Neural semi-Markov CRF for Monolingual Word Alignment

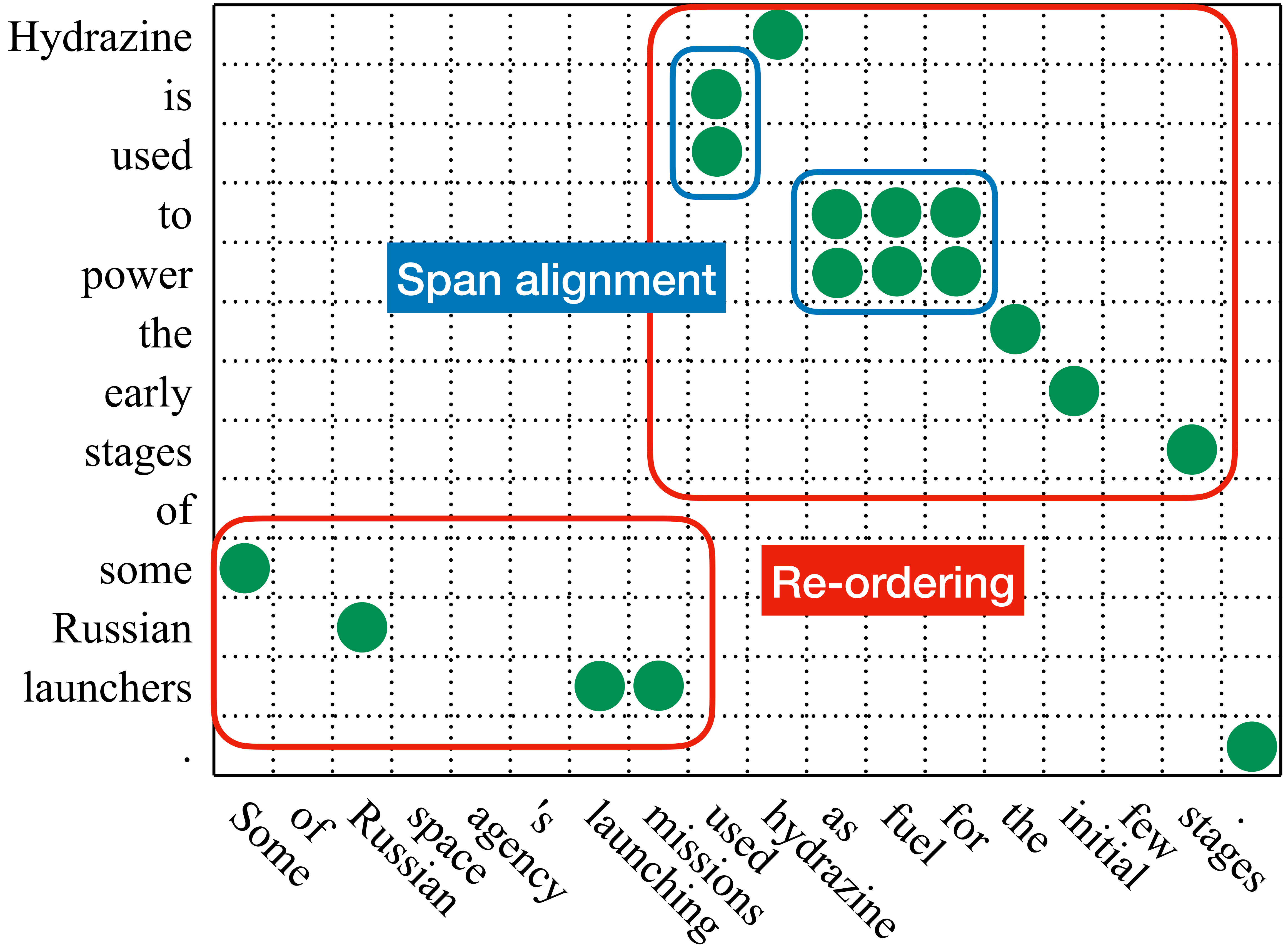
Wuwei Lan*, **Chao Jiang*** and Wei Xu

Georgia Tech
School of Interactive Computing
College of Computing

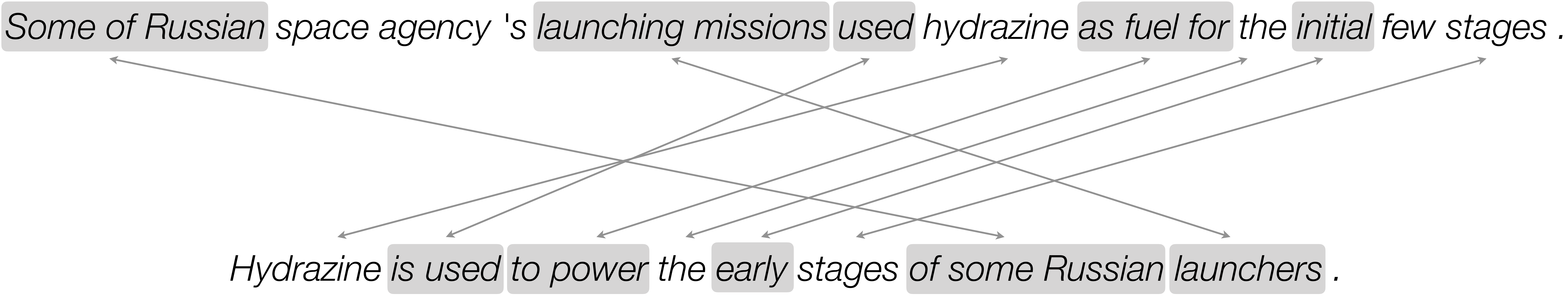
Monolingual Word Alignment is Challenging

- Aims to align words or phrases with similar meaning in two sentences in the same language.
