THE OHIO STATE UNIVERSITY

CSE 5525: Foundations of Speech and Language Processing

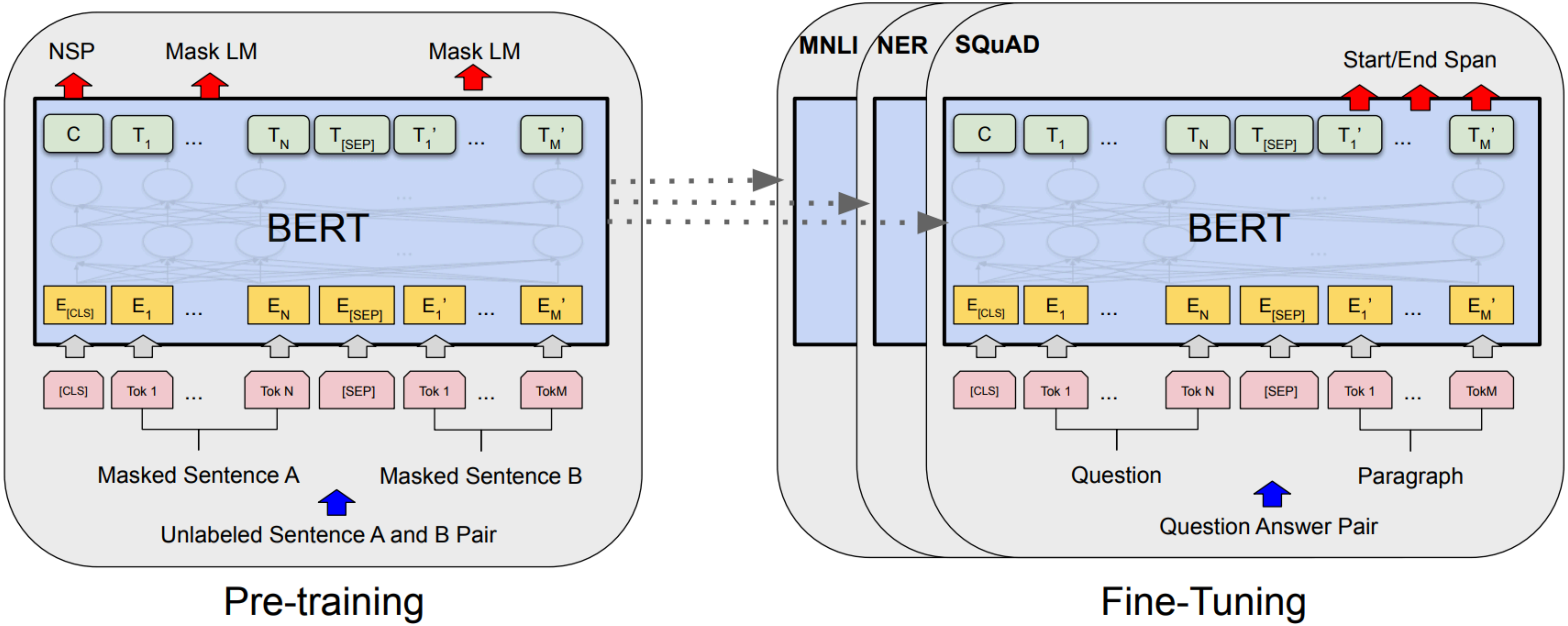
Pre-trained Models & Machine Translation

Huan Sun (CSE@OSU)

Slides were largely adapted from Prof. Greg Durrett @ UT Austin.

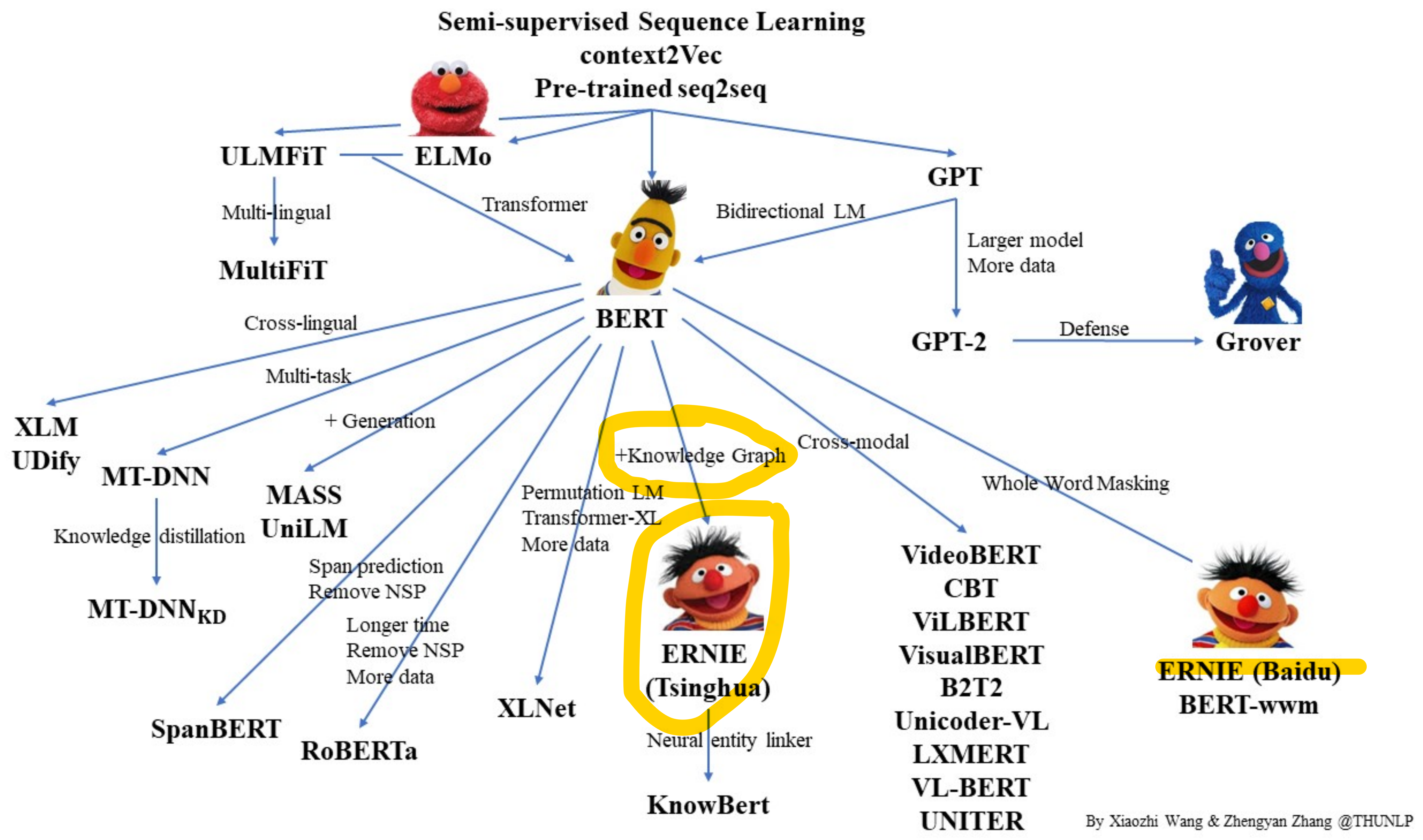
Pre-training

Pre-training on text data



NSP: next sentence prediction
LM: language model

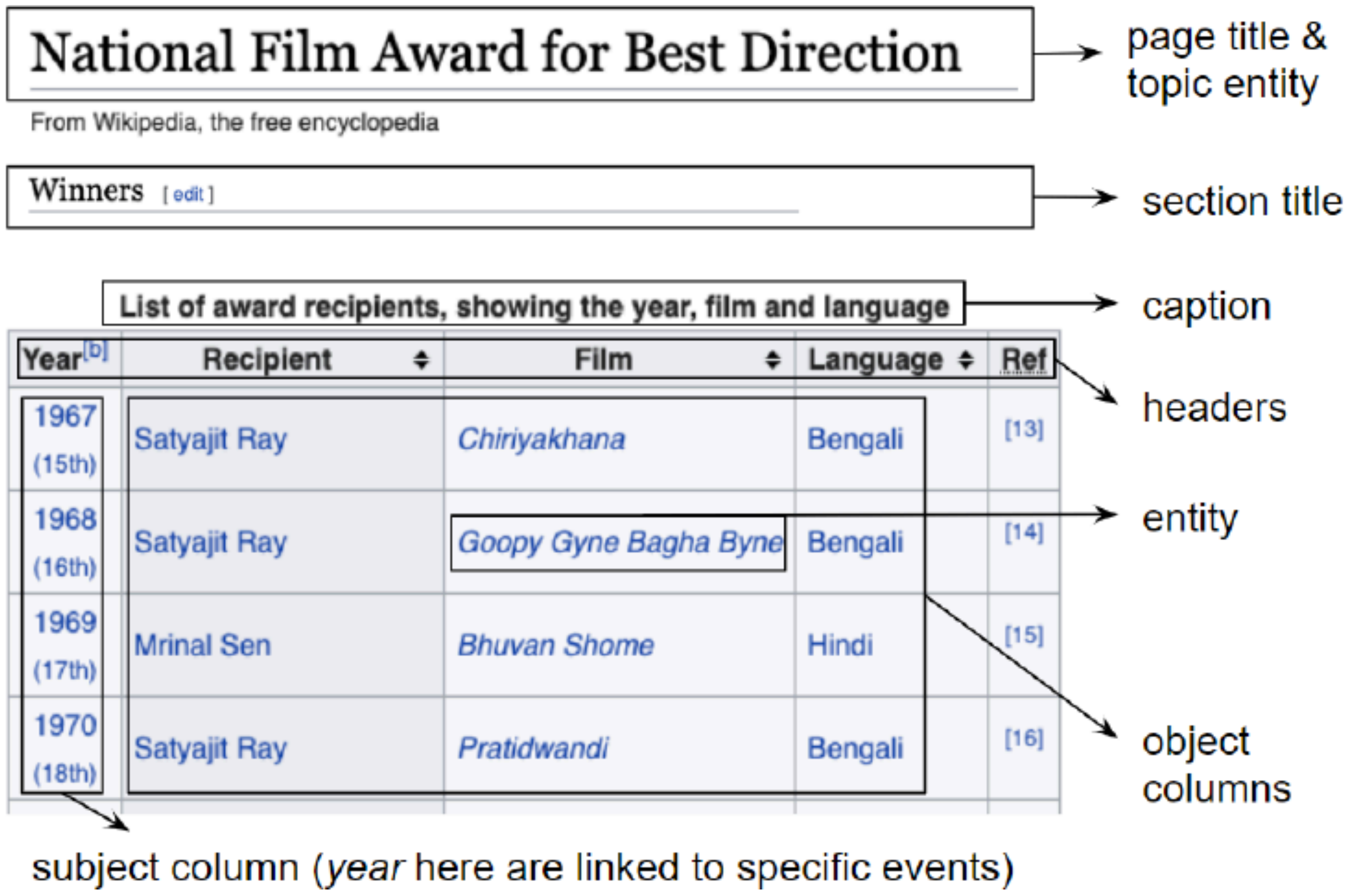
Devlin et al., 2019
https://arxiv.org/abs/1810.04805



Pretrained language models

https://github.com/thunlp/PLMpapers

TURL: Table Understanding via Representation Learning

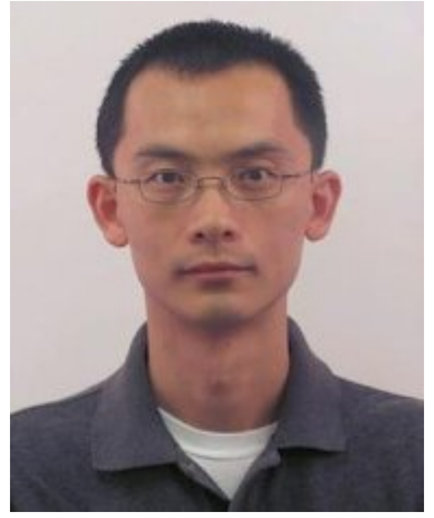


A pre-training/fine-tuning paradigm to relational Web tables and test the general model on 6 table-focused tasks [VLDB'21]

Xiang Deng (work done at OSU)

Alyssa Lees (Google)

Will Wu (Google)



Cong Yu (Google)

How about semi-structured data, like relational web tables?

