Improved Relation Extraction with Feature-Rich Compositional Embedding Models

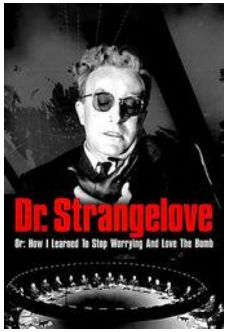
Mo Yu*
Matt Gormley*
Mark Dredze

September 21, 2015
EMNLP

*Co-first authors

JOHNS HOPKINS UNIVERSITY

FCM or: How I Learned to Stop Worrying (about Deep Learning) and Love Features



Mo Yu*

Matt Gormley*

Mark Dredze

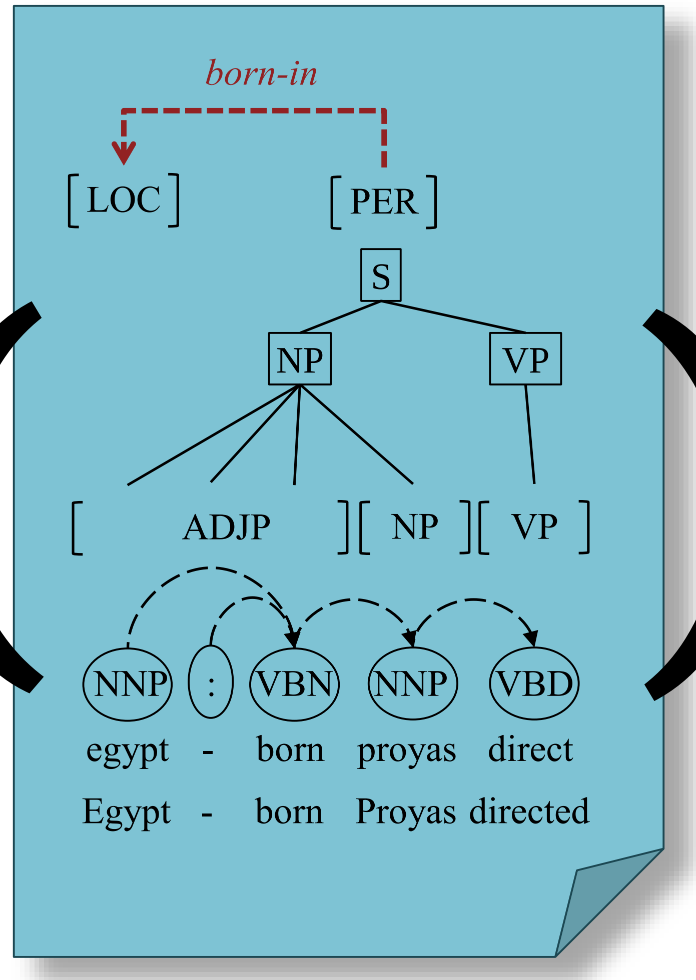