- Spans may not follow linguistic boundaries, and annotation is expensive and time-consuming.



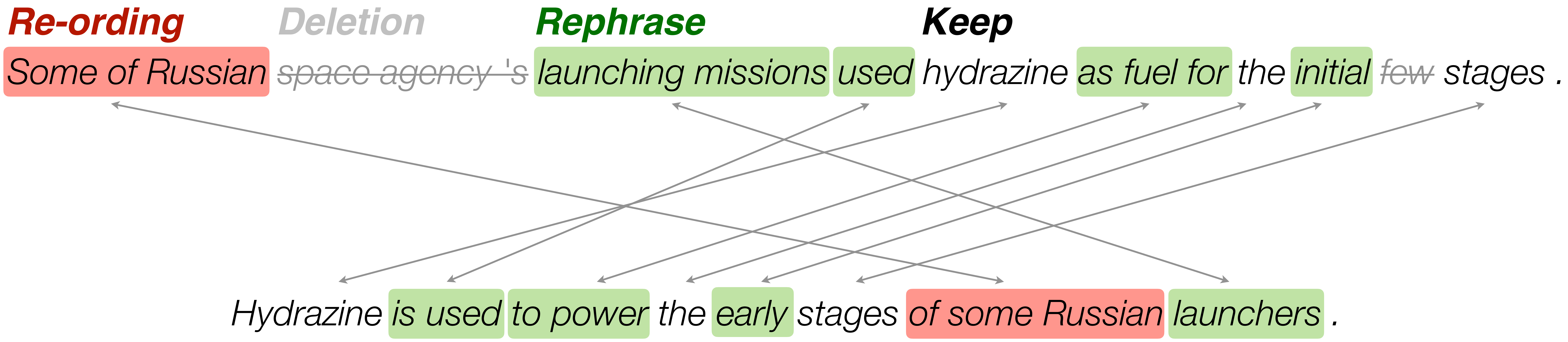
Utility of Monolingual Word Alignment

- Support the analysis of human editing operations.



Utility of Monolingual Word Alignment

- Support the analysis of human editing operations.