List of largest technology companies by revenue

From Wikipedia, the free encyclopedia

Companies are ranked by total revenues for their respective fiscal years ended on or before March 31, 2019.[1] All data in the table is taken from the Fortune Global 500 list of technology sector companies for 2019[2] unless otherwise specified.

2019 list [edit]

Rank		Company	Fiscal year ending	Revenue ($B) USD	Employees	Headquarters
1		Apple Inc.	2019	$265.595	132,000	Cupertino, California, US
2		Samsung Electronics	2019	$221.579	309,630	Suwon, South Korea
3		Foxconn	2019	$175.617	667,680	New Taipei City, Taiwan
4		Alphabet Inc.	2019	$136.819	98,771	Mountain View, California, US
5		Microsoft	2019	$110.360	131,000	Redmond, Washington, US
6		Huawei	2019	$109.030	188,000	Shenzhen, China
7		Dell Technologies	2019	$90.621	157,000	Round Rock, Texas, US
8		Hitachi	2019	$85.507	295,941	Tokyo, Japan
9		IBM	2019	$79.591	381,100	Armonk, New York, US
10		Sony	2019	$78.157	114,400	Tokyo, Japan
11		Panasonic	2019	$72.178	271,869	Osaka, Japan
12		Intel	2019	$70.848	107,400	Santa Clara, California, US
13		HP Inc.	2019	$58.472	55,000	Palo Alto, California, US
14		Facebook Inc.	2019	$55.838	35,587	Menlo Park, California, US
15		LG Electronics	2019	$55.757	72,600	Seoul, South Korea
16		Lenovo Group	2019	$51.037	57,000	Quarry Bay, Hong Kong[4]

1. https://en.wikipedia.org/wiki/List_of_largest_technology_companies_by_revenue
2. Michael J. Cafarella, Alon Halevy, Daisy Zhe Wang, Eugene Wu, and Yang Zhang. 2008. WebTables: exploring the power of tables on the web. Proc. VLDB Endow. 1, 1 (August 2008), 538–549.
3. Cafarella, Michael J., et al. "Uncovering the Relational Web." WebDB. 2008.

Challenges

1. How to model the table meta-data and table body at the same time?

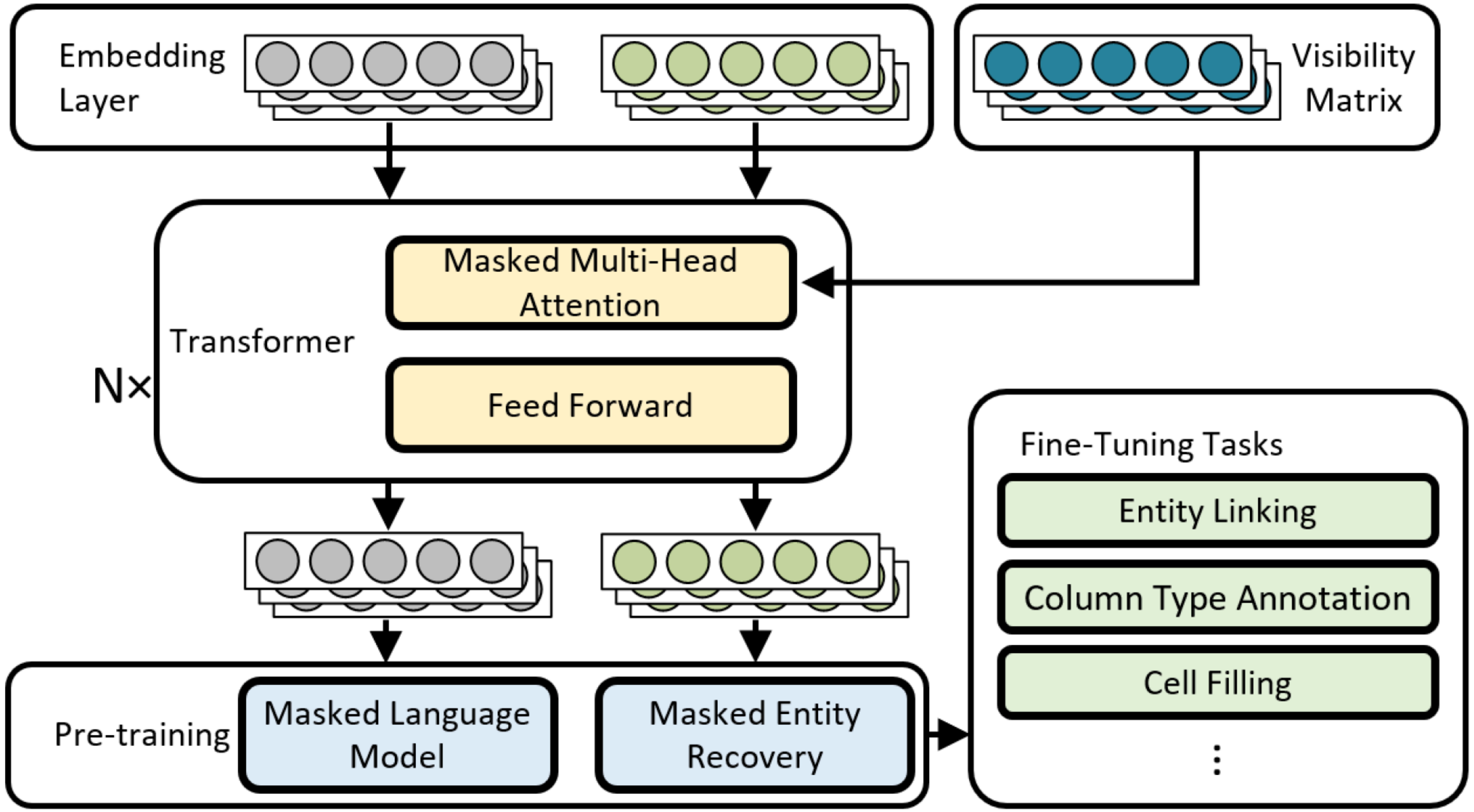
2. How to model the row-and-column structure in the table?

3. How to learn the relational knowledge in Web tables in pre-training?

4. How to effectively apply the pre-trained model in downstream tasks?

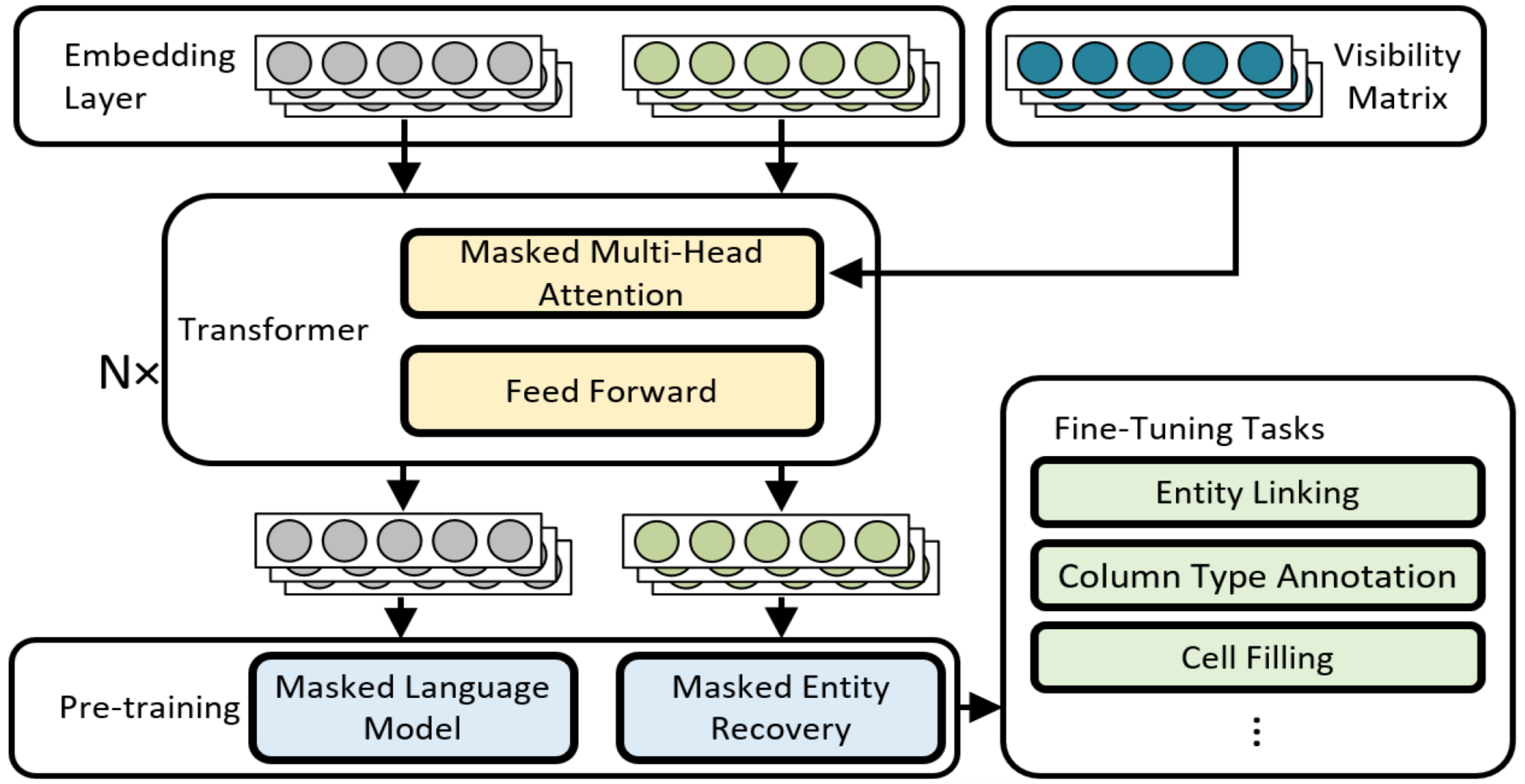
TURL: Table Understanding via Representation Learning

1. A structure-aware transformer for table encoding
2. Two pretraining objectives for learning factual knowledge from web tables