September 21, 2015

EMNLP

*Co-first authors

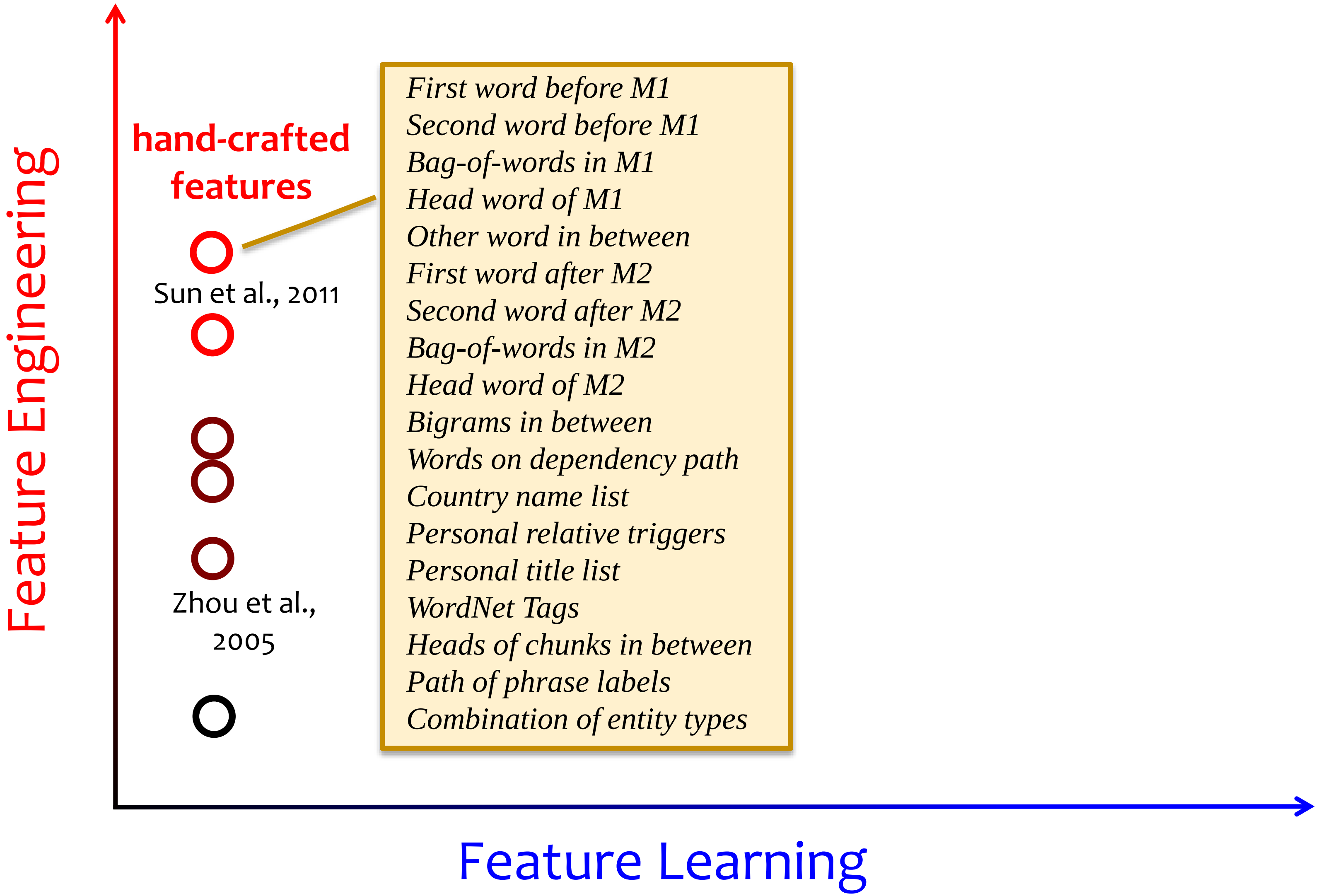
Handcrafted Features

$p(y|x) \propto \exp(\Theta_y \bullet f($

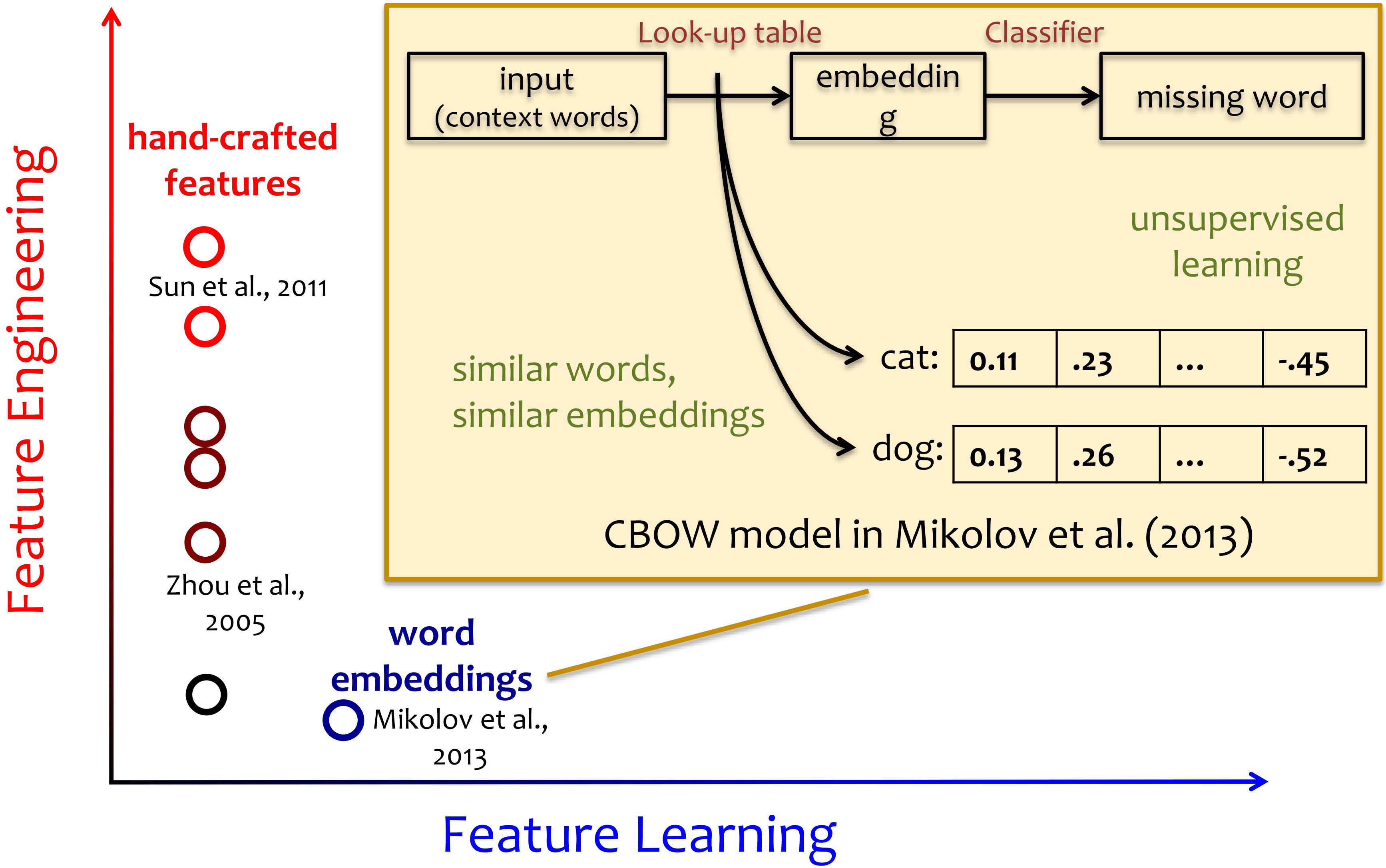


$))$

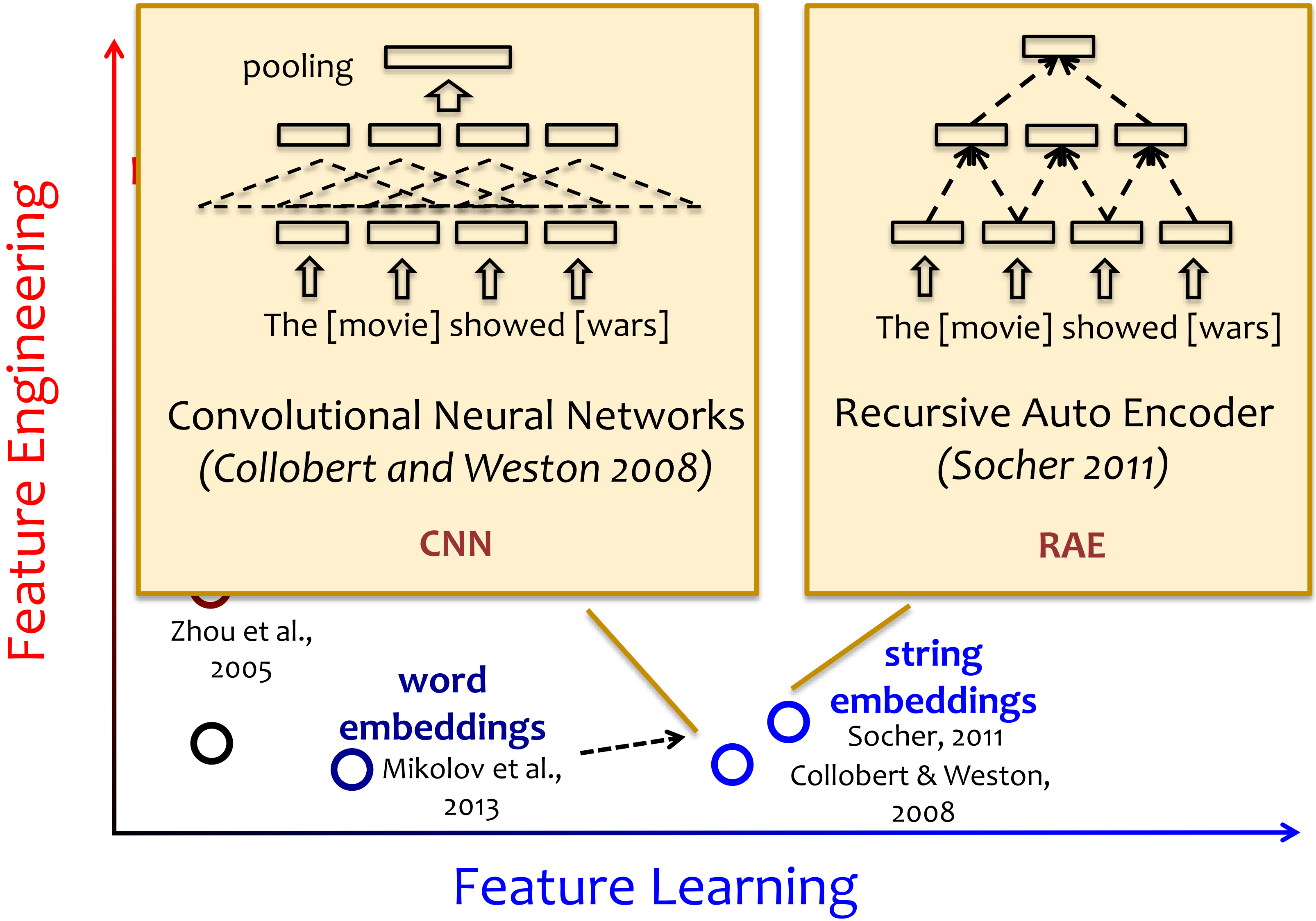
Where do features come from?



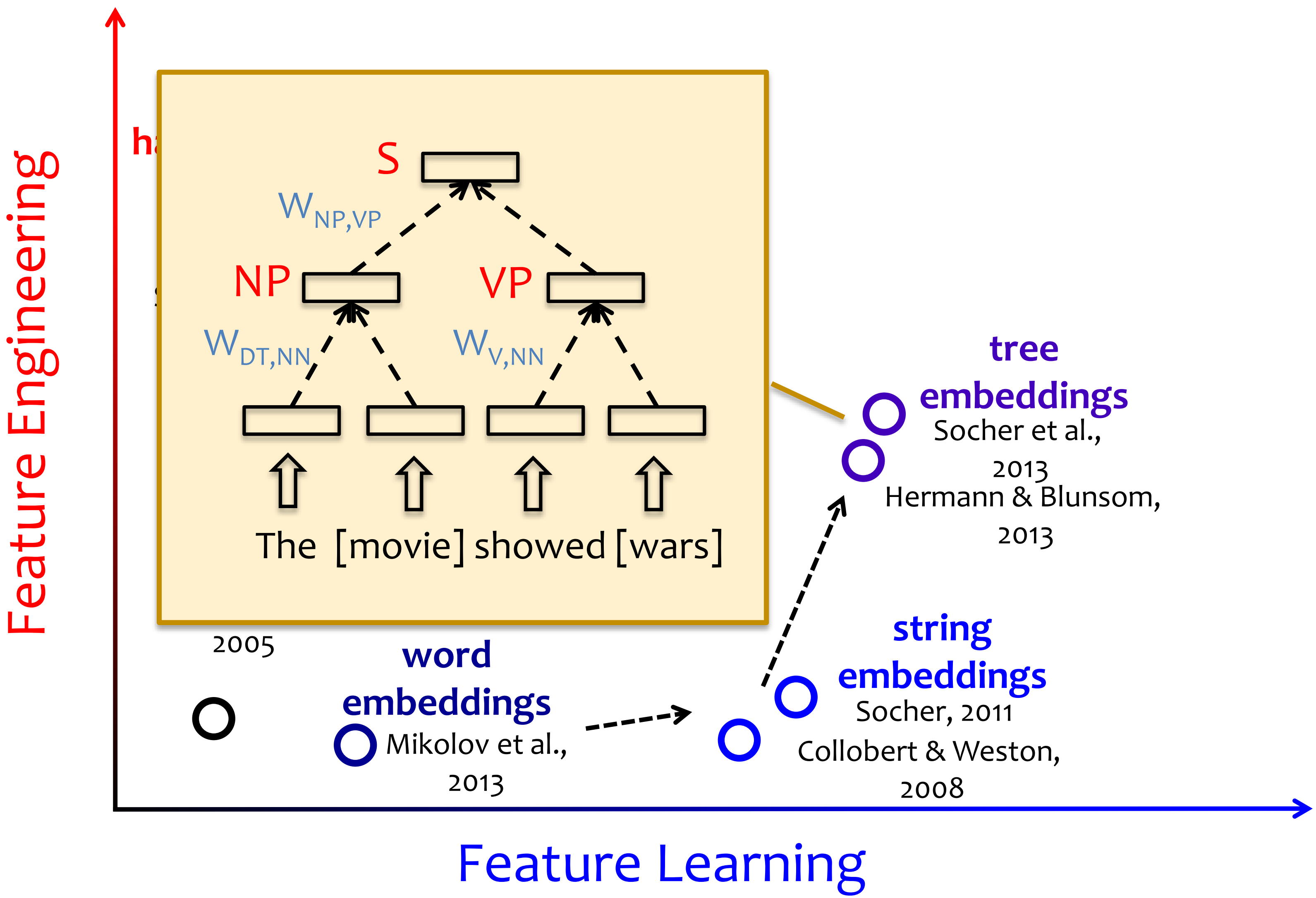
Where do features come from?