Utility of Monolingual Word Alignment

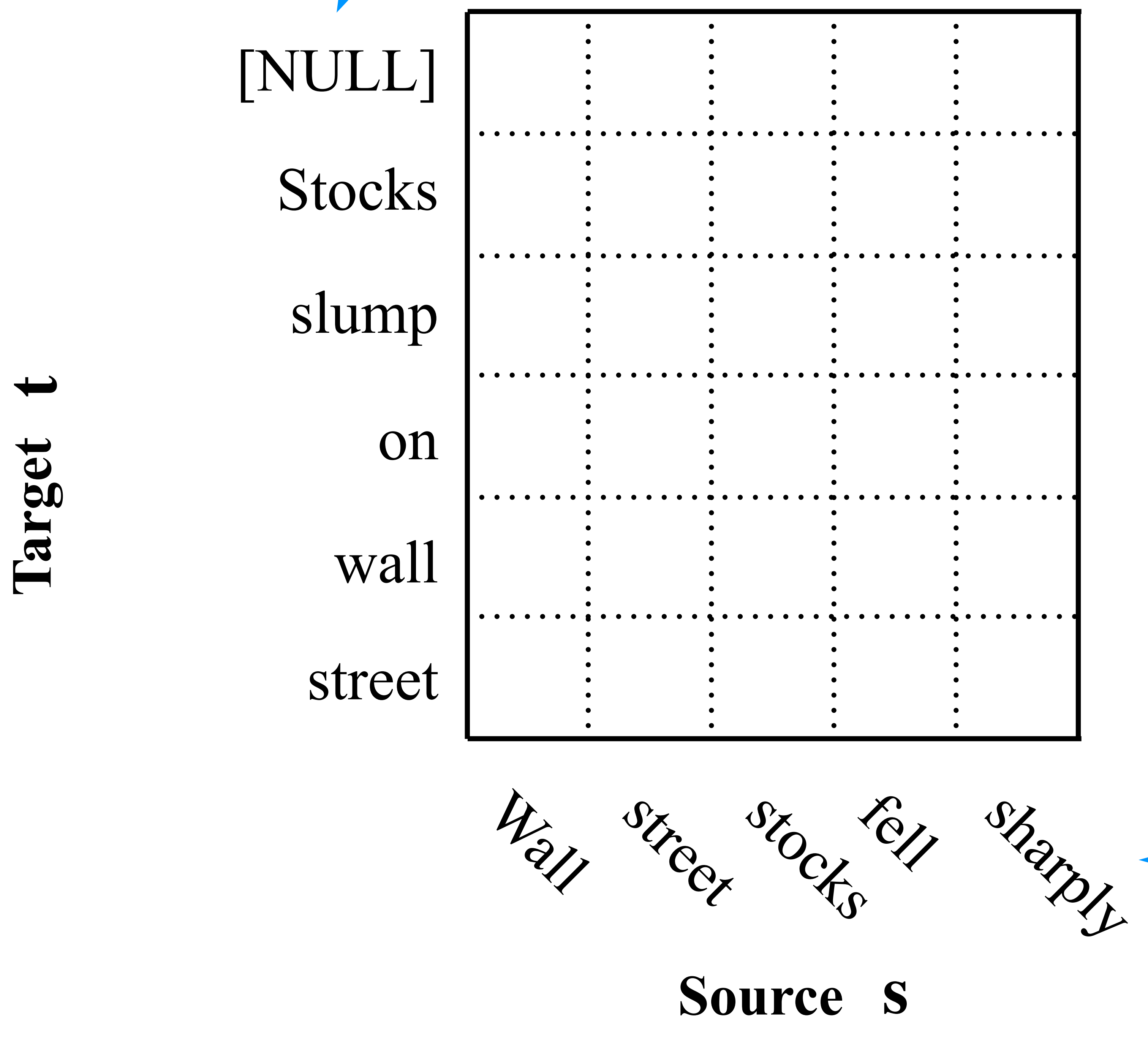
- Support the analysis of human editing operations.
- Improve the performance of text-to-text generation tasks (this work).
- Improve the interpretability of NLU tasks (SemEval 2016 on iSTS).
- Used for label projection and data argumentation (Culkin el at., 2021).

Our Solution for Monolingual Word Alignment

- Neural semi-markov CRF Alignment Model
  - Unify word and phrase alignment by variable-length spans.
  - 92.4 F1 on in-domain evaluation.
- A Multi-genre Monolingual Word Alignment Benchmark
  - Covers 4 different genres (MTRef, Wiki, news, scientific writing).
  - 9k sentence pairs annotated by in-house annotators.

Semi-CRF Alignment Model

Tokens in target sentence are labels. [NULL] label means no alignment.