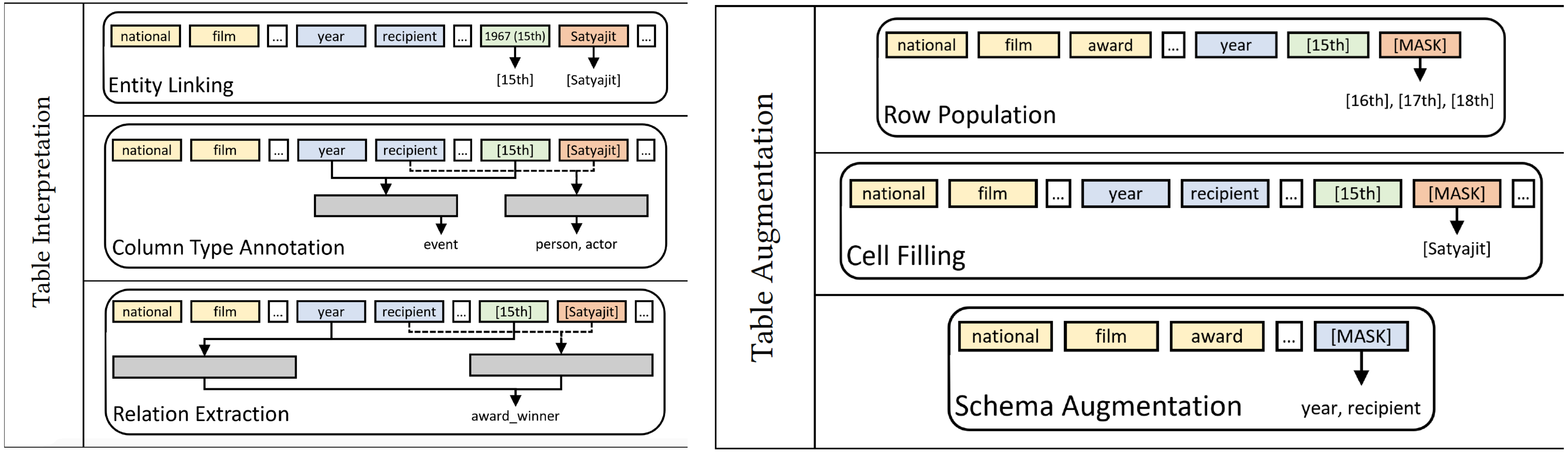


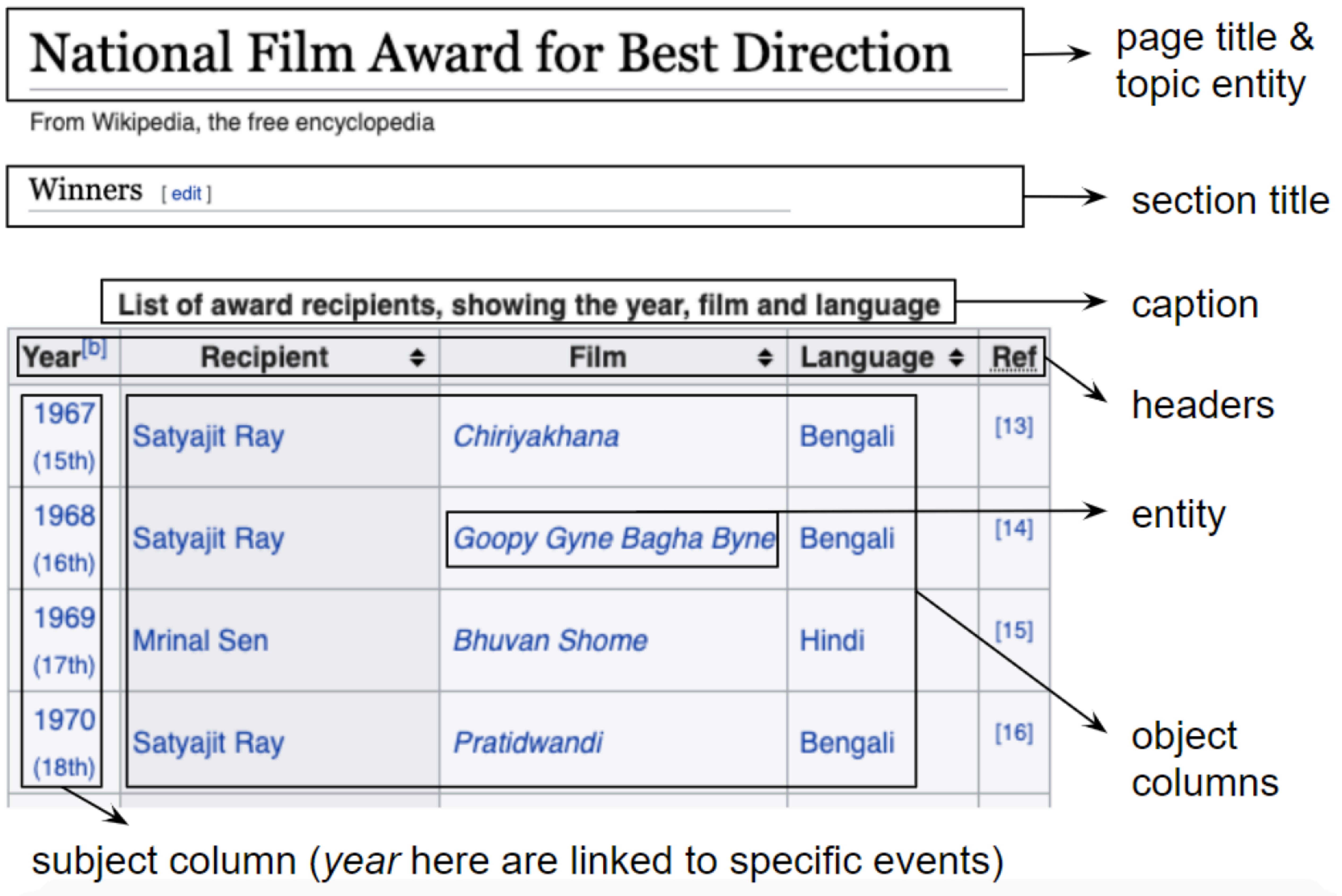
TURL: Table Understanding via Representation Learning

1. A structure-aware transformer for table encoding
2. Two pretraining objective for learn factual knowledge from web tables



3. Finetune and evaluate on 6 table-focused tasks

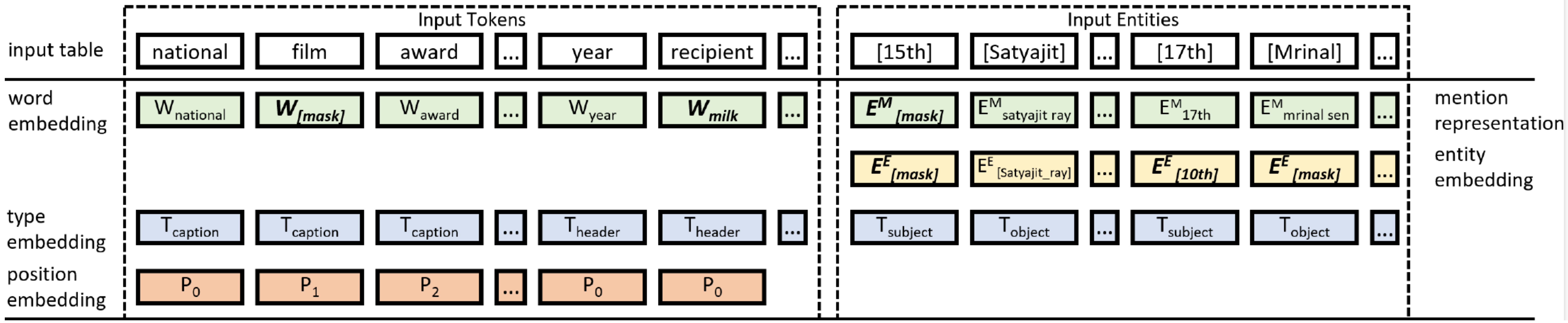




After pre-processing, 570171 / 5036 / 4964 tables for pre-training / validation / testing

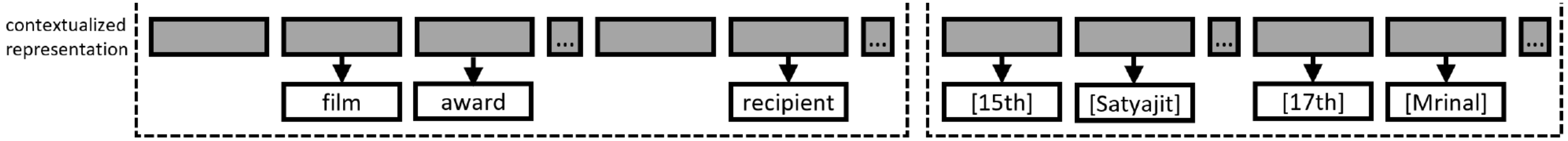
Bhagavatula, Chandra Sekhar, Thanapon Noraset, and Doug Downey. "TabEL: entity linking in web tables." International Semantic Web Conference. Springer, Cham, 2015.

Input:

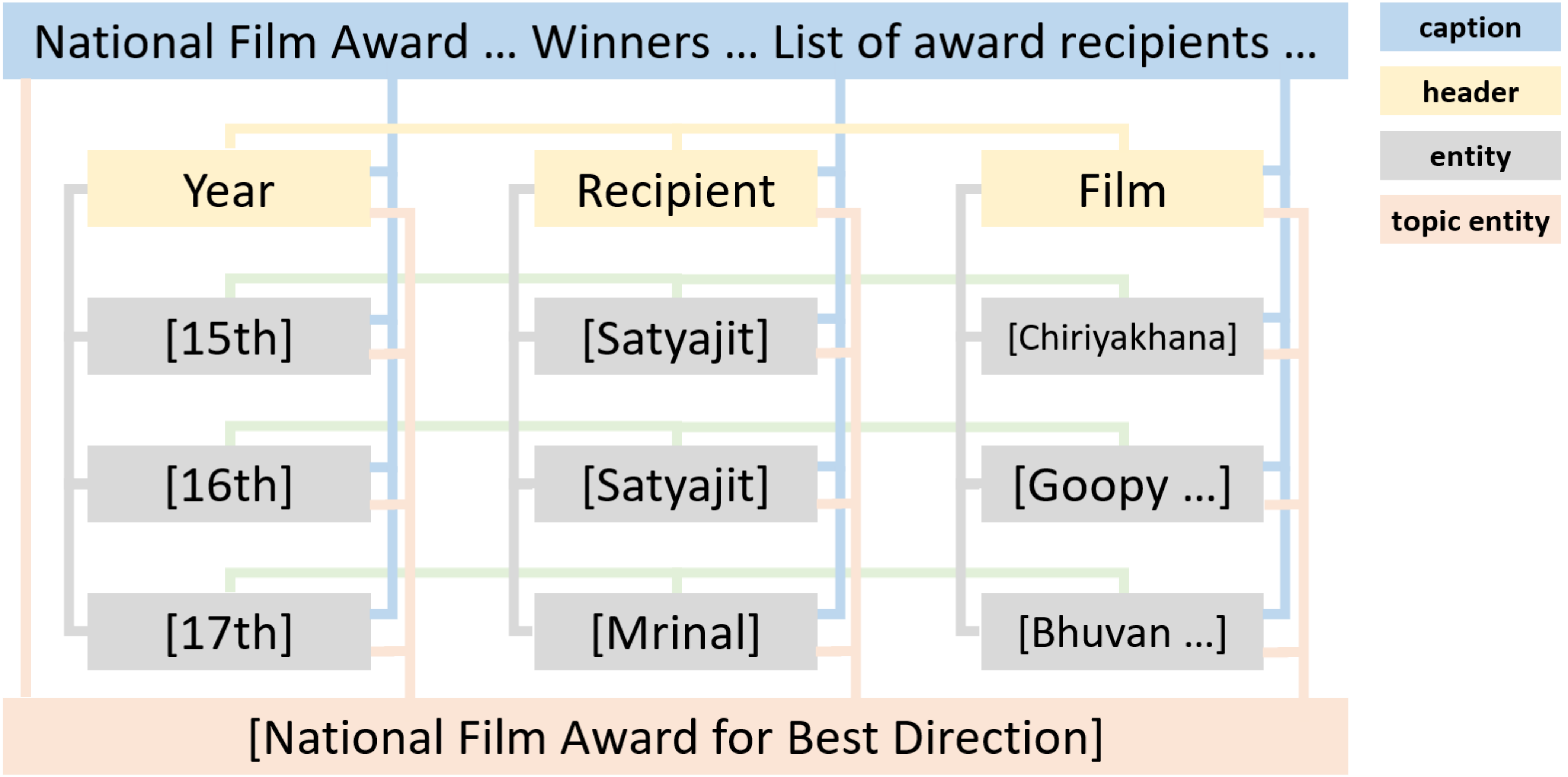


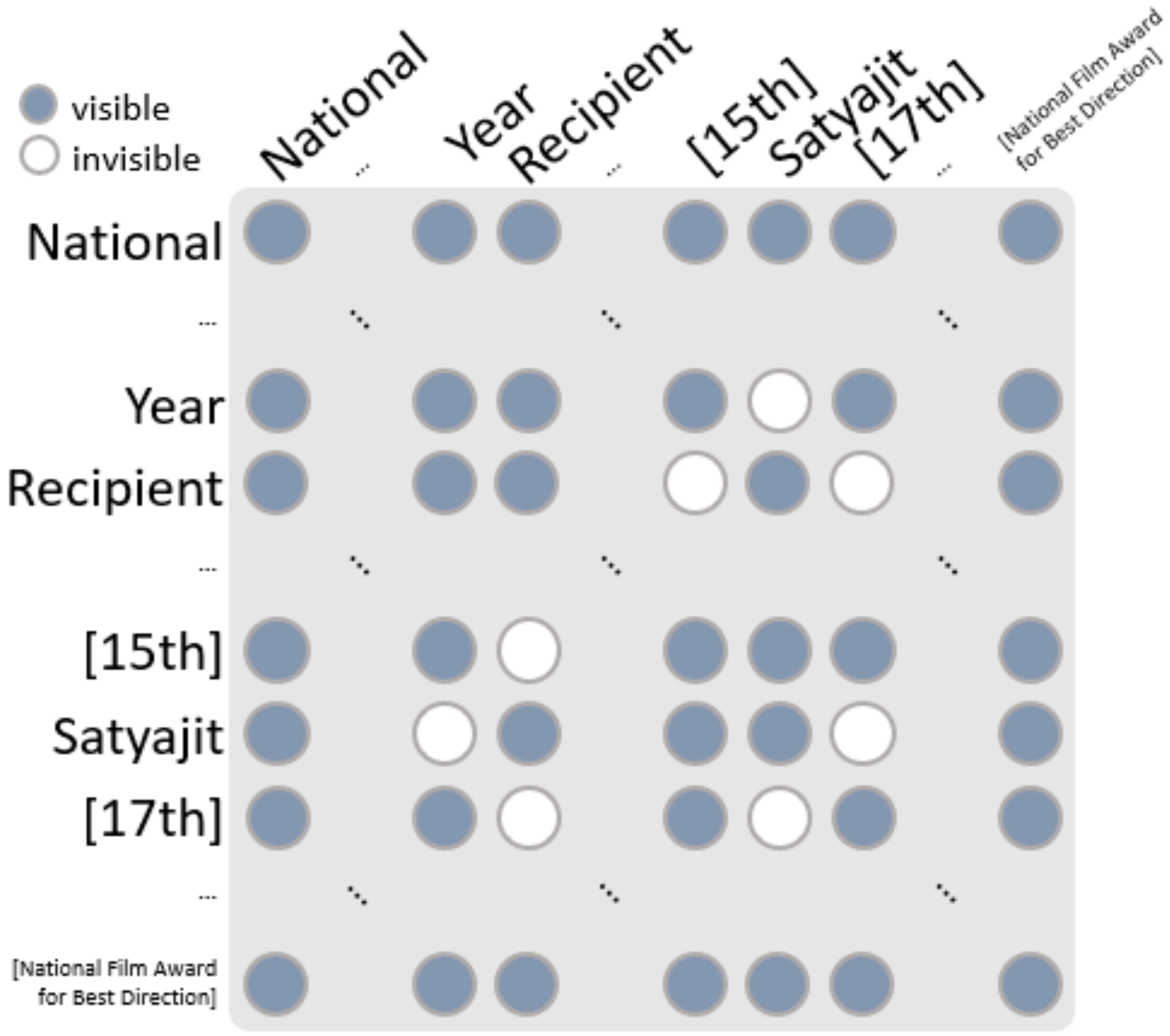
- We use type and position embeddings to represent different types of the table
- Reuse pre-trained embeddings when possible
- Each entity has a unique entity embedding and one mention embedding which is obtained from its surface form in the table

Output:



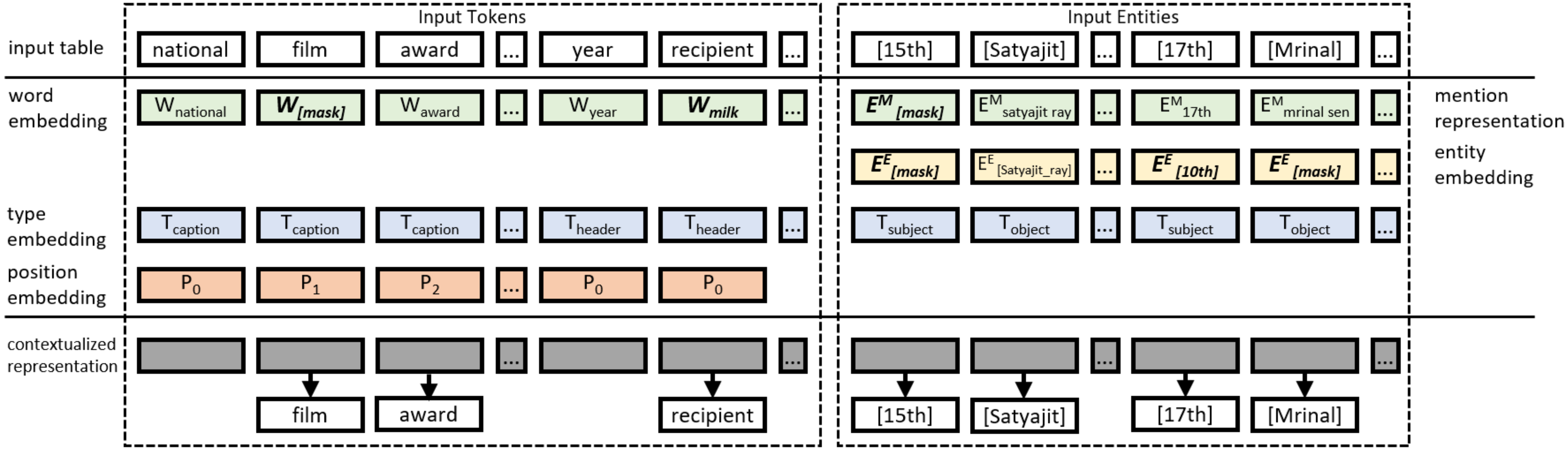
Visibility matrix





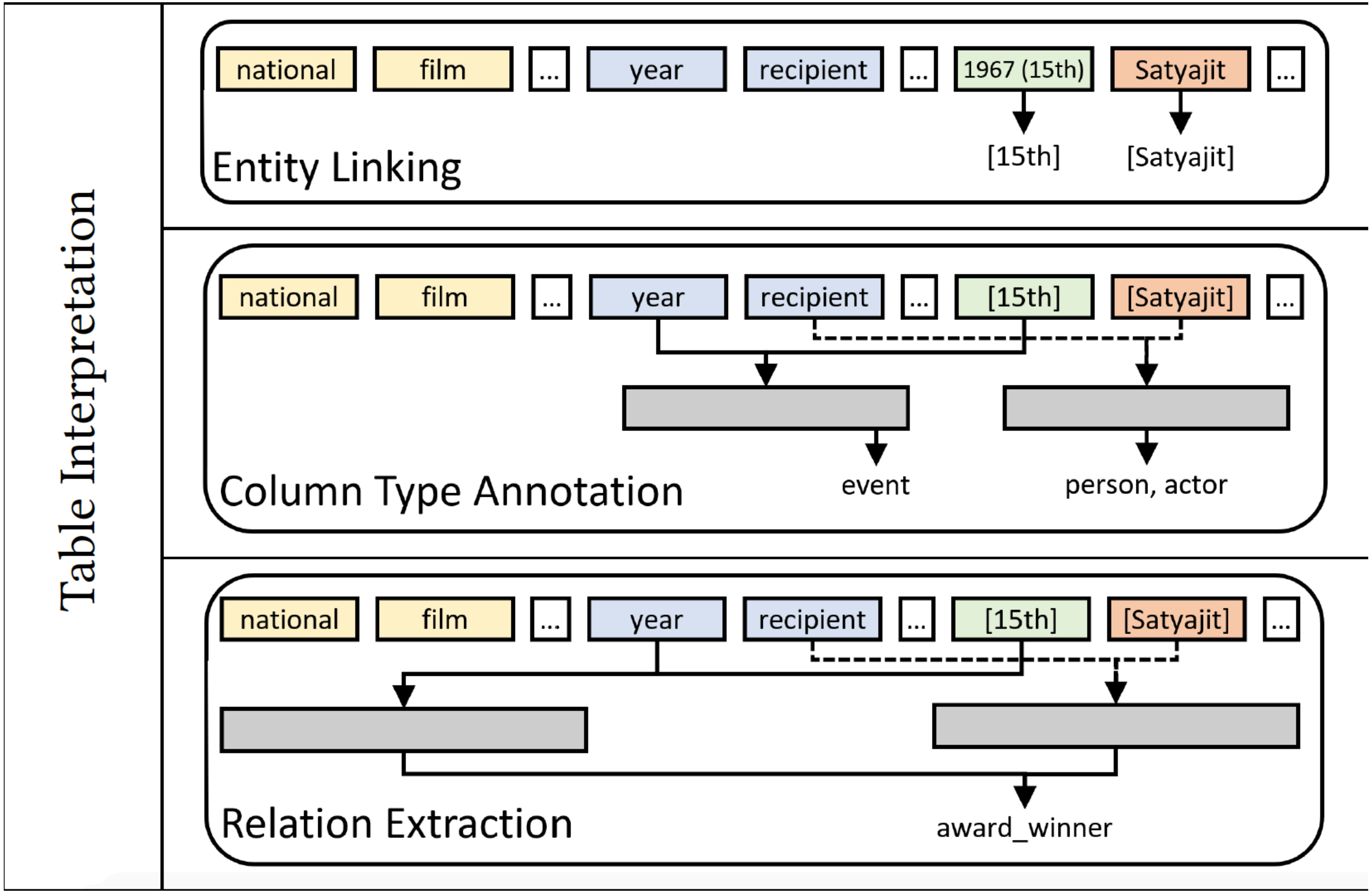
- A table is treated as a graph, so each component can only aggregate from its neighbors
- Apply mask in self-attention to model the row-column structure

Pre-training objectives:

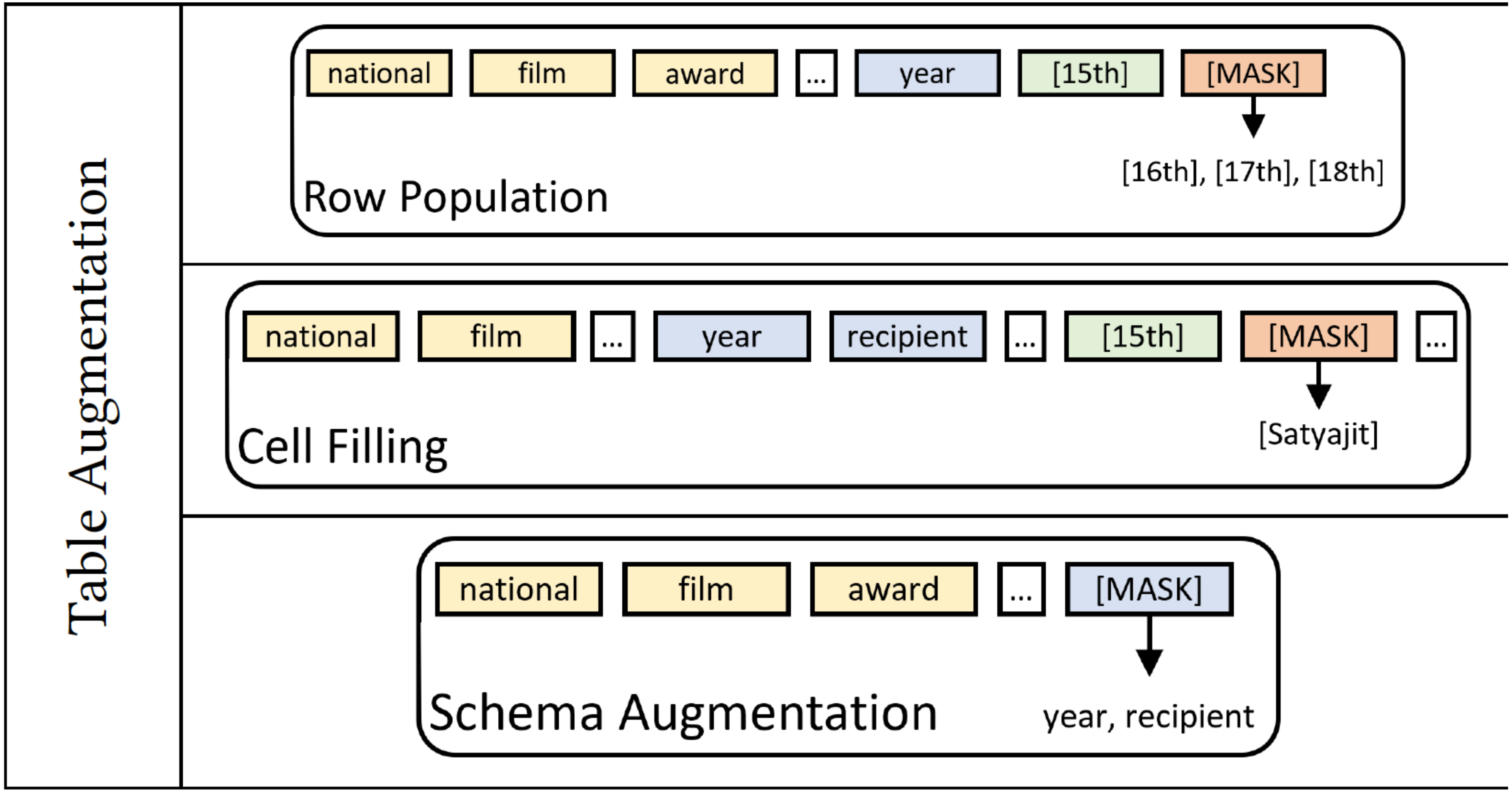


- Masked language model (MLM): learn the construction of table meta-data
- Masked entity recovery (MER): learn factual knowledge in tables
  - Mask mention and entity embedding to predict based on context
  - Mask only entity embedding to predict with help of entity mention, like entity linking