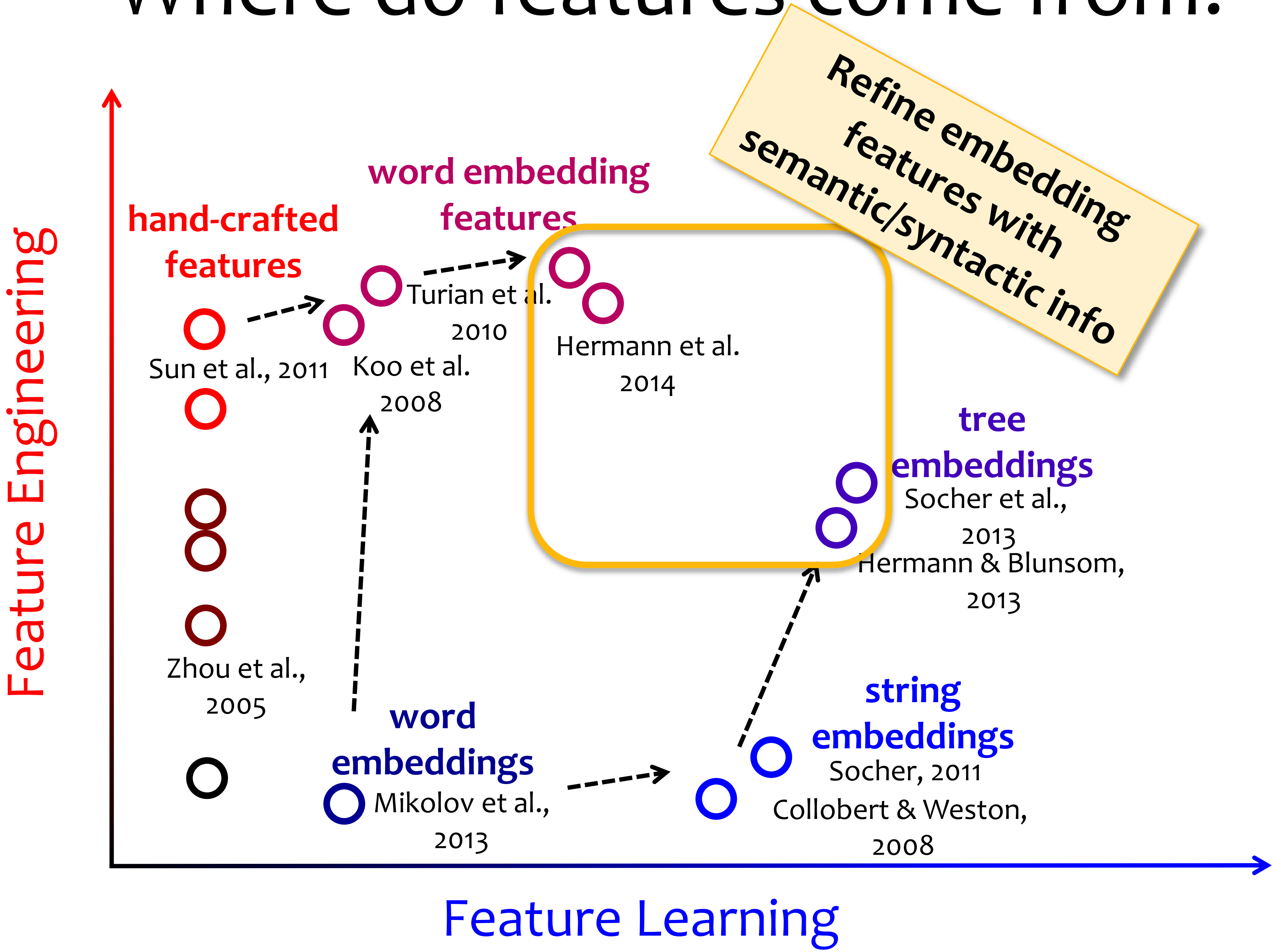
Where do features come from?



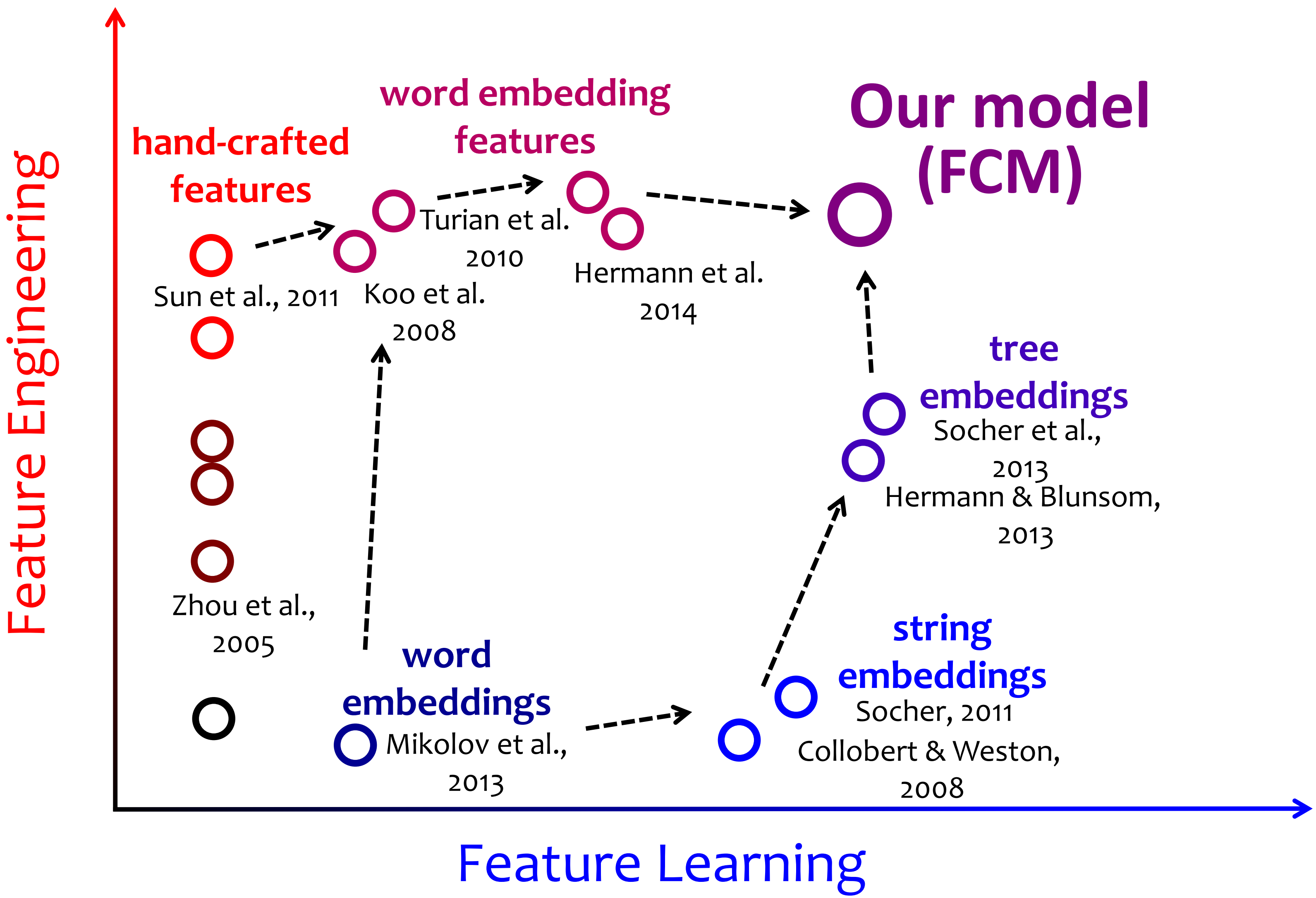
Where do features come from?



Where do features come from?



Where do features come from?