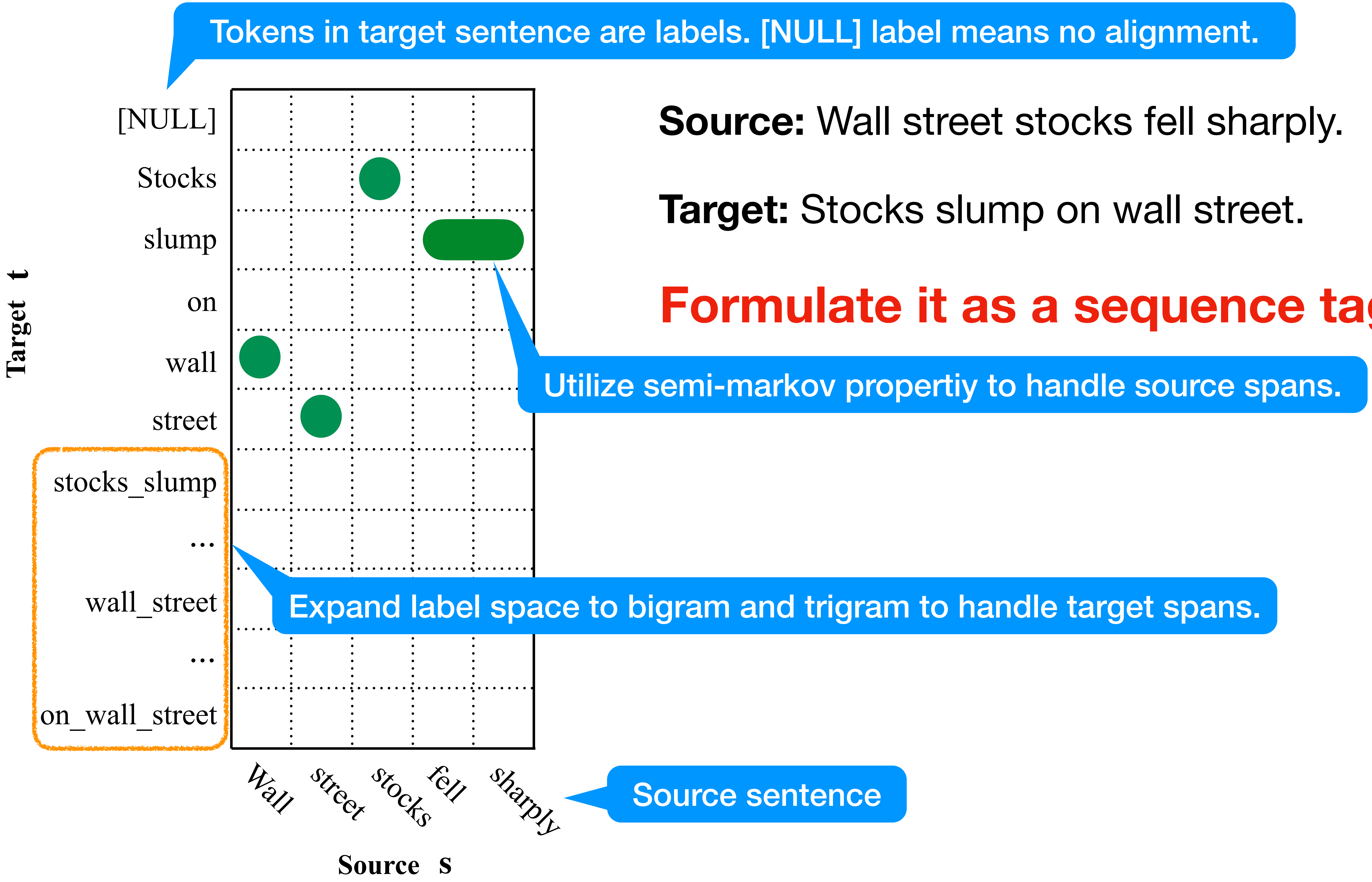
Source sentence

Source: Wall street stocks fell sharply.

Target: Stocks slump on wall street.

Formulate it as a sequence tagging problem.

