing, based on a combination of knowledge from FrameNet, two probabilistic models trained on SemEval'07 data, and expedient heuristics.

It achieves improvements over the state of the art at each stage of processing.

Thank You