Feature-rich Compositional Embedding Model (FCM)

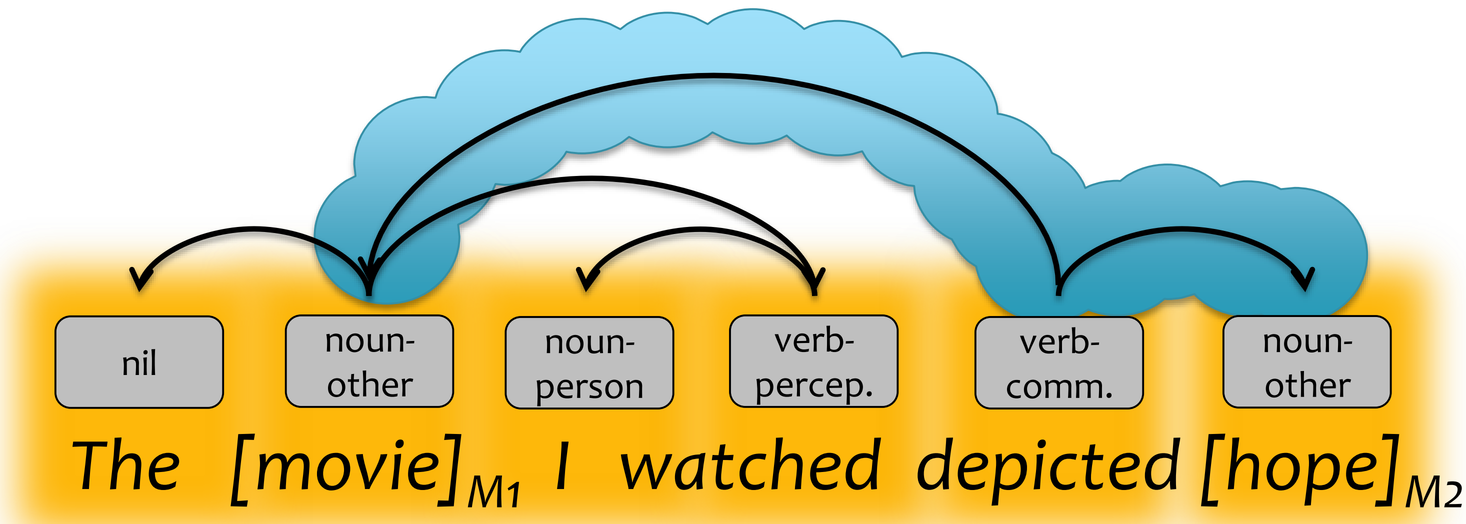
Goals for our Model:

1. Incorporate semantic/syntactic structural information
2. Incorporate word meaning
3. Bridge the gap between **feature engineering** and **feature learning** – but remain as **simple** as possible

Feature-rich Compositional Embedding Model (FCM)

Per-word Features:

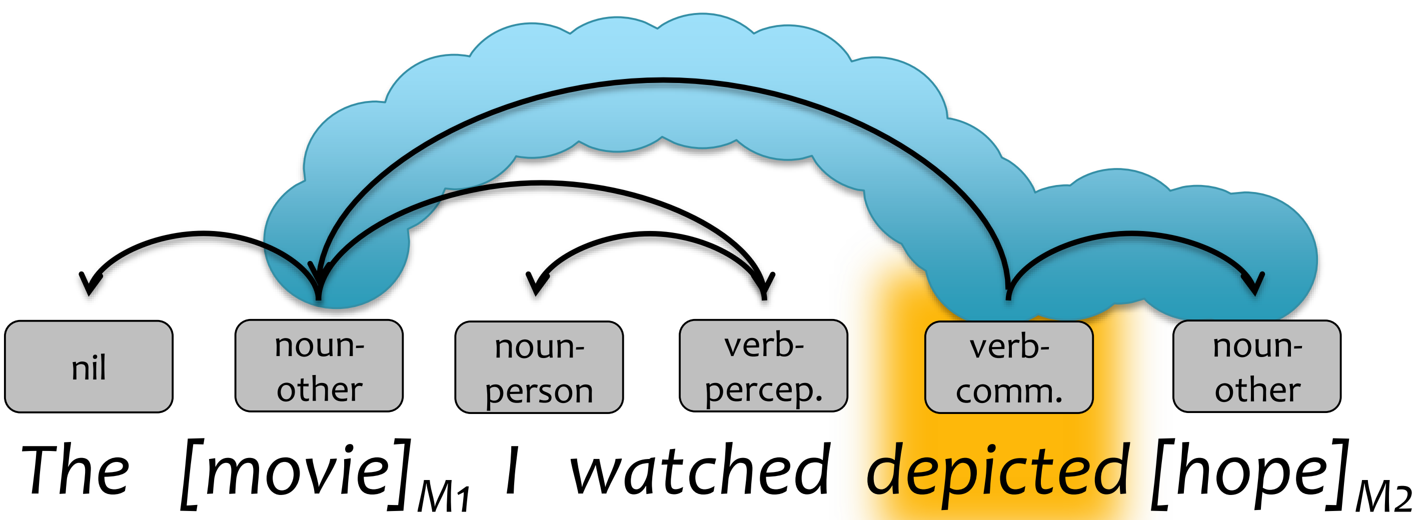
	f_1	f_2	f_3	f_4	f_5	f_6
`on-path(w_i)`	0	1	0	0	1	1
`is-between(w_i)`	0	0	1	1	1	0
`head-of-M1(w_i)`	0	1	0	0	0	0
`head-of-M2(w_i)`	0	0	0	0	0	1
`before-M1(w_i)`	1	0	0	0	0	0
`before-M2(w_i)`	0	0	0	0	1	0
…	…	…	…	…	…	…



Feature-rich Compositional Embedding Model (FCM)

Per-word Features:

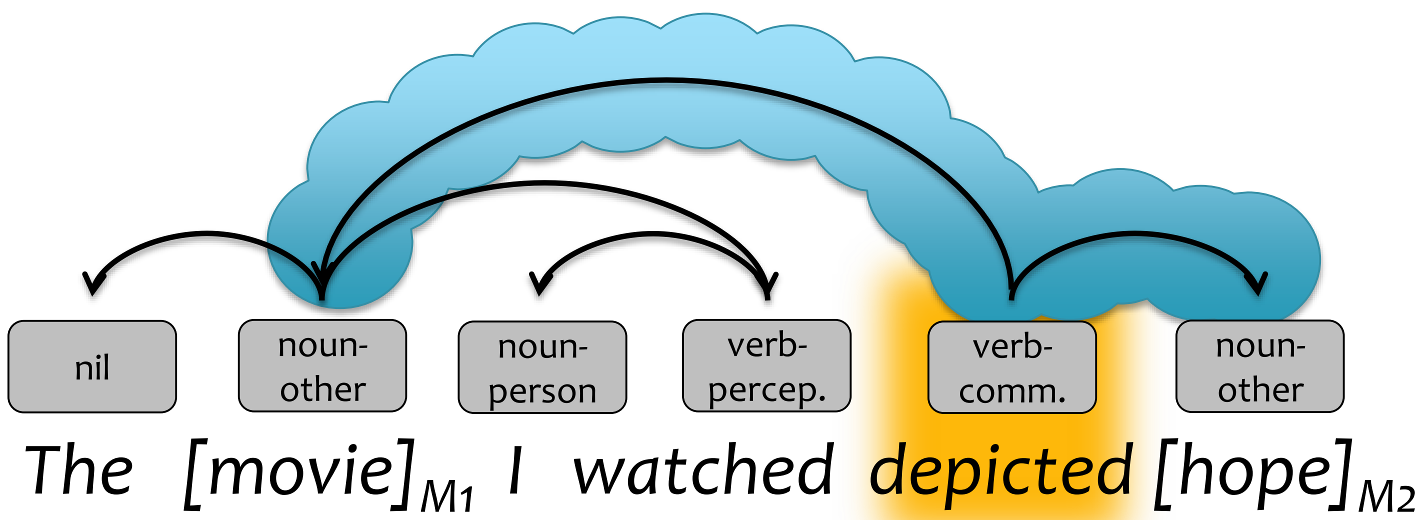
	f_5
`on-path(w_i)`	1
`is-between(w_i)`	1
`head-of-M1(w_i)`	0
`head-of-M2(w_i)`	0
`before-M1(w_i)`	0
`before-M2(w_i)`	1
…	…



Feature-rich Compositional Embedding Model (FCM)

Per-word Features: (with conjunction)

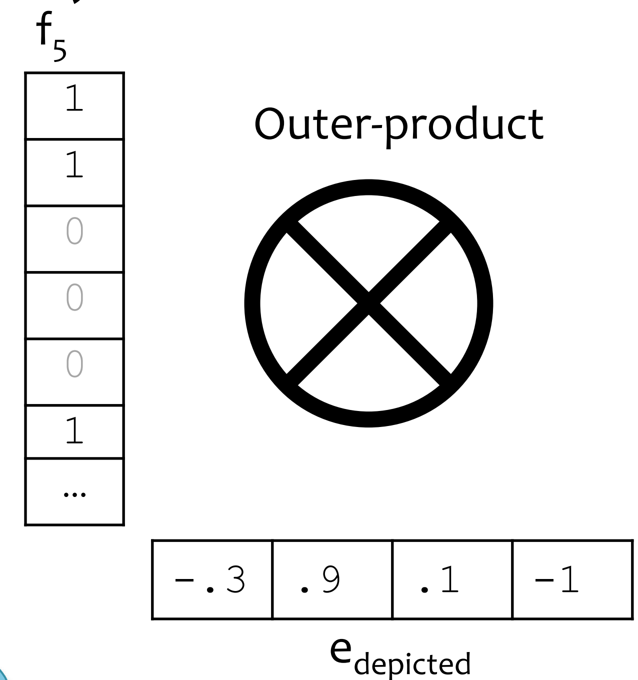
	f_5
on-path(w_i) & w_i= "depicted"	1
is-between(w_i) & w_i= "depicted"	1
head-of-M1(w_i) & w_i= "depicted"	0
head-of-M2(w_i) & w_i= "depicted"	0
before-M1(w_i) & w_i= "depicted"	0
before-M2(w_i) & w_i= "depicted"	1
…	…