Semi-CRF Alignment Model



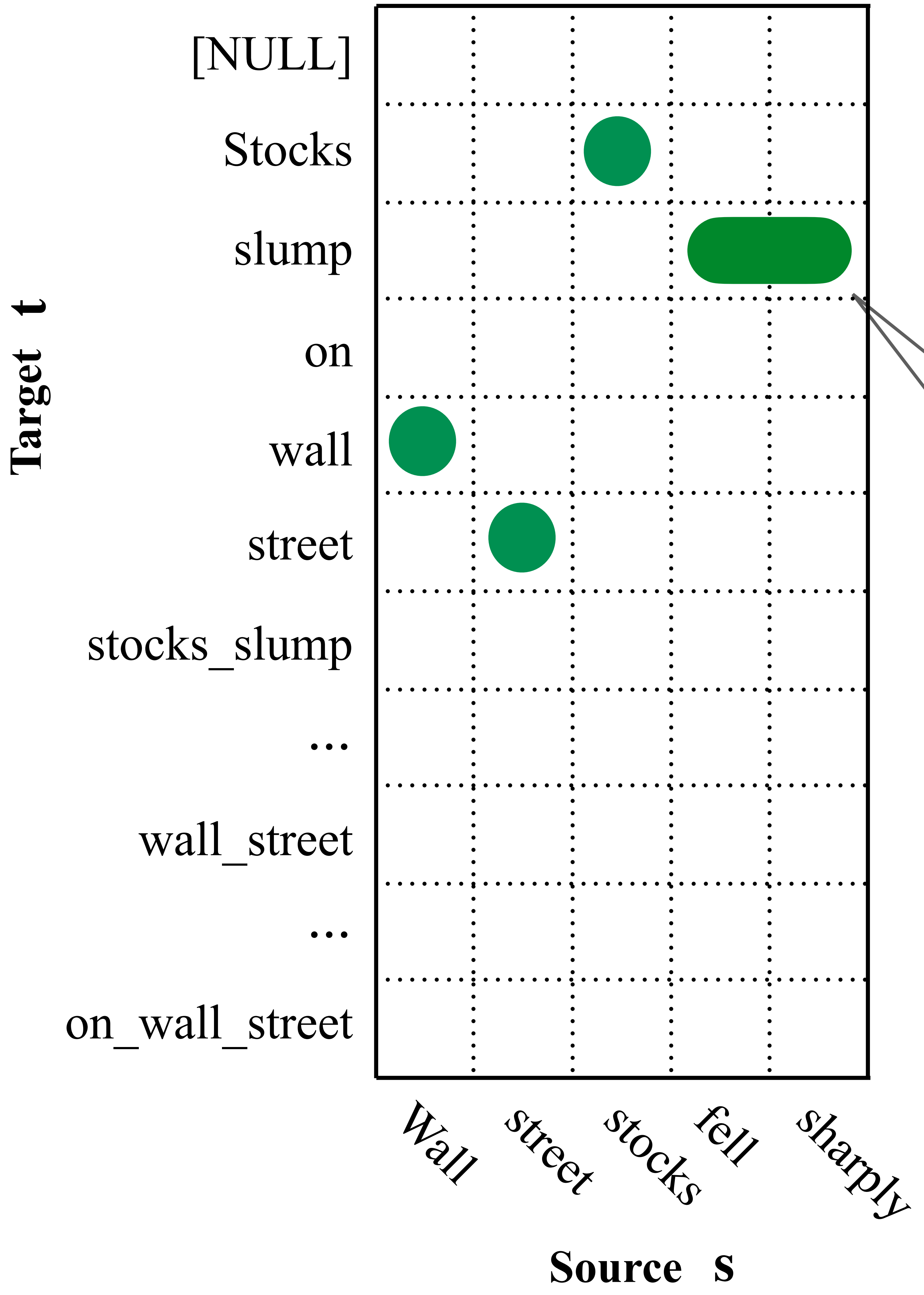
Source: Wall street stocks fell sharply.

Target: Stocks slump on wall street.

Formulate it as a sequence tagging problem.

