HIT-SCIR

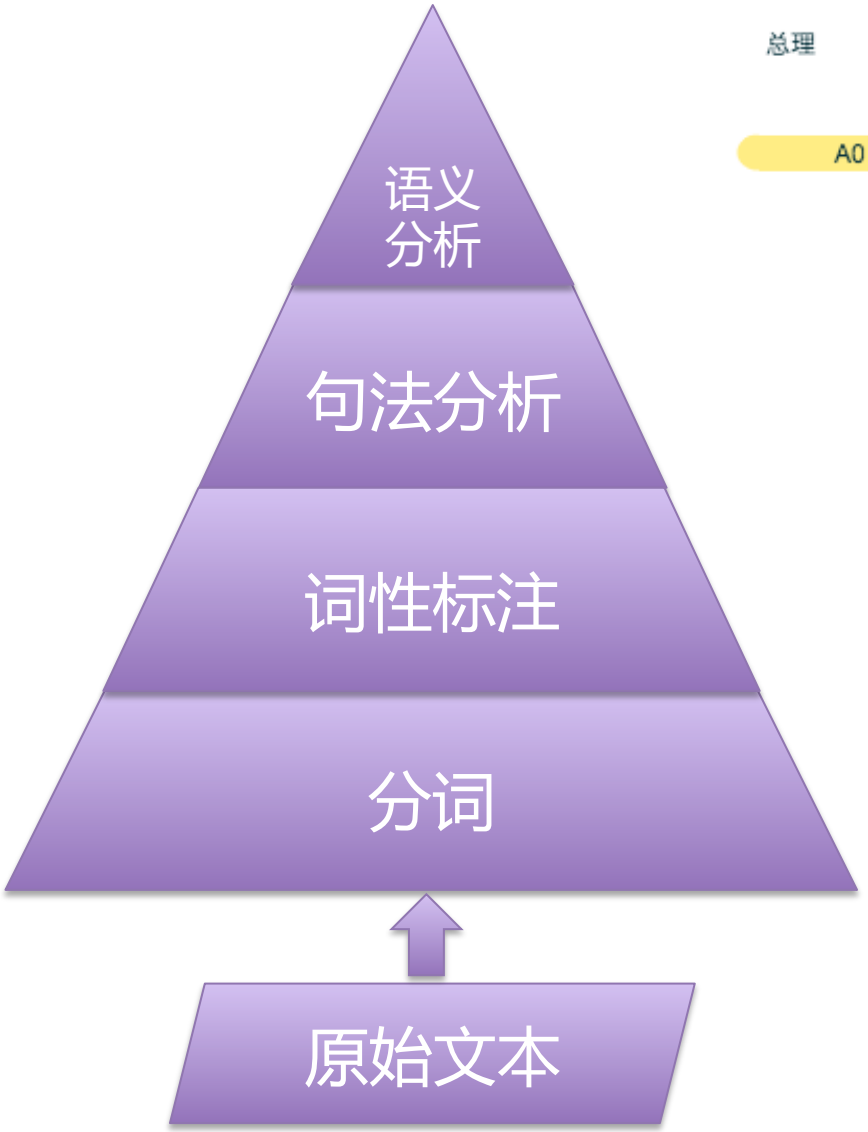
Deep Learning for NLP

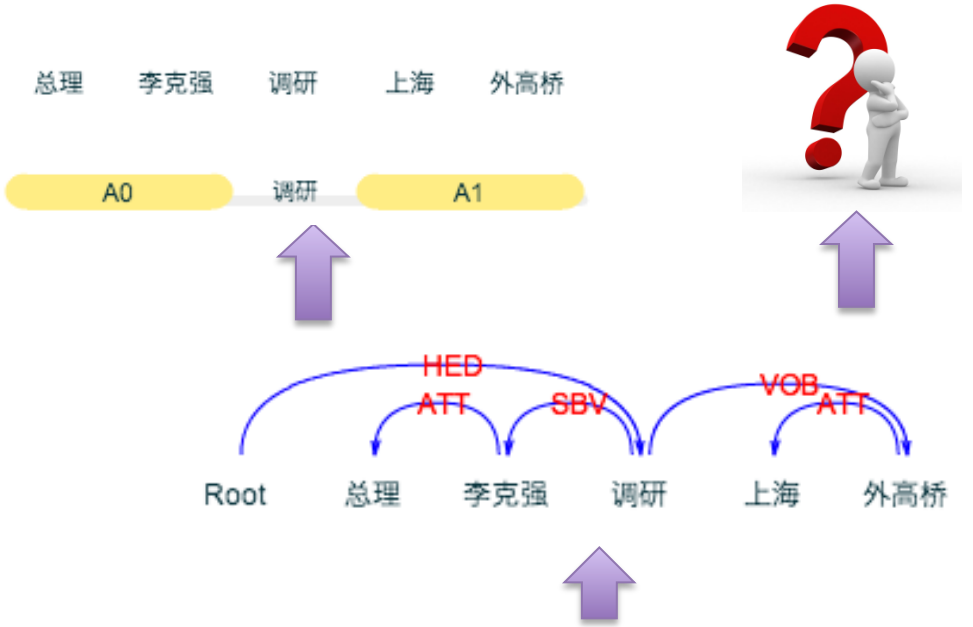
Wanxiang Che
Research Center for Social Computing and Information Retrieval
Harbin Institute of Technology
2015-5-6