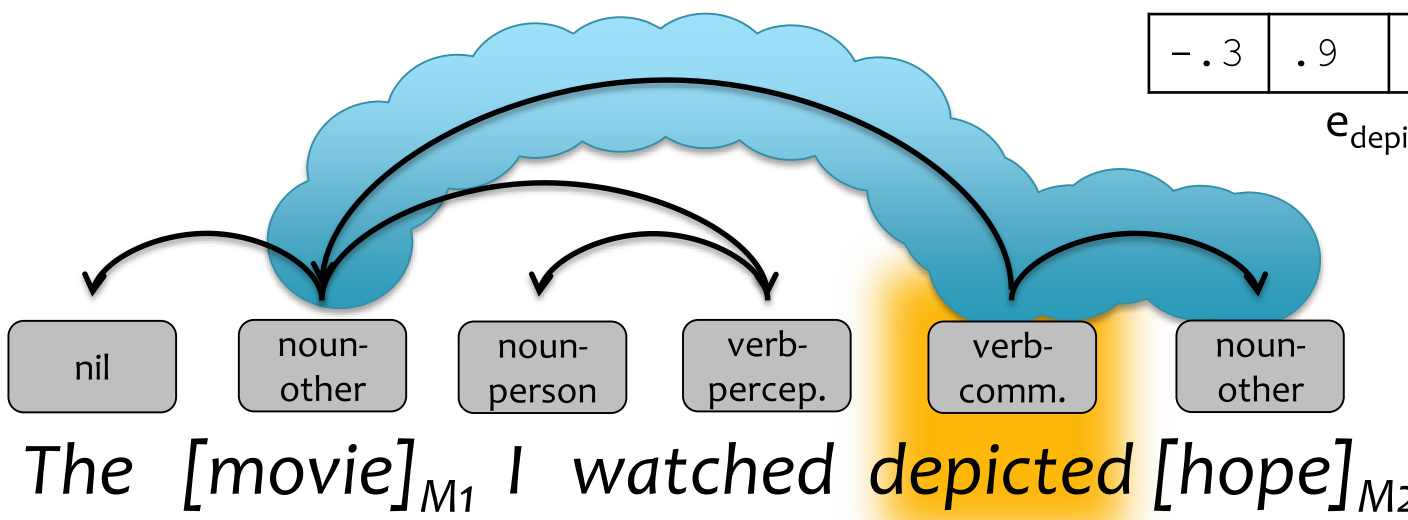


Feature-rich Compositional Embedding Model (FCM)

Per-word Features: (with soft conjunction)

```
on-path(w_i)
is-between(w_i)
head-of-M1(w_i)
head-of-M2(w_i)
before-M1(w_i)
before-M2(w_i)
    …
```





Feature-rich Compositional Embedding Model (FCM)

Per-word Features: (with soft conjunction)

```
on-path(w_i)
is-between(w_i)
head-of-M1(w_i)
head-of-M2(w_i)
before-M1(w_i)
before-M2(w_i)
   …
```

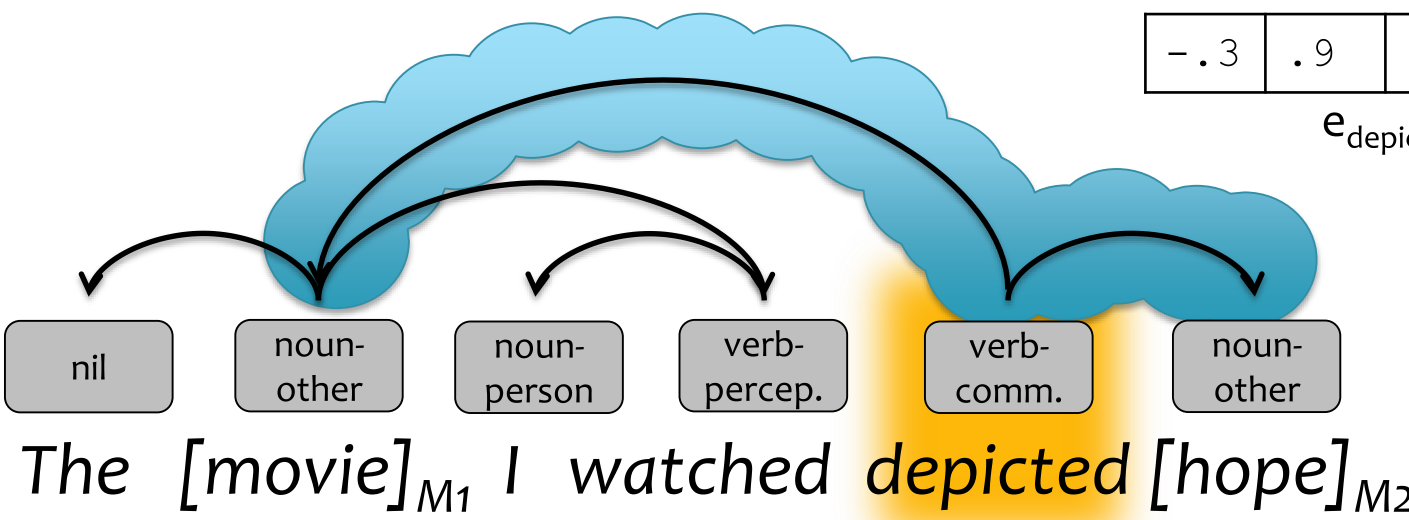
f_5

1	
1	
0	
0	
0	
1	
…	

-.3	.9	.1	-1
-.3	.9	.1	-1
-.3	.9	.1	-1
0	0	0	0
0	0	0	0
-.3	.9	.1	-1
…	…	…	…

-.3	.9	.1	-1

$e_{depicted}$



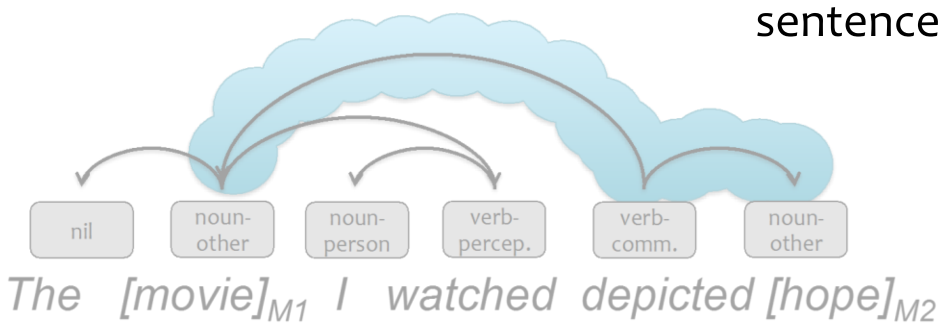
Feature-rich Compositional Embedding Model (FCM)

$$p(y|x) \propto \exp\left(T_y \odot \sum_{i=1}^{n} f_i \otimes e_{w_i} \right)$$