Semi-CRF Alignment Model

Span Interaction Matrix



Span representation based on SpanBERT

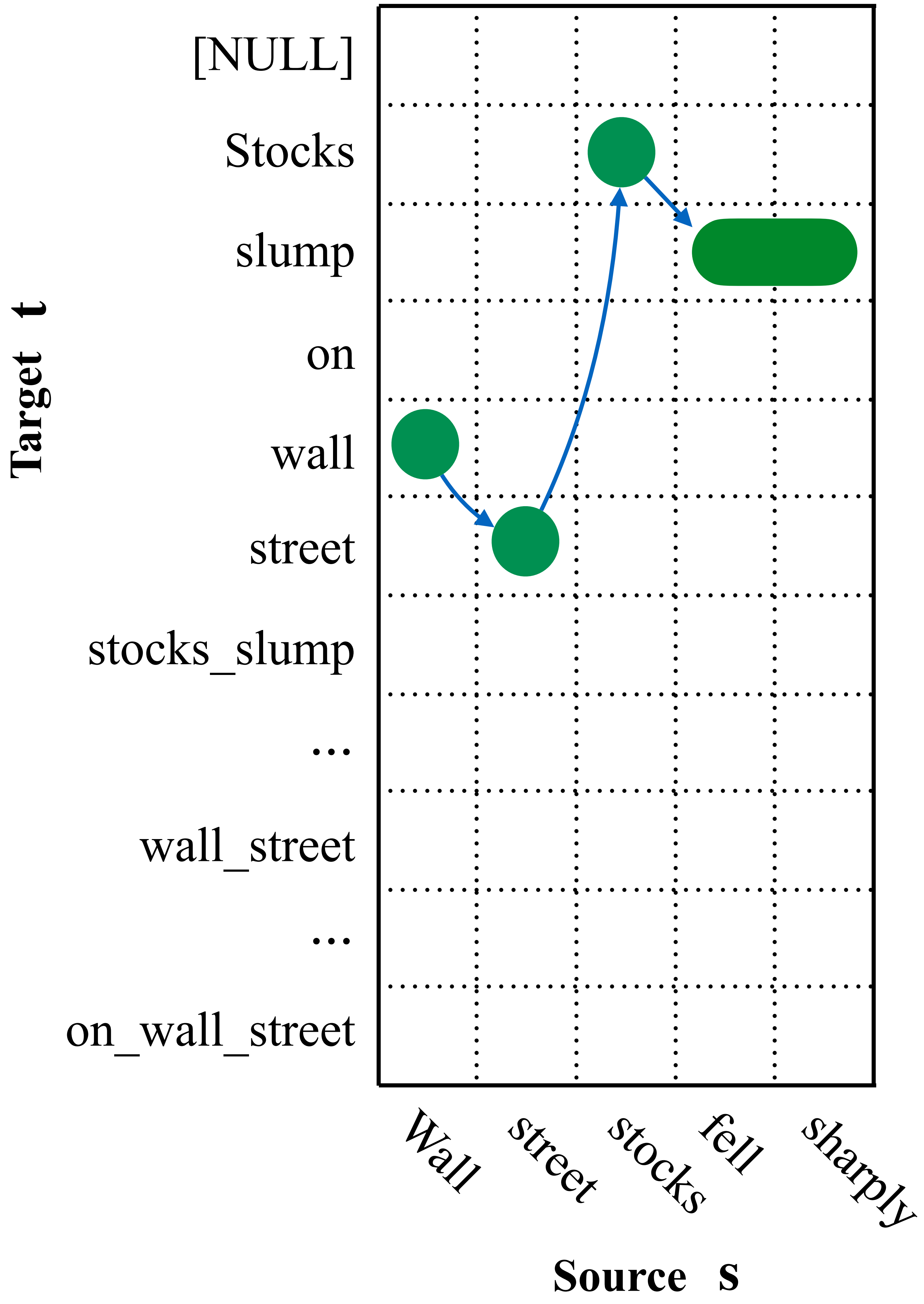
$$h_i^s = (e_{start(i)}; e_{end(i)}; att_i)$$

$$score(s_i, t_j) = \mathrm{FFNN}(h_i^s; h_j^t; |h_i^s - h_j^t|; h_i^s \circ h_j^t)$$

2-layer FFNN to capture semantic similarity between (s_i, t_j)

Semi-CRF Alignment Model

Alignment Label Transition



semi-Markov Conditional Random Fields for span alignment

$$\Psi(\mathbf{a}, \mathbf{s}, \mathbf{t}) = \sum_i score(s_i, t_{a_i}) + T(a_{i-1}, a_i) + cost(\mathbf{a}, \mathbf{a}^*)$$