6 downstream tasks:



6 downstream tasks:



Selected results

Method	F1	P	R
BERT-based	90.94	91.18	90.69
TURL + fine-tuning (only table metadata)	92.13	91.17	93.12
TURL + fine-tuning	**94.91**	**94.57**	**95.25**
w/o table metadata	93.85	93.78	93.91
w/o learned embedding	93.35	92.90	93.80

Relation extraction from Web tables

TURL: Table Understanding via Representation Learning

- Introduce the pre-training/fine-tuning paradigm to relational web tables and related tasks

- A structure-aware transformer encoder to model relational tables and Masked Entity Recovery pretraining objective to learn the semantics as well as the factual knowledge about entities in relational tables.

- a benchmark that consists of 6 different tasks for table understanding.

[VLDB'21]

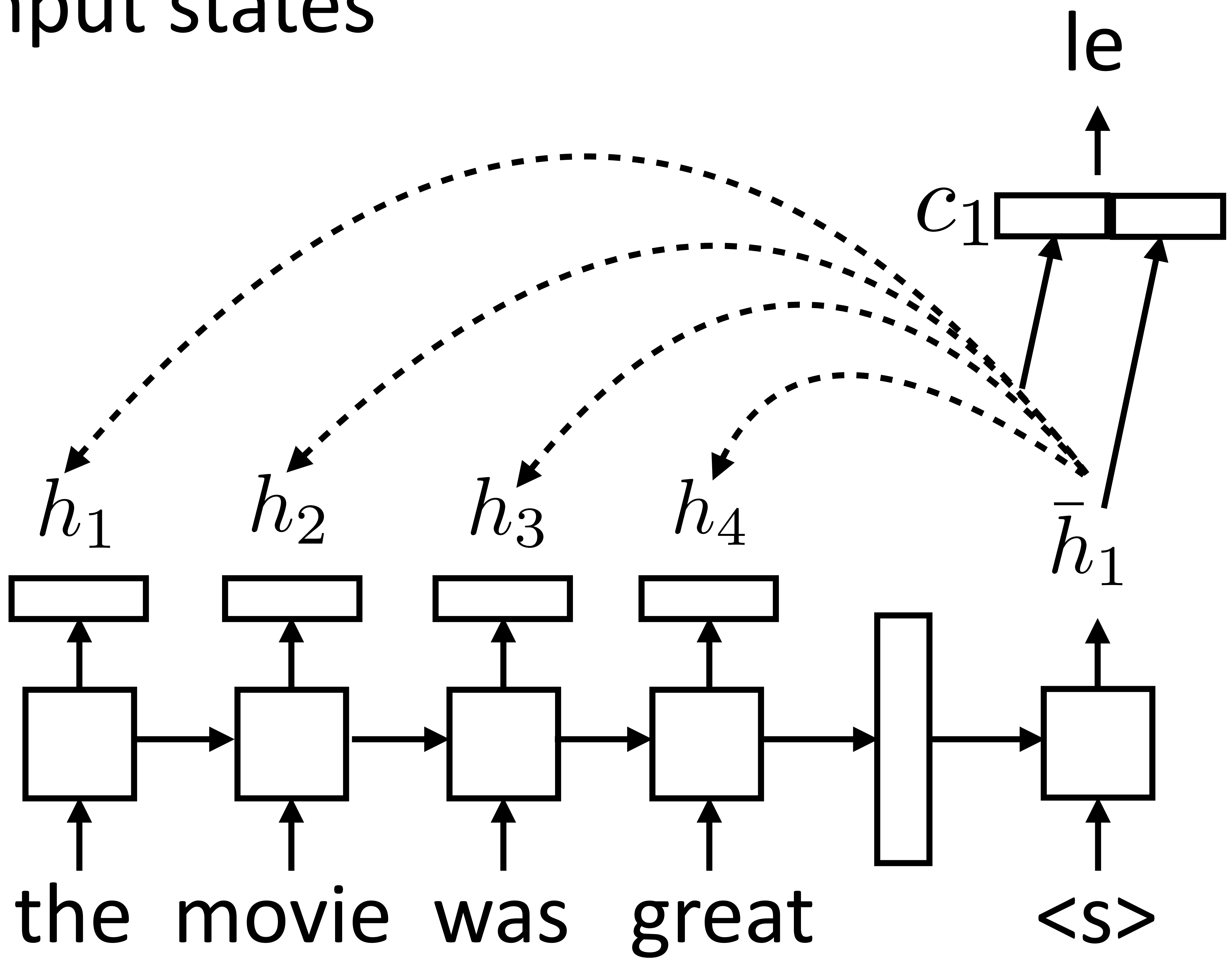
How about other tasks like table-based semantic parsing or QA?

Structure-Grounded Pretraining for Text-to-SQL, arXiv'20

Machine Translation

Recall: Attention

- For each decoder state, compute weighted sum of input states



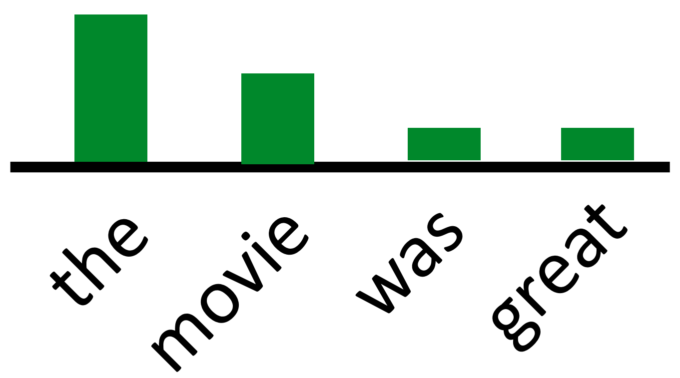
- No attn: $P(y_i|\mathbf{x}, y_1, \ldots, y_{i-1}) = \mathrm{softmax}(W\bar{h}_i)$

$$P(y_i|\mathbf{x}, y_1, \ldots, y_{i-1}) = \mathrm{softmax}(W[c_i; \bar{h}_i])$$

$$c_i = \sum_j \alpha_{ij} h_j$$

- Weighted sum of input hidden states (vector)

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{j'} \exp(e_{ij'})}$$



$$e_{ij} = f(\bar{h}_i, h_j)$$

- Some function *f* (TBD)

This Lecture

- MT basics, evaluation
- Word alignment
- Language models
- Phrase-based decoders

MT Basics

MT

People's Daily, August 30, 2017

Trump Pope family watch a hundred years a year in the White House balcony

MT Ideally

- *I have a friend* => `∃x friend(x,self)` => *J'ai un ami*

  J'ai une amie (friend is female)

  - May need information you didn't think about in your representation
  - Hard for semantic representations to cover everything

- Everyone has a friend => `∃x∀y friend(x,y)` / `∀x∃y friend(x,y)` => Tous a un ami

  - Can often get away without doing all disambiguation — same ambiguities may exist in both languages

Phrase-Based MT

- Key idea: translation works better the bigger chunks you use

Phrase-Based MT

- Key idea: translation works better the bigger chunks you use
- Remember phrases from training data, translate piece-by-piece and stitch those pieces together to translate
  - How to identify phrases? Word alignment over source-target bitext
  - How to stitch together? Language model over target language
  - Decoder takes phrases and a language model and searches over possible translations

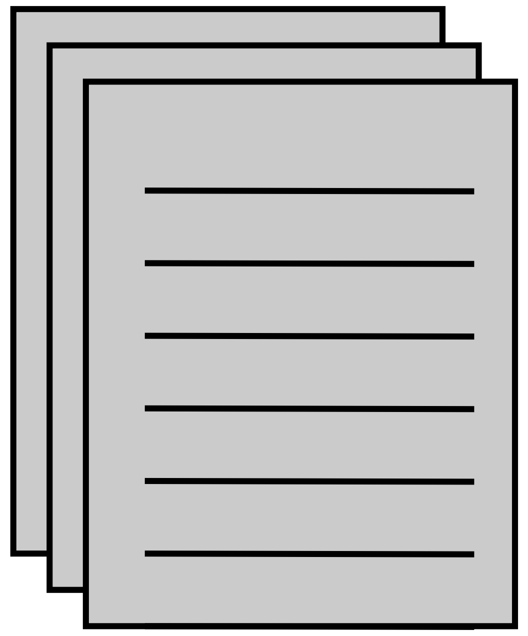
Phrase-Based MT

- Key idea: translation works better the bigger chunks you use
- Remember phrases from training data, translate piece-by-piece and stitch those pieces together to translate
  - How to identify phrases? Word alignment over source-target bitext
  - How to stitch together? Language model over target language
  - Decoder takes phrases and a language model and searches over possible translations
- NOT like standard discriminative models (take a bunch of translation pairs, learn a ton of parameters in an end-to-end way)

Phrase-Based MT

```
cat ||| chat ||| 0.9
the cat ||| le chat ||| 0.8
dog ||| chien ||| 0.8
house ||| maison ||| 0.6
my house ||| ma maison ||| 0.9
language ||| langue ||| 0.9
…
```

Phrase table P(f|e)



Language model P(e)

Unlabeled English data

$$P(e|f) \propto P(f|e)P(e)$$

Noisy channel model: combine scores from translation model + language model to translate foreign to English

"Translate faithfully but make fluent English"

Evaluating MT

- Fluency: does it sound good in the target language?
- Fidelity/adequacy: does it capture the meaning of the original?