And finally, exponentiates and renormalizes

Then takes the dot-product with a parameter tensor

Our full model sums over each word in the sentence

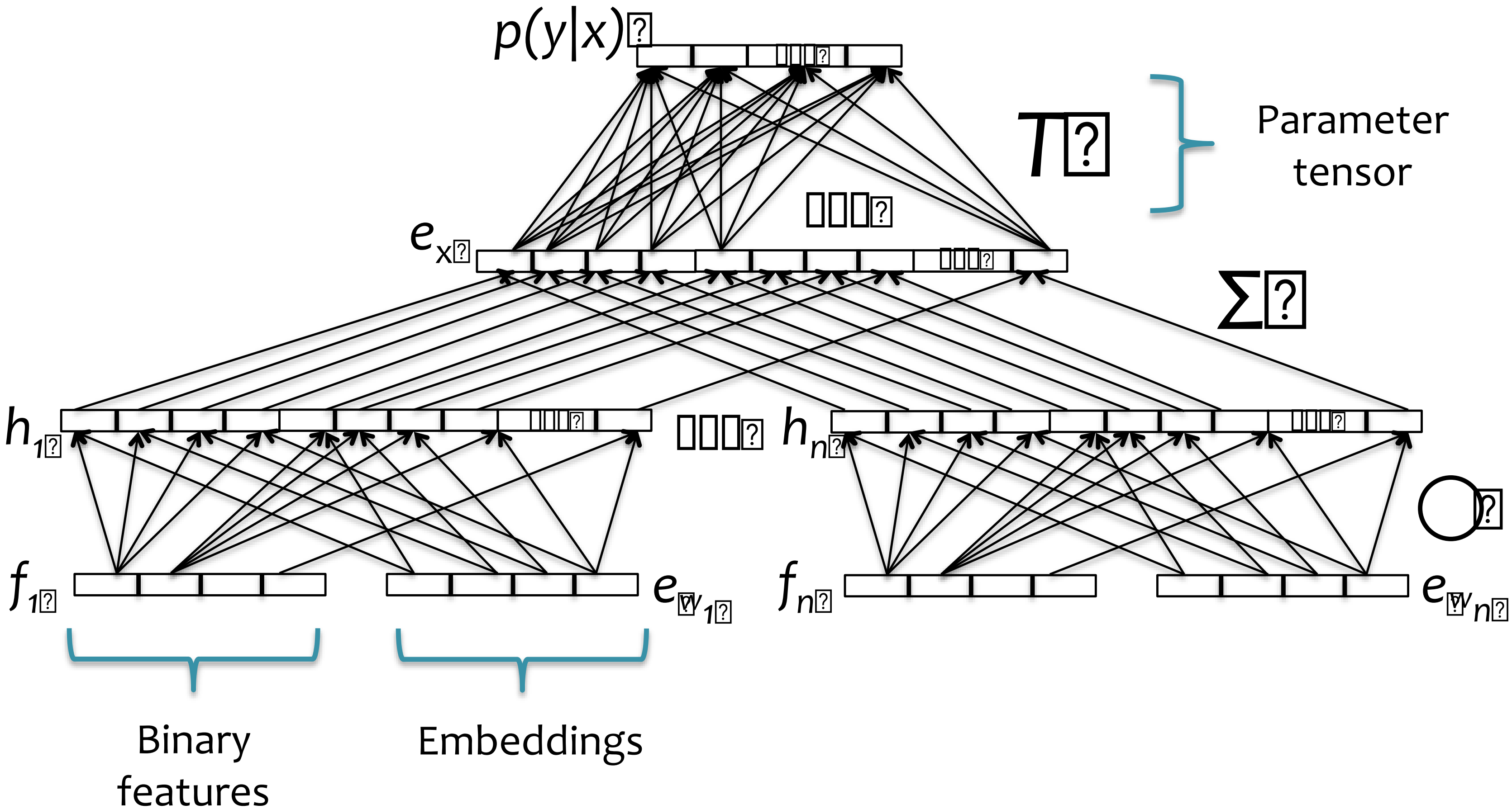


Features for FCM

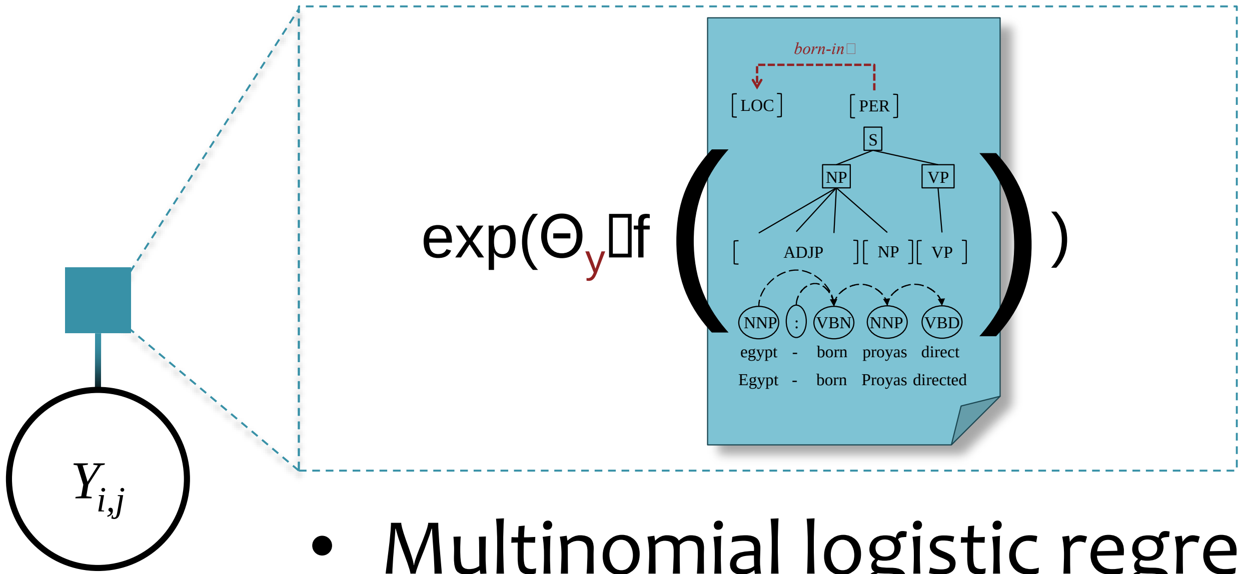
- Let **M1** and **M2** denote the *left* and *right* entity mentions
- **Our per-word Binary Features:**
  - `head of` M1
  - `head of` M2
  - `in-between` M1 `and` M2
  - `-2, -1, +1, or +2 of` M1
  - `-2, -1, +1, or +2 of` M2
  - `on dependency path between` M1 `and` M2
- **Optionally:**
  Add the entity type of M1, M2, or both

FCM as a Neural Network

- Embeddings are (optionally) treated as model parameters
- A log-bilinear model
- We initialize, then *fine-tune* the embeddings

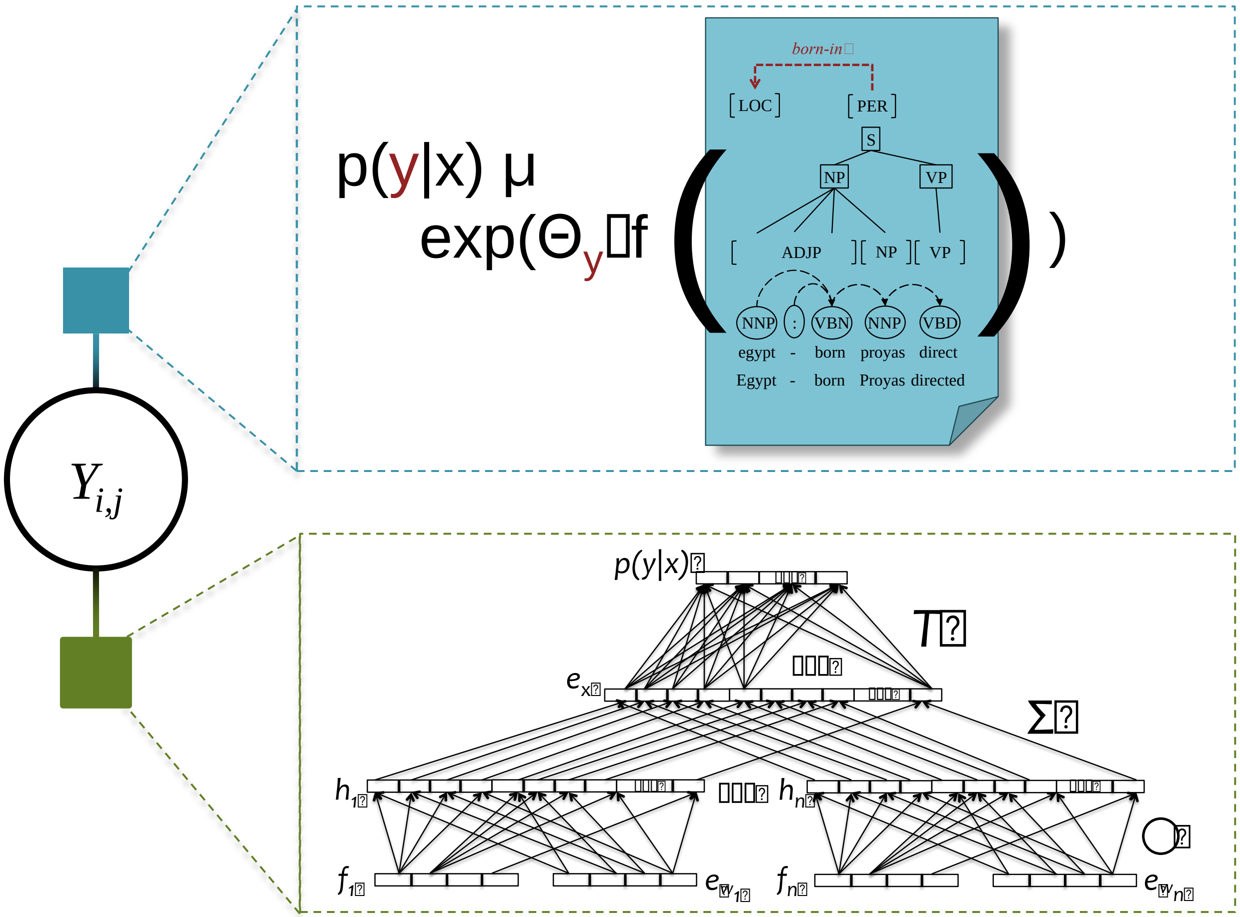


Baseline Model



- Multinomial logistic regression (*standard approach*)
- Bring in all the usual binary NLP features (Sun et al., 2011)
  - `type of the left entity mention`
  - `dependency path between mentions`
  - `bag of words in right mention`
  - …

Hybrid Model: Baseline + FCM



Product of Experts:

$$p(y|x) = \frac{1}{Z(x)} p_{\text{Baseline}}(y|x)\, p_{\text{FCM}}(y|x)$$

Experimental Setup

ACE 2005

- **Data:** 6 domains
  - Newswire (nw)
  - Broadcast Conversation (bc)
  - Broadcast News (bn)
  - Telephone Speech (cts)
  - Usenet Newsgroups (un)
  - Weblogs (wl)

- **Train:** bn+nw (~3600 relations)
  Dev: ½ of bc
  Test: ½ of bc, cts, wl`

- **Metric:** Micro F1
  (given entity mention)