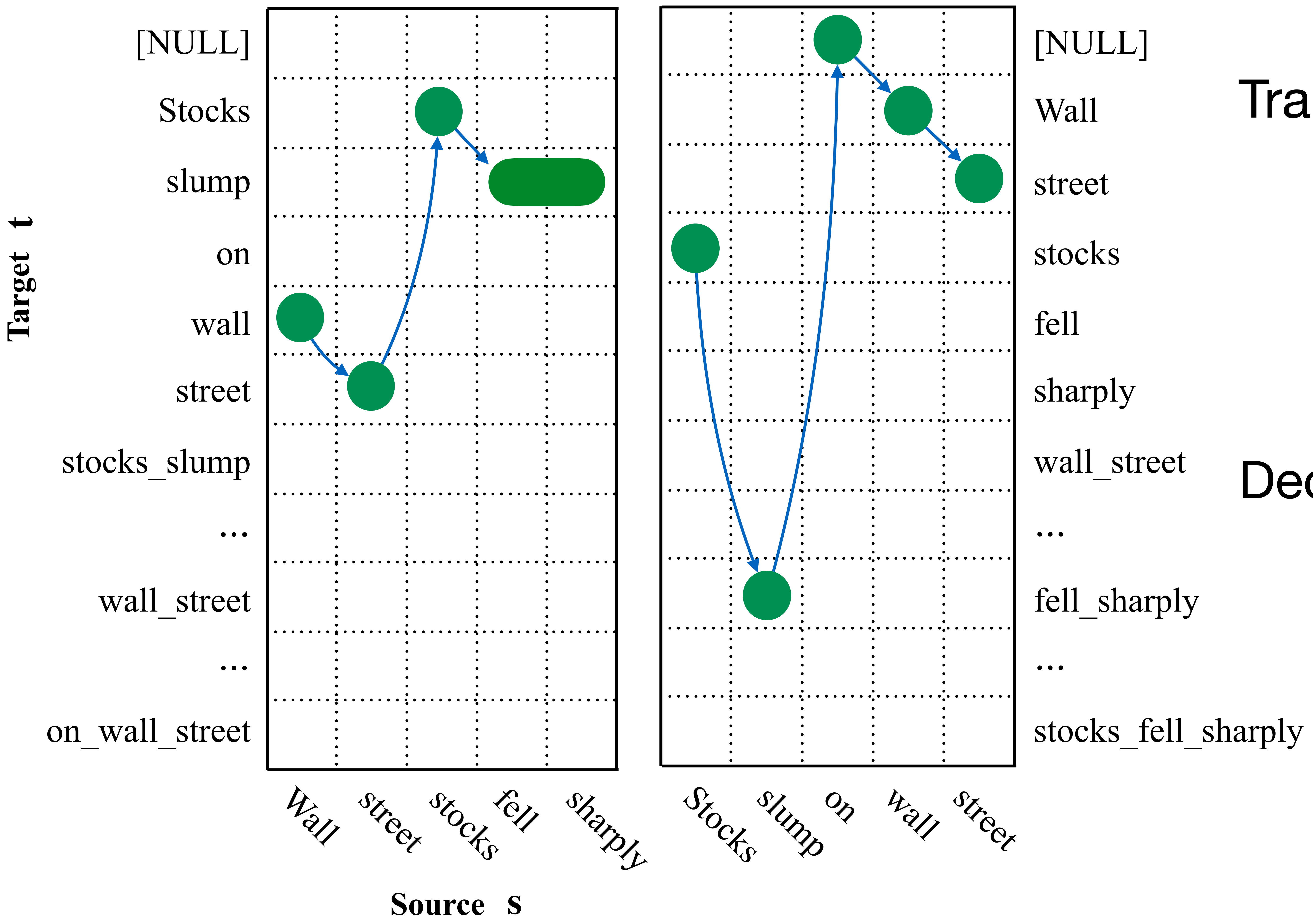
Negative Log-likelihood Loss **Hamming Loss**

$$P(\mathbf{a} \mid \mathbf{s}, \mathbf{t}) = \frac{exp\left(\Psi(\mathbf{a}, \mathbf{s}, \mathbf{t})\right)}{\sum_{\mathbf{a} \in \mathbf{A}} exp\left(\Psi(\mathbf{a}, \mathbf{s}, \mathbf{t})\right)}$$

all possible alignments

Semi-CRF Alignment Model

Bi-directional Training / Decoding



Training objective:

$$\sum_{\mathbf{s,t,a}} - logP(\mathbf{a}_{s2t} \,|\, \mathbf{s}, \mathbf{t}) - logP(\mathbf{a}_{t2s} \,|\, \mathbf{t}, \mathbf{s})$$

Source-to-target **Target-to-source**

Decoding:

Viterbi-like Algorithm + Intersect + Expand

To handle spans longer than 3

Multi-Genre Monolingual Word Alignment Benchmark

Annotated by in-house annotators, covers four different domains and the largest to data.