Evaluating MT

- Fluency: does it sound good in the target language?
- Fidelity/adequacy: does it capture the meaning of the original?
- BLEU score: geometric mean of 1-, 2-, 3-, and 4-gram *precision* vs. a reference, multiplied by brevity penalty (penalizes short translations)

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right).$$

- Typically $n = 4$, $w_i = 1/4$

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}.$$

r = length of reference

c = length of prediction

Evaluating MT

- Fluency: does it sound good in the target language?
- Fidelity/adequacy: does it capture the meaning of the original?
- BLEU score: geometric mean of 1-, 2-, 3-, and 4-gram *precision* vs. a reference, multiplied by brevity penalty (penalizes short translations)

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right).$$

- Typically $n = 4$, $w_i = 1/4$

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}.$$

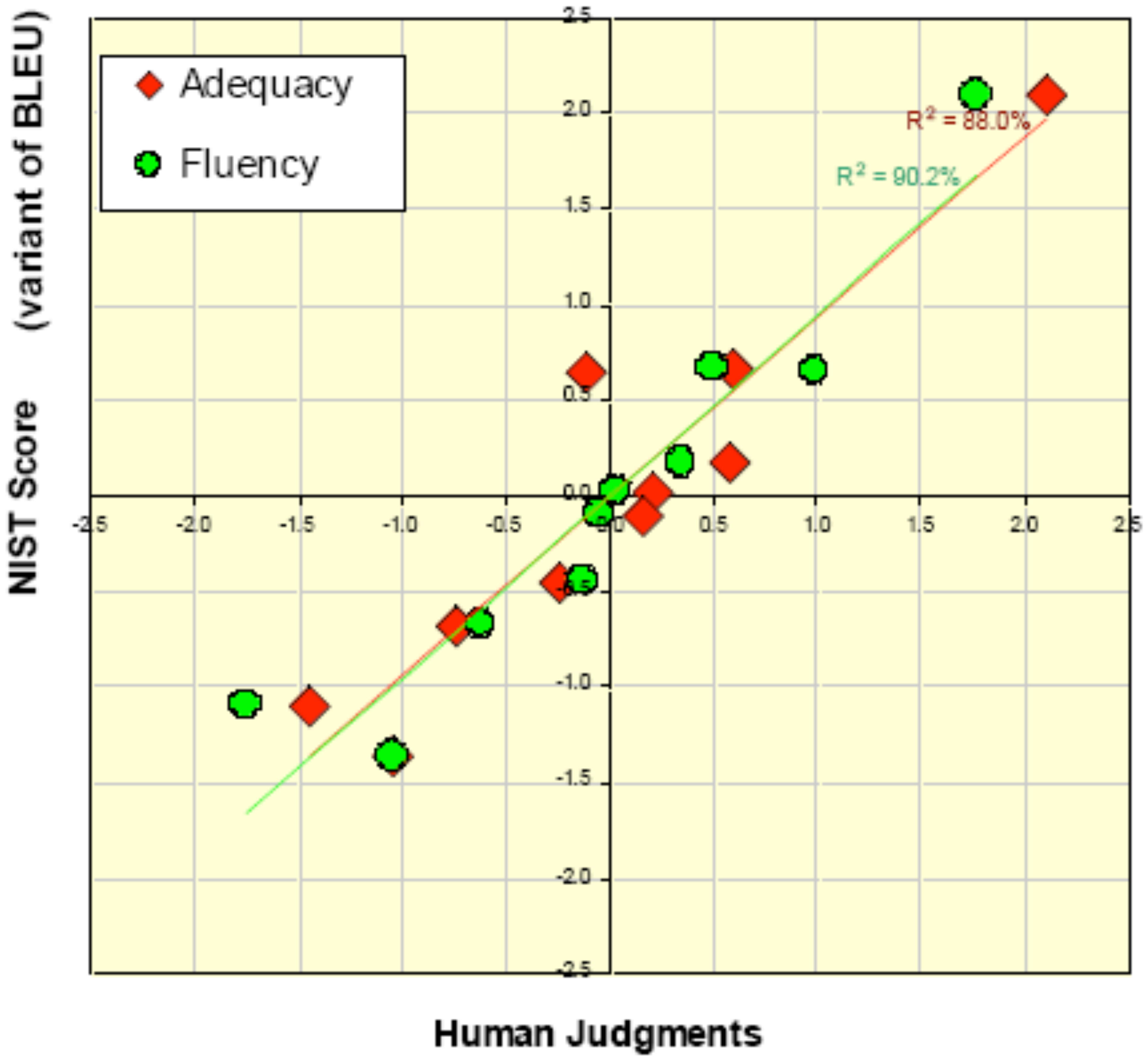
r = length of reference

c = length of prediction

- Does this capture fluency and adequacy?

BLEU Score

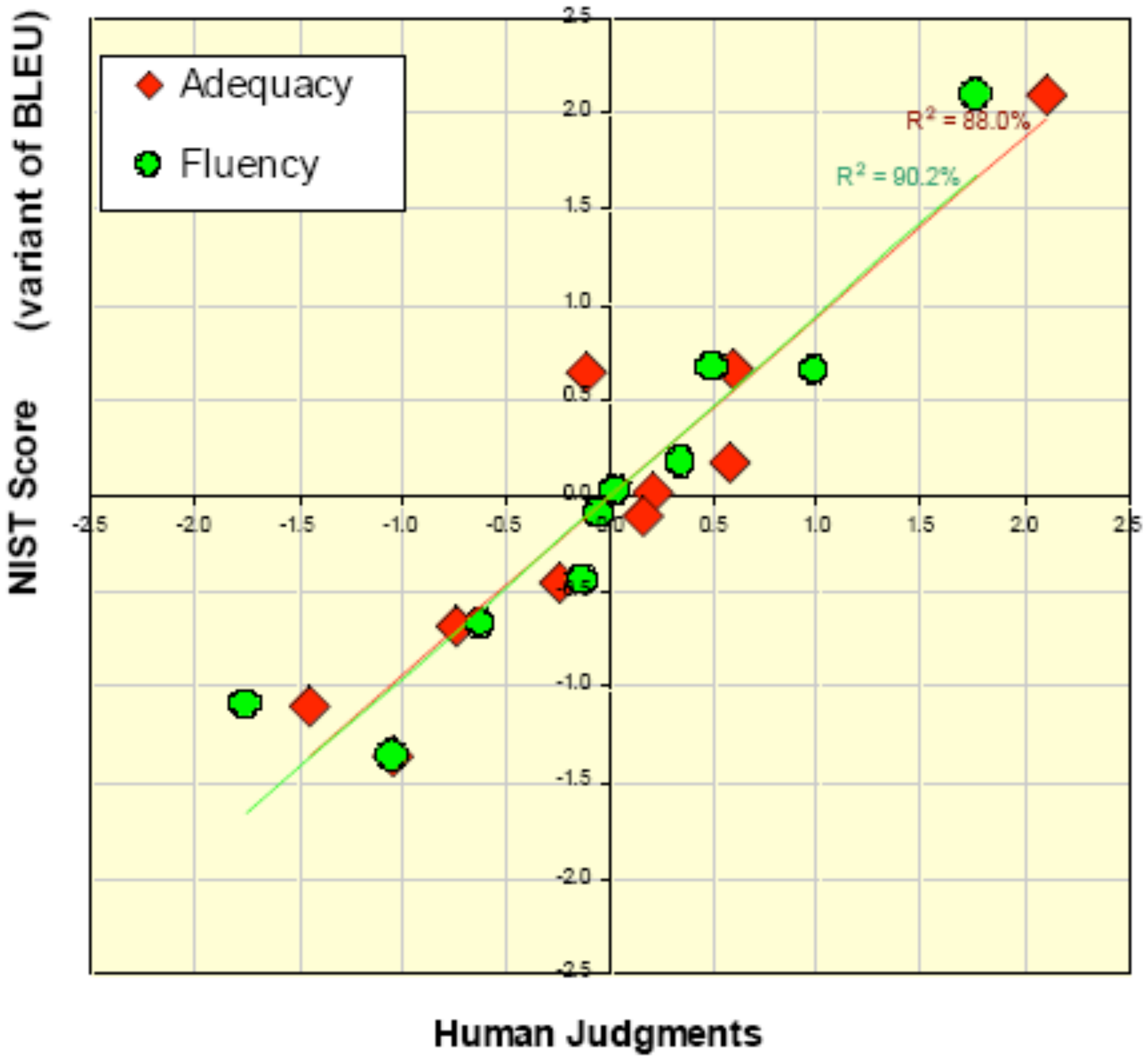
- At a *corpus* level, BLEU correlates pretty well with human judgments



slide from G. Doddington (NIST)

BLEU Score

- At a *corpus* level, BLEU correlates pretty well with human judgments
- Better methods with human-in-the-loop
- BLEU scores + user studies



slide from G. Doddington (NIST)

Word Alignment

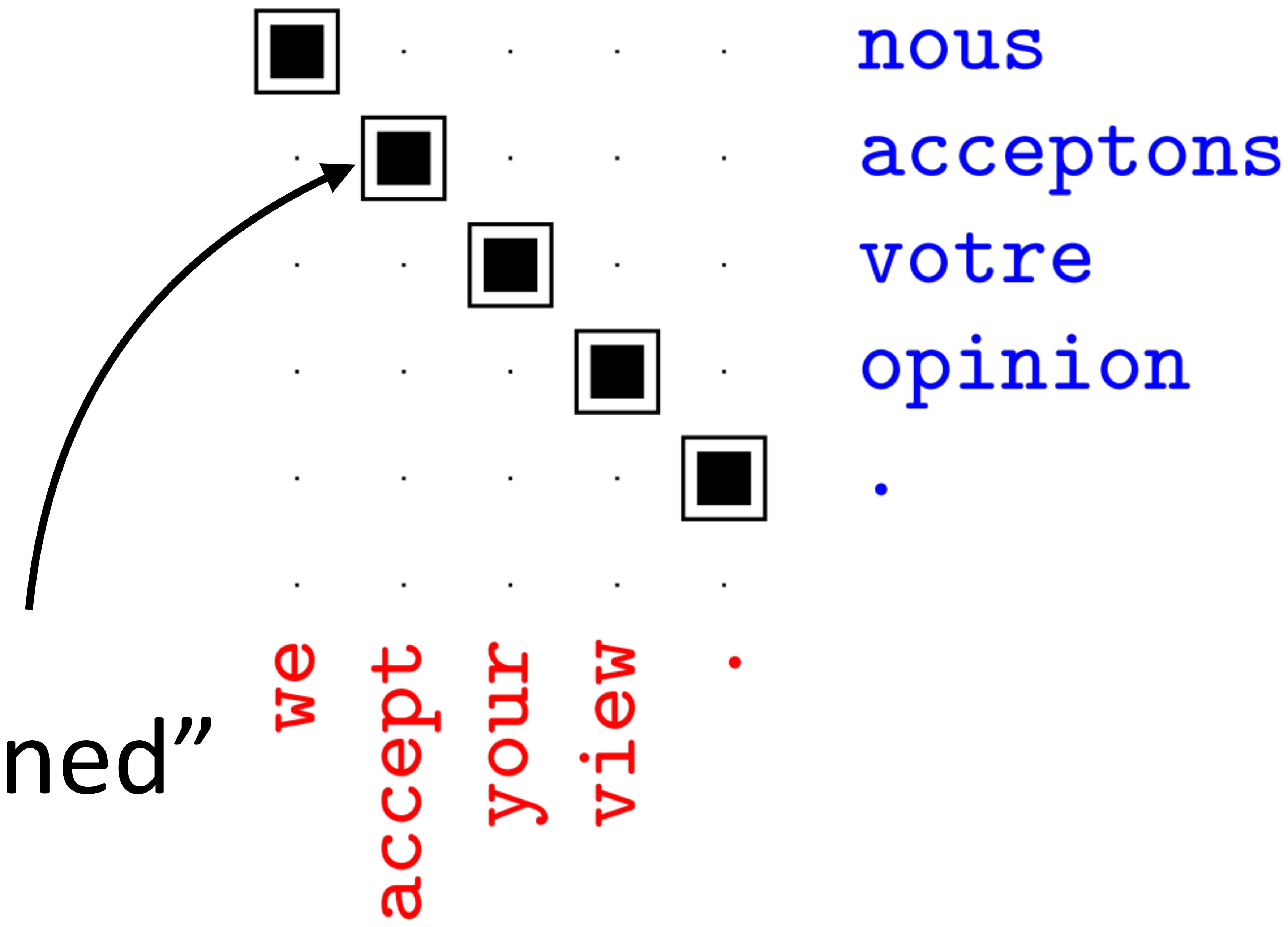
Word Alignment

- Input: a bitext, pairs of translated sentences

  nous acceptons votre opinion . ||| we accept your view

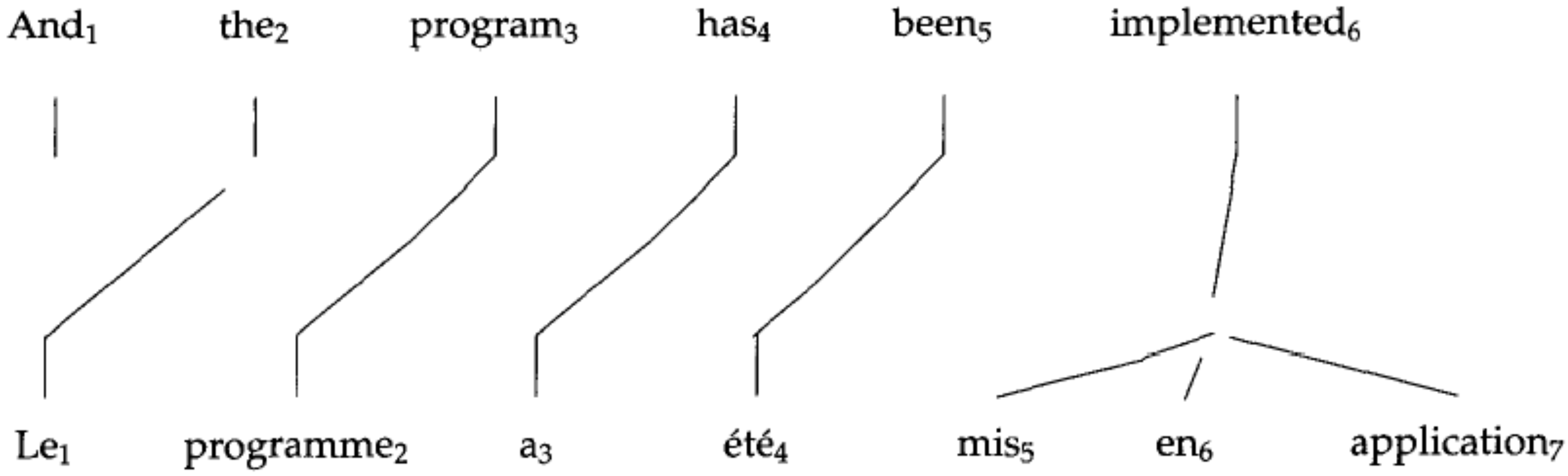
  nous allons changer d'avis ||| we are going to change our minds