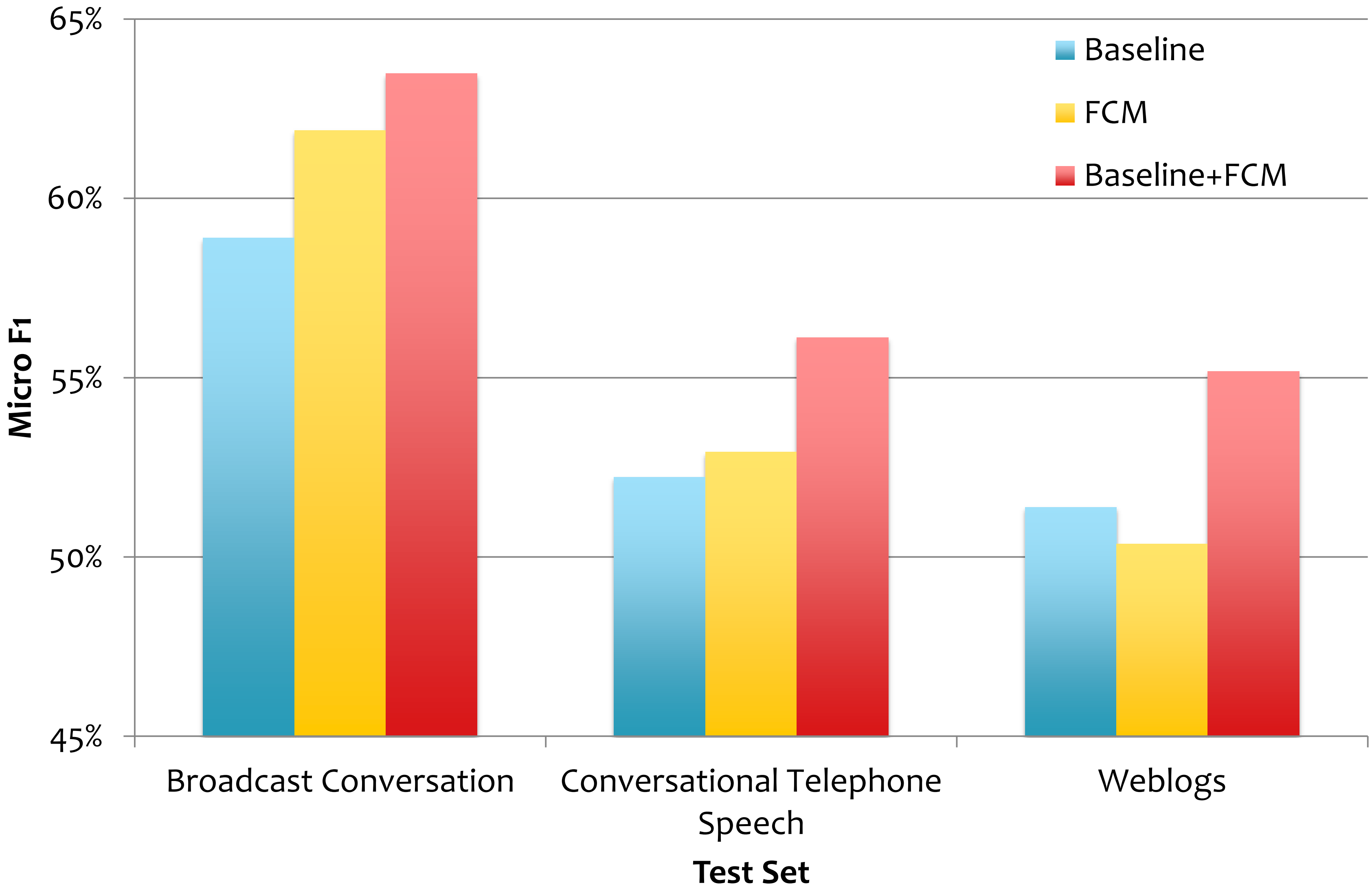
SemEval-2010 Task 8

- **Data:** Web text

- **Train:**
  Dev:
  Test:
  Standard split from shared task

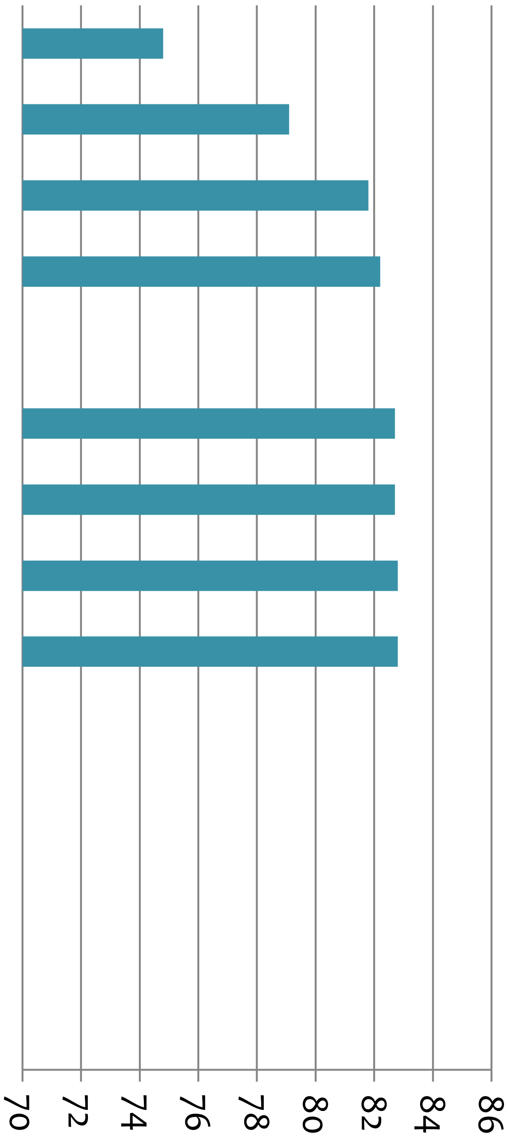
- **Metric:** Macro F1
  (given entity boundaries)

ACE 2005 Results



SemEval-2010 Results

Source	Classifier	F1
Socher et al. (2012)	RNN	74.8
Socher et al. (2012)	MVRNN	79.1
Hashimoto et al. (2015)	RelEmb	81.8
Rink and Harabagiu (2010)	SVM	82.2
	Best in SemEval-2010 Shared Task	
Zeng et al. (2014)	CNN	82.7
Santos et al. (2015)	CR-CNN (log-loss)	82.7
Liu et al. (2015)	DepNN	82.8
Hashimoto et al. (2015)	RelEmb (task-spec-emb)	82.8

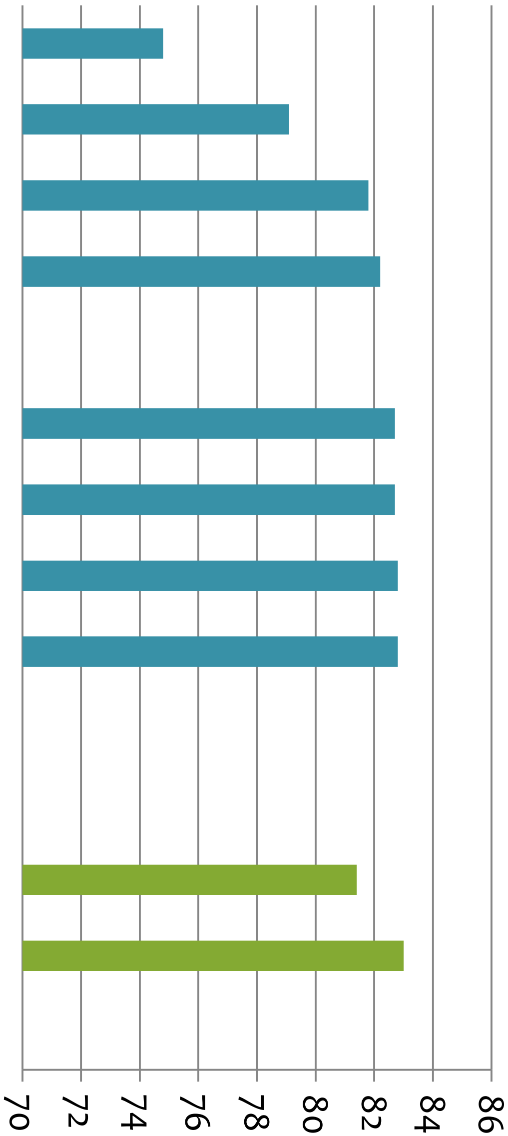


SemEval-2010 Results

Source	Classifier	F1
Socher et al. (2012)	RNN	74.8
Socher et al. (2012)	MVRNN	79.1
Hashimoto et al. (2015)	RelEmb	81.8
Rink and Harabagiu (2010)	SVM	82.2