MultiMWA	Size	Length	%word/phrase	%identical/non-id	Genre
MTRef	3,998	22 / 17	62.0 / 38.0	52.6 / 47.3	News
Wiki	4,099	30 / 29	95.6 / 4.4	94.1 / 5.9	Wikipedia
Newsela	500	27 / 23	74.6 / 25.4	67.1 / 32.9	News
arXiv	200	29 / 28	96.6 / 3.4	93.4 / 6.6	Scientific writing
Total	8,797	26 / 23	79.3 / 20.7	73.8 / 26.2	All above

Experiments on MultiMWA Benchmark

Achieves SOTA performance on both in-domain and out-of-domain evaluation.

	In-domain	Out-of-domain		
	MTReference	Newsela	arXiv	Wikipedia

Experiments on MultiMWA Benchmark

Achieves SOTA performance on both in-domain and out-of-domain evaluation.

	In-domain	Out-of-domain		
	MTReference	Newsela	arXiv	Wikipedia
JacanaToken (Yao et al. 2013a)	76.2	79.8	95.8	95.8
JacanaPhrase (Yao et al. 2013b)	75.8	79.4	93.7	94.9
PipelineAligner (Sultan et al. 2014)	74.8	80.3	96.5	97.1

Experiments on MultiMWA Benchmark

Achieves SOTA performance on both in-domain and out-of-domain evaluation.

	In-domain	Out-of-domain		
	MTReference	Newsela	arXiv	Wikipedia
JacanaToken (Yao et al. 2013a)	76.2	79.8	95.8	95.8
JacanaPhrase (Yao et al. 2013b)	75.8	79.4	93.7	94.9
PipelineAligner (Sultan et al. 2014)	74.8	80.3	96.5	97.1
Our Neural CRF aligner	90.8	86.6	95.7	97.0

Experiments on MultiMWA Benchmark

Achieves SOTA performance on both in-domain and out-of-domain evaluation.

	In-domain	Out-of-domain		
	MTReference	Newsela	arXiv	Wikipedia
JacanaToken (Yao et al. 2013a)	76.2	79.8	95.8	95.8
JacanaPhrase (Yao et al. 2013b)	75.8	79.4	93.7	94.9
PipelineAligner (Sultan et al. 2014)	74.8	80.3	96.5	97.1
Our Neural CRF aligner	90.8	86.6	95.7	97.0
Our Neural semi-CRF aligner	92.4	87.2	97.3	97.4
	🚀16.2 F1	🚀6.9 F1	🚀0.8 F1	🚀0.3 F1

Text Simplification: EditNTS* + Our Aligner

Word alignment can help to explicitly learn edit operations (addition, deletion and paraphrase).

Deletion ***Paraphrase***

With Canadian collaborators, Lloyd performed laboratory simulations of his model.

Lloyd performed successful laboratory experiments of his model.

Addition

Used edit labels derived from our aligner for training:

DEL, DEL, DEL, KEEP, KEEP, ADD('successful'), KEEP, REPLACE-S, ADD('experiments'), REPLACE-E, KEEP, KEEP, KEEP

New Labels!

* EditNTS: An Neural Programmer-Interpreter Model for Sentence Simplification through Explicit Editing, Dong et al., ACL 2019

Text Simplification Experiments

Our aligner improves the SOTA text simplification model - EditNTS on two benchmark datasets.

	Newsela-Auto				Wiki-Auto			
	SARI	add	keep	del	SARI	add	keep	del
Complex (input)	11.8	0.0	35.5	0.0	24.0	0.0	74.6	0.0
Simple (reference)	86.9	84.7	78.4	97.6	81.7	66.2	97.5	81.5
EditNTS (Dong et al. 2019)	36.6	1.1	32.9	75.7	36.8	**2.1**	68.4	39.8
EditNTS + Aligner	**37.5**	**1.3**	**33.4**	**77.9**	**37.4**	1.9	**69.5**	**40.9**

Take Away

- Neural semi-markov CRF Alignment Model
  - Unify word and phrase alignment by variable-length spans.
  - 92.4 F1 on in-domain evaluation.
- A Multi-genre Monolingual Word Alignment Benchmark
  - Covers 4 different genres (MTRef, Wiki, news, scientific writing).
  - 9k sentence pairs annotated by in-house annotators.

Trained model / code / data available at
github.com/chaojiang06/neural-Jacana