- Output: alignments between words in each sentence
  - We will see how to turn these into phrases



"accept and acceptons are aligned"

1-to-Many Alignments



Word Alignment

- Models P(**f**|**e**): probability of "French" sentence being generated from "English" sentence according to a model

- Latent variable model: $P(\mathbf{f}|\mathbf{e}) = \sum_{\mathbf{a}} P(\mathbf{f}, \mathbf{a}|\mathbf{e}) = \sum_{\mathbf{a}} P(\mathbf{f}|\mathbf{a}, \mathbf{e})P(\mathbf{a})$

- Correct alignments should lead to higher-likelihood generations, so by optimizing this objective we will learn correct alignments

Decoding

Recall: *n*-gram Language Models

$$P(\mathbf{w}) = P(w_1)P(w_2|w_1)P(w_3|w_1, w_2)\ldots$$

- *n*-gram models: distribution of next word is a multinomial conditioned on previous *n*-1 words $P(w_i|w_1, \ldots, w_{i-1}) = P(w_i|w_{i-n+1}, \ldots, w_{i-1})$

I visited San ______ put a distribution over the next word

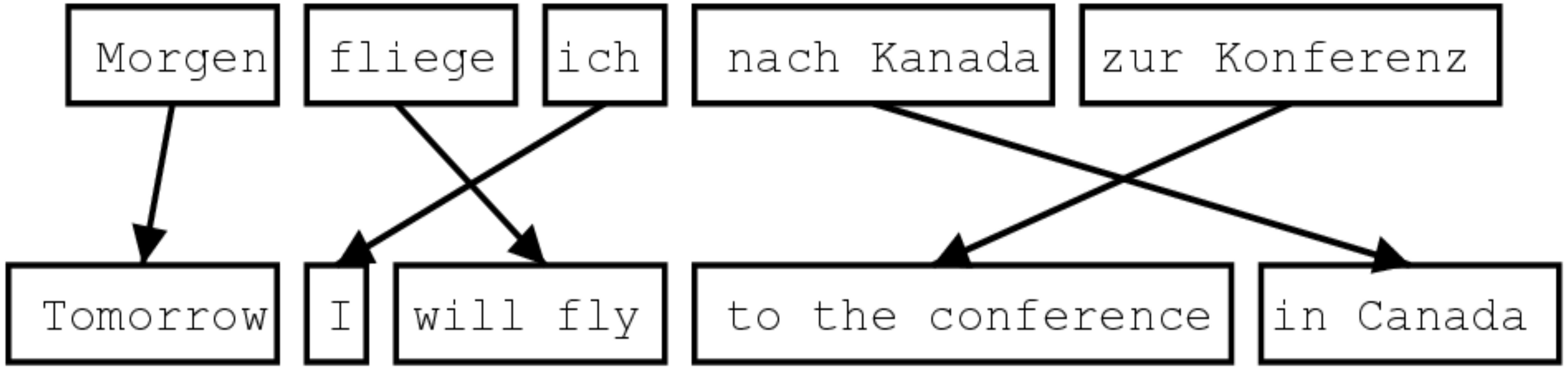
$$P(w|\text{visited San}) = \frac{\text{count}(\text{visited San}, w)}{\text{count}(\text{visited San})}$$

Maximum likelihood estimate of this 3-gram probability from a corpus

- Typically use ~5-gram language models for translation

Phrase-Based Decoding

- Inputs:
  - n-gram language model: $P(e_i|e_1,\ldots,e_{i-1}) \approx P(e_i|e_{i-n-1},\ldots,e_{i-1})$
  - Phrase table: set of phrase pairs (**e**, **f**) with probabilities P(**f**|**e**)
- What we want to find: **e** produced by a series of phrase-by-phrase translations from an input **f**, possibly with reordering:



Phrase lattices are big!

这	7人	中包括	来自	法国	和	俄罗斯	的	宇航	员	.
the	7 people	including	by some		**and**	the russian	**the**	the astronauts		,
it	7 people included		by france		and the	the russian		international astronautical	of rapporteur .	
this	7 out	including the	**from**	the french	and the russian		the fifth		.	
these	7 among	including from		the french and		of the russian	of	space	members	.
that	7 persons	including from the		of france	and to	russian	of the	aerospace	members .	
	7 include		from the	of france and		russian		**astronauts**		. the
	7 numbers include		**from france**		and russian		of astronauts who			. ”
	7 populations include		those from france		and russian			astronauts .		
	7 deportees included		come from	**france**	**and russia**		in	astronautical	personnel	;
	7 philtrum	including those from		**france and**		**russia**	a space		**member**	
		including representatives from		france and the		**russia**		astronaut		
		include	came from	**france and russia**			by cosmonauts			
		include representatives from		french	**and russia**			cosmonauts		
		include	came from france		and russia 's			cosmonauts .		
		includes	coming from	french and		russia 's		cosmonaut		
				french and russian			's	astronavigation	member .	
				french	**and russia**		**astronauts**			
					and russia 's				special rapporteur	
					, and	**russia**			rapporteur	
					, and russia				rapporteur .	
					, and russia					
					or	russia 's				

Slide credit: Dan Klein

Phrase-Based Decoding

- Input

lo haré | rápidamente | .

- Translations

I'll do it | quickly | .

quickly | I'll do it | .

The decoder…

tries different segmentations,

translates phrase by phrase,

and considers reorderings.

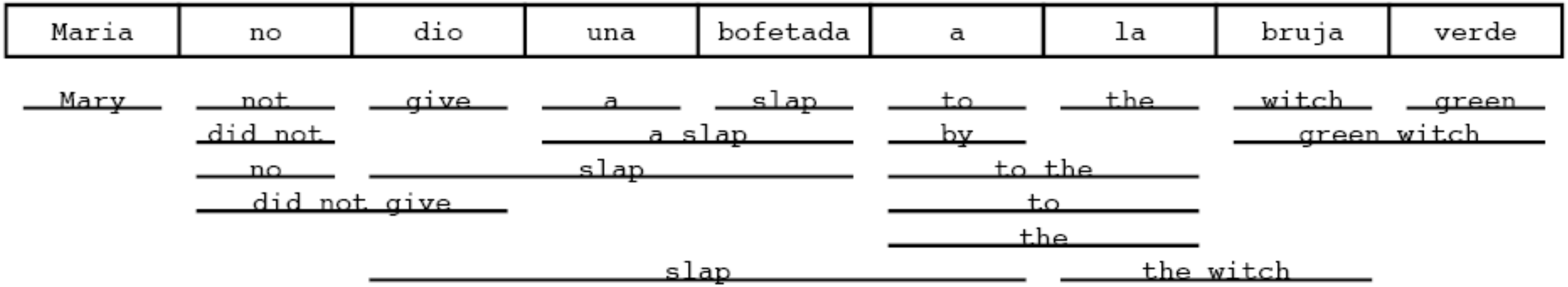
$$\arg\max_{\mathbf{e}} [P(\mathbf{f}|\mathbf{e}) \cdot P(\mathbf{e})]$$

- Decoding objective (for 3-gram LM)

$$\arg\max_{\mathbf{e}} \left[\prod_{\langle \bar{e}, \bar{f} \rangle} P(\bar{f}|\bar{e}) \cdot \prod_{i=1}^{|\mathbf{e}|} P(e_i|e_{i-1}, e_{i-2}) \right]$$

Slide credit: Dan Klein

Monotonic Translation

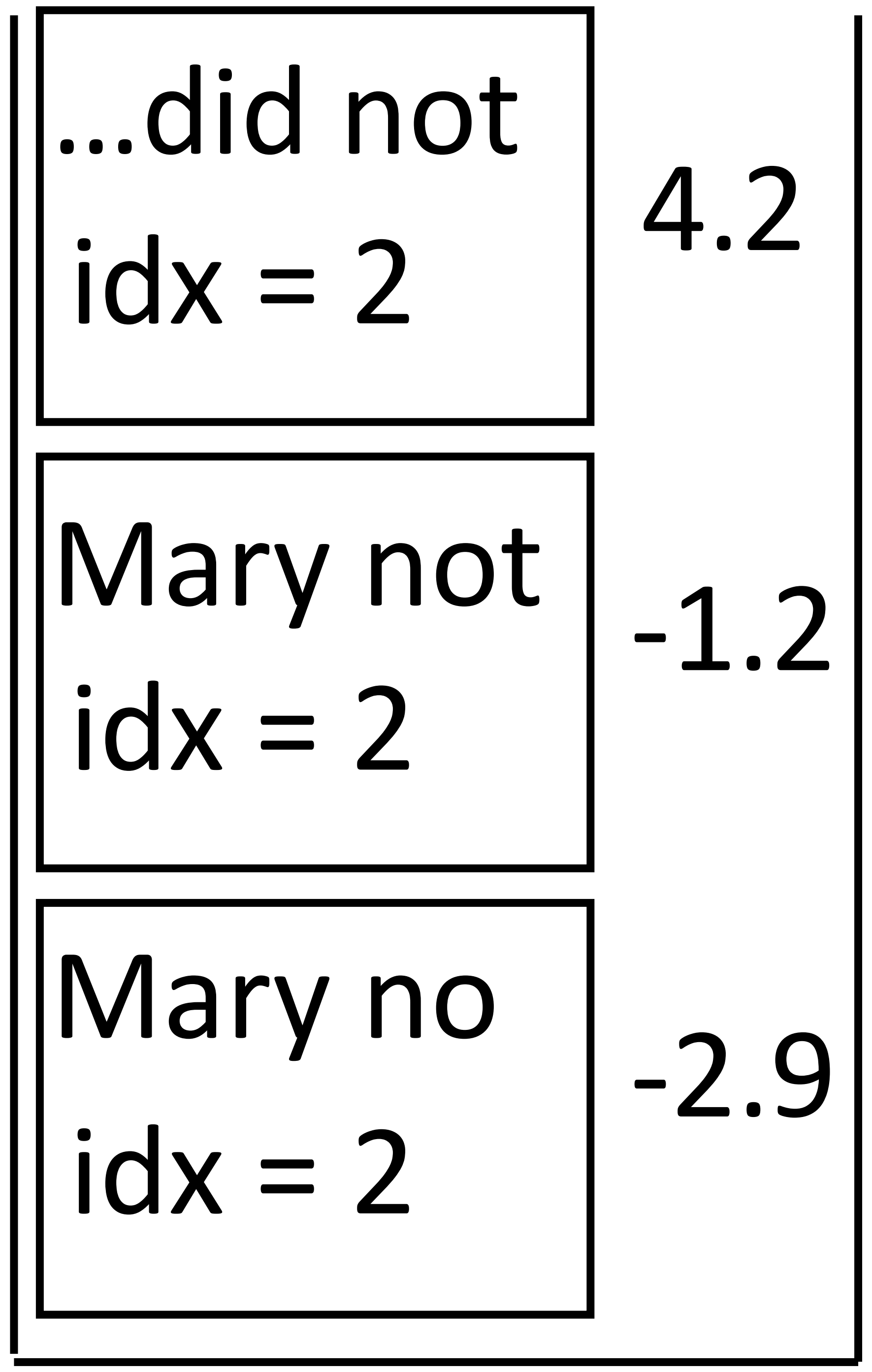


- If we translate with beam search, what state do we need to keep in the beam?
  - What have we translated so far?
  - What words have we produced so far?
  - When using a 3-gram LM, only need to remember the last 2 words!