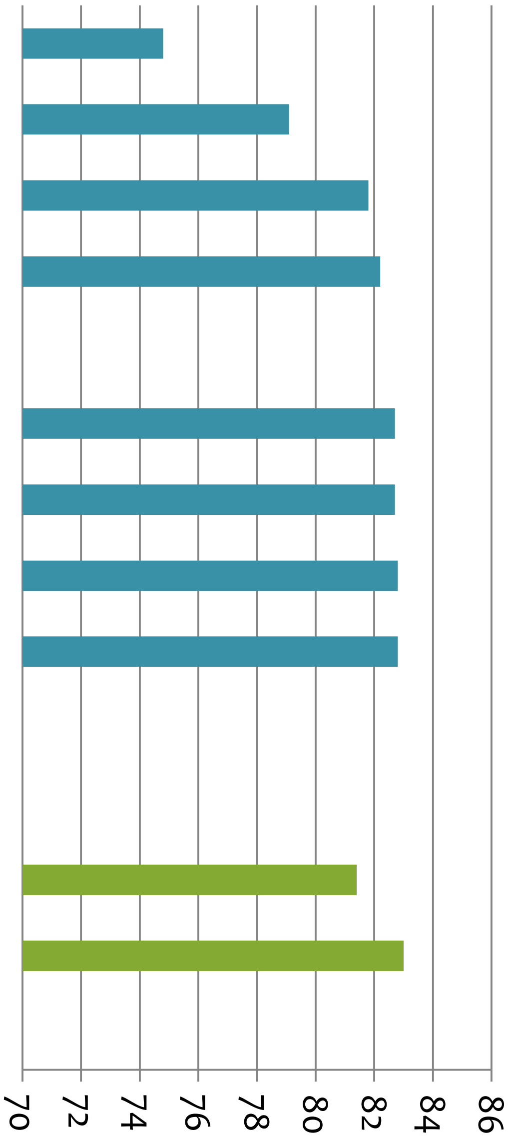
Best in SemEval-2010 Shared Task

Source	Classifier	F1
Zeng et al. (2014)	CNN	82.7
Santos et al. (2015)	CR-CNN (log-loss)	82.7
Liu et al. (2015)	DepNN	82.8
Hashimoto et al. (2015)	RelEmb (task-spec-emb)	82.8
	FCM (log-linear)	81.4
	FCM (log-bilinear)	83.0



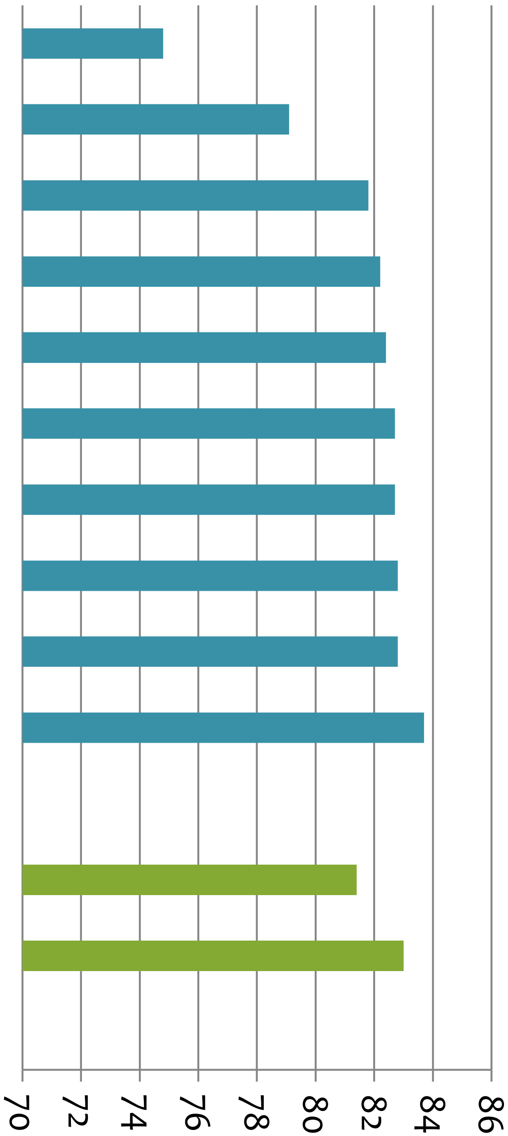
SemEval-2010 Results

Source	Classifier	F1
Socher et al. (2012)	RNN	74.8
Socher et al. (2012)	MVRNN	79.1
Hashimoto et al. (2015)	RelEmb	81.8
Rink and Harabagiu (2010)	SVM	82.2
Zeng et al. (2014)	CNN	82.7
Santos et al. (2015)	CR-CNN (log-loss)	82.7
Liu et al. (2015)	DepNN	82.8
Hashimoto et al. (2015)	RelEmb (task-spec-emb)	82.8
	FCM (log-linear)	81.4
	FCM (log-bilinear)	83.0



SemEval-2010 Results

Source	Classifier	F1
Socher et al. (2012)	RNN	74.8
Socher et al. (2012)	MVRNN	79.1
Hashimoto et al. (2015)	RelEmb	81.8
Rink and Harabagiu (2010)	SVM	82.2
Xu et al. (2015)	SDP-LSTM	82.4
Zeng et al. (2014)	CNN	82.7
Santos et al. (2015)	CR-CNN (log-loss)	82.7
Liu et al. (2015)	DepNN	82.8
Hashimoto et al. (2015)	RelEmb (task-spec-emb)	82.8
Xu et al. (2015)	SDP-LSTM (full)	83.7
	FCM (log-linear)	81.4
	FCM (log-bilinear)	83.0

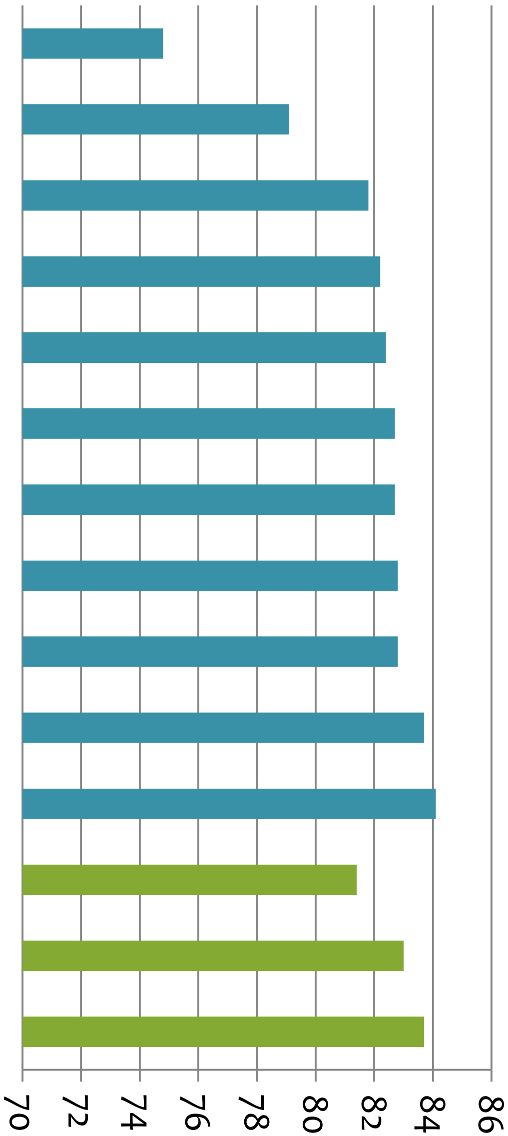


SemEval-2010 Results

Source	Classifier	F1
Socher et al. (2012)	RNN	74.8
Socher et al. (2012)	MVRNN	79.1
Hashimoto et al. (2015)	RelEmb	81.8
Rink and Harabagiu (2010)	SVM	82.2
Xu et al. (2015)	SDP-LSTM	82.4
Zeng et al. (2014)	CNN	82.7
Santos et al. (2015)	CR-CNN (log-loss)	82.7
Liu et al. (2015)	DepNN	82.8
Hashimoto et al. (2015)	RelEmb (task-spec-emb)	82.8
Xu et al. (2015)	SDP-LSTM (full)	83.7
	FCM (log-linear)	81.4
	FCM (log-bilinear)	83.0
	FCM (log-bilinear) (task-spec-emb)	83.7

SemEval-2010 Results

Source	Classifier	F1
Socher et al. (2012)	RNN	74.8
Socher et al. (2012)	MVRNN	79.1
Hashimoto et al. (2015)	RelEmb	81.8
Rink and Harabagiu (2010)	SVM	82.2
Xu et al. (2015)	SDP-LSTM	82.4
Zeng et al. (2014)	CNN	82.7
Santos et al. (2015)	CR-CNN (log-loss)	82.7
Liu et al. (2015)	DepNN	82.8
Hashimoto et al. (2015)	RelEmb (task-spec-emb)	82.8
Xu et al. (2015)	SDP-LSTM (full)	83.7
Santos et al. (2015)	CR-CNN (ranking-loss)	**84.1**
	FCM (log-linear)	81.4
	FCM (log-bilinear)	83.0
	FCM (log-bilinear) (task-spec-emb)	83.7



Takeaways

FCM bridges the gap between feature **engineering** and feature **learning**

If you are allergic to deep learning:

- Try the FCM for your task: it is **simple, easy-to-implement**, and was shown to be **effective** for two relation benchmarks

If you are a deep learning expert:

- Inject the FCM (i.e. **outer product of features and embeddings**) into your fancy deep network

Questions?

Two open source implementations:

- **Java:** (Within the Pacaya framework) https://github.com/mgormley/pacaya
- **C++:** (From our NAACL 2015 paper on LRFCM) https://github.com/Gorov/ERE_RE