$$\arg\max_{\mathbf{e}} \left[\prod_{\langle \bar{e}, \bar{f} \rangle} P(\bar{f}|\bar{e}) \cdot \prod_{i=1}^{|\mathbf{e}|} P(e_i|e_{i-1}, e_{i-2}) \right]$$

Koehn (2004)

Monotonic Translation



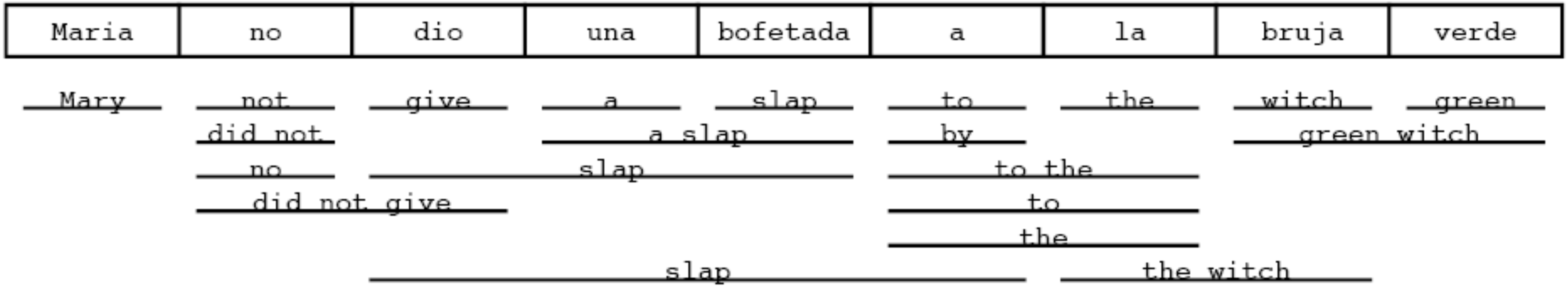
$$\text{score} = \log [\underbrace{P(\text{Mary})\, P(\text{not} \mid \text{Mary})}_{\text{LM}}\ \underbrace{P(\text{Maria} \mid \text{Mary})\, P(\text{no} \mid \text{not})}_{\text{TM}}]$$

In reality: score = $\alpha \log P(\text{LM}) + \beta \log P(\text{TM})$

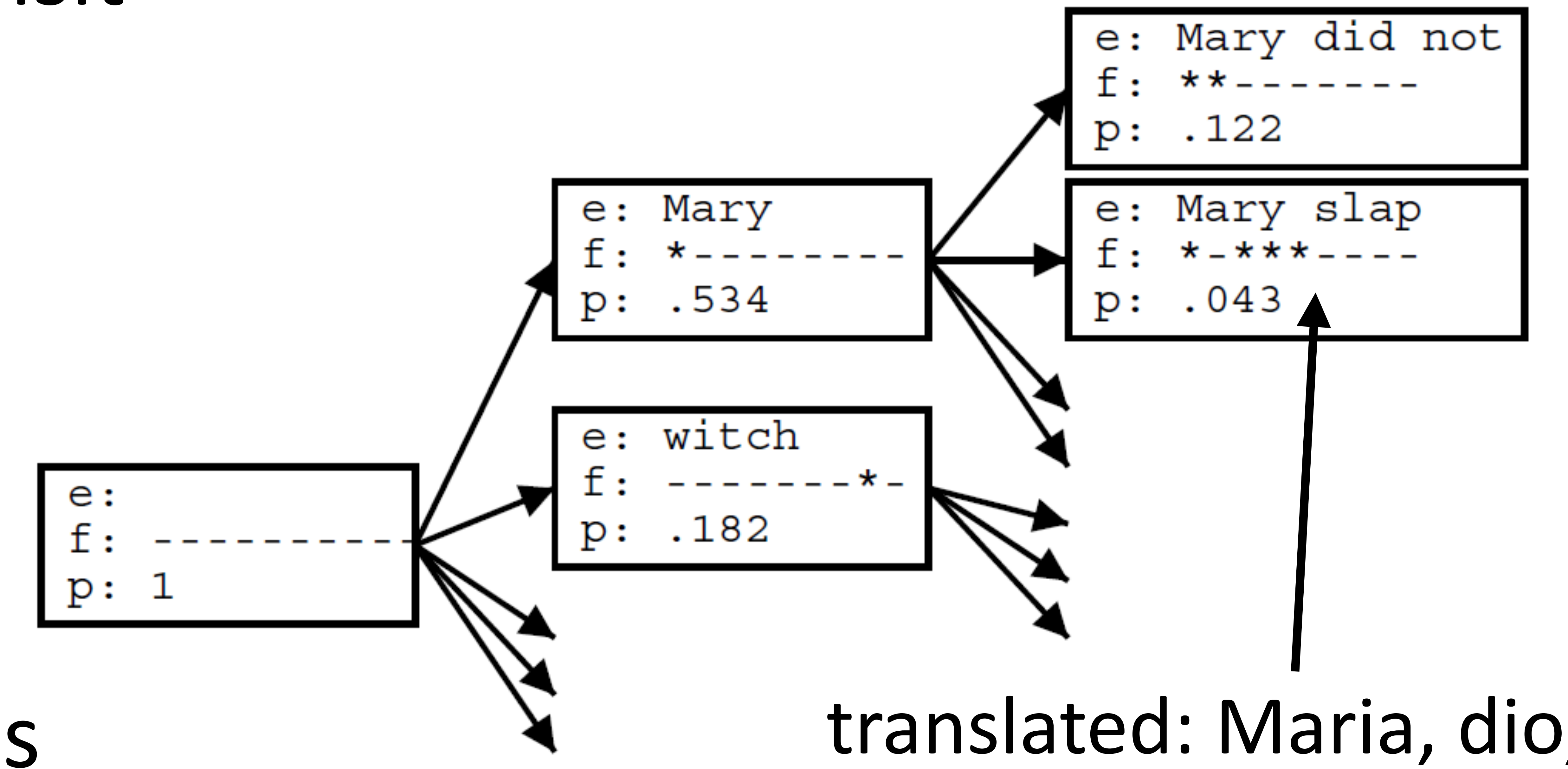
…and TM is broken down into several features

Koehn (2004)

Non-Monotonic Translation



- Non-monotonic translation: can visit source sentence “out of order”
- State needs to describe which words have been translated and which haven’t
- Big enough phrases already capture lots of reorderings, so this isn’t as important as you think



translated: Maria, dio, una, bofetada

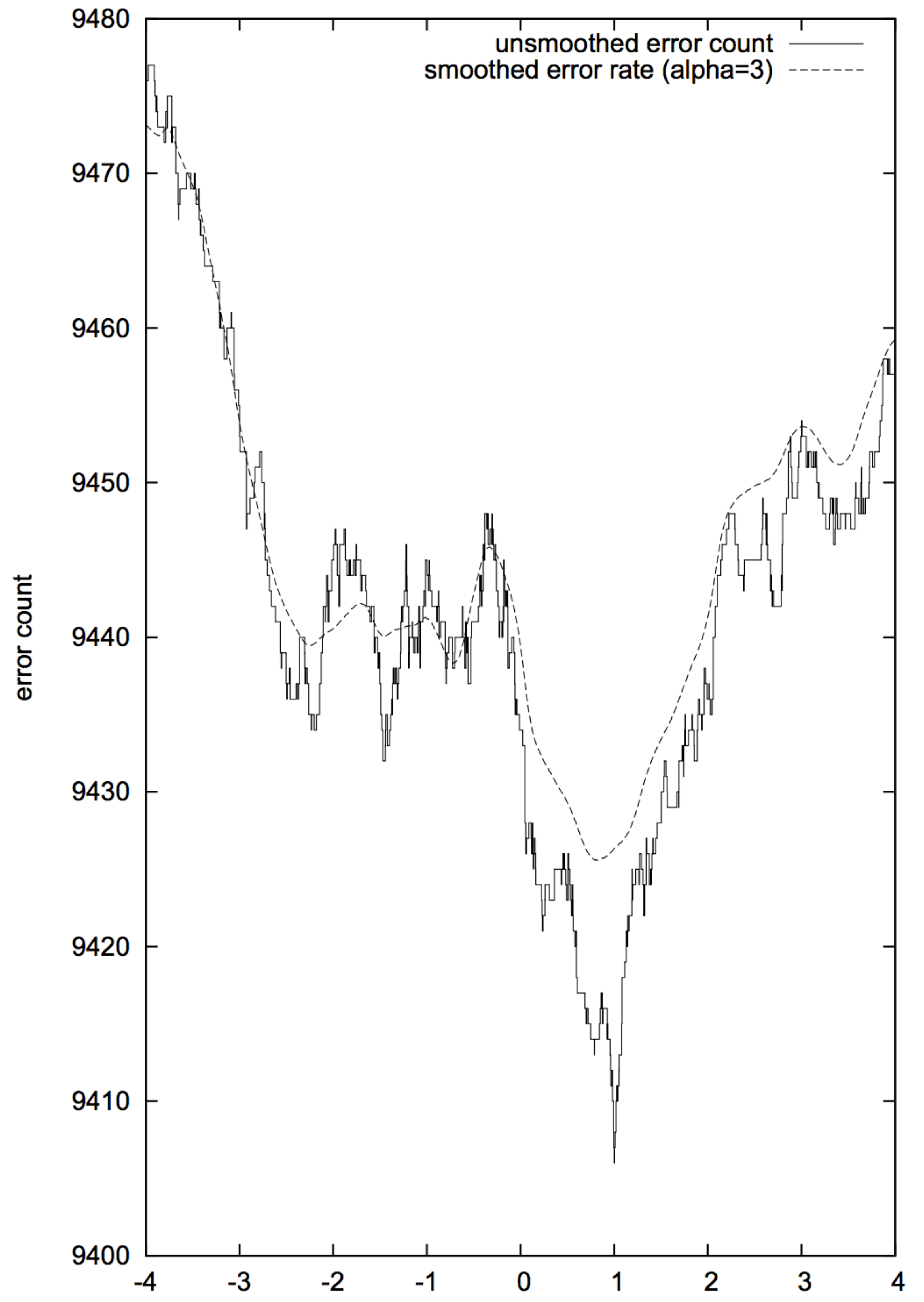
Training Decoders

score = α log P(LM) + β log P(TM)

…and TM is broken down into several feature

- Usually 5-20 feature weights to set, want to optimize for BLEU score which is not differentiable

- MERT (Och 2003): decode to get 1000-best translations for each sentence in a small training set (<1000 sentences), do line search on parameters to directly optimize for BLEU



Moses

- Toolkit for machine translation due to Philipp Koehn + Hieu Hoang
  - Pharaoh (Koehn, 2004) is the decoder from Koehn's thesis
- Moses implements word alignment, language models, and this decoder, plus *a ton* more stuff
  - Highly optimized and heavily engineered, could more or less build SOTA translation systems with this from 2007-2015
- Next time: results on these and comparisons to neural methods

Takeaways

- Phrase-based systems consist of 3 pieces: aligner, language model, decoder
  - HMMs work well for alignment
  - N-gram language models are scalable and historically worked well
  - Decoder requires searching through a complex state space
- Lots of system variants incorporating syntax