自然语言处理的目标

- 语言是思维的载体，是人类交流思想、表达情感最自然、最直接、最方便的工具
- 人类历史上以语言文字形式记载和流传的知识占知识总量的80%以上
- 高效、高精度自然语言处理（NLP）系统，致力于让机器理解人类的语言

语言分析任务





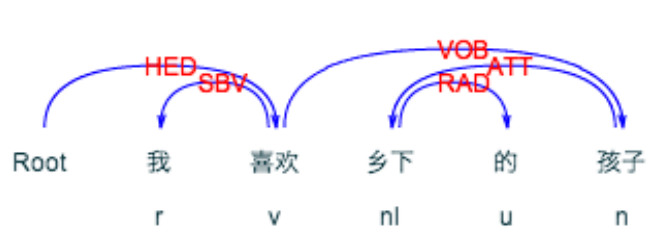
总理/n 李克强/nh 调研/v 上海/ns 外高桥/ns

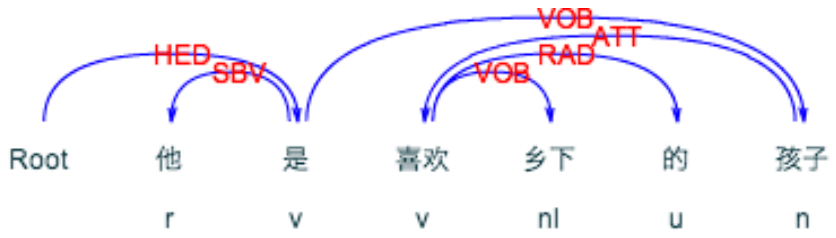
总理 李克强 调研 上海 外高桥

总理李克强调研上海外高桥

句法分析

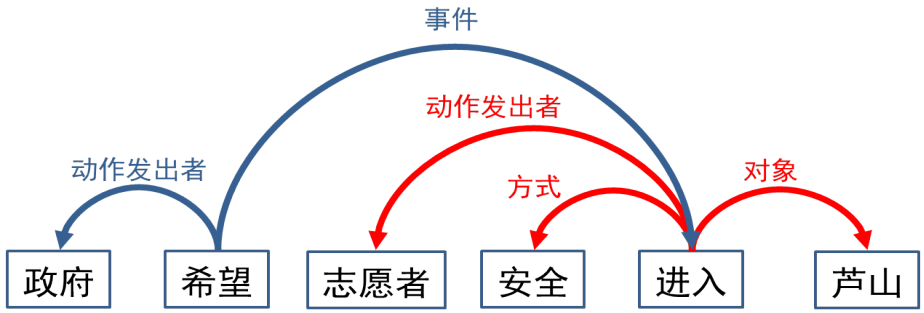
- 将输入文本从序列形式转化为树状结构，刻画句子内部的句法关系，是自然语言处理的核心问题
- 常见的句法关系
  - 主、谓、宾、定、状、补、...
- 句法歧义
  - 如：“喜欢乡下的孩子”





语义依存分析

- 语义依存图表示（句法语义结合体）



- 语义依存关系不因句子表层形式而变化

张三吃了苹果

张三把苹果吃了

苹果被张三吃了

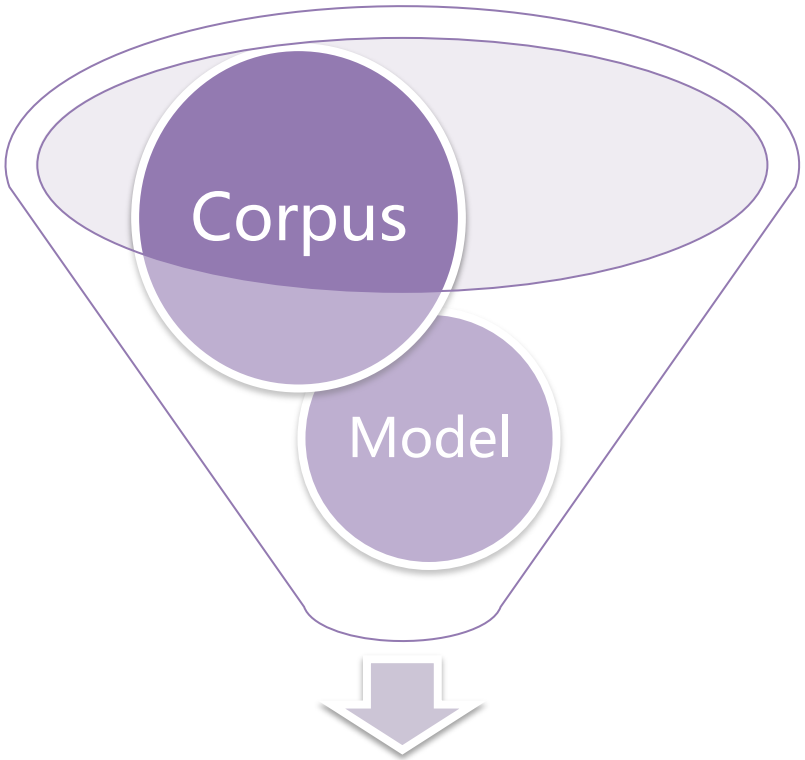
} 吃(张三，苹果)

自然语言处理的主要困难

- 语言和语义之间的一对多关系
  - 苹果、小米
  - 严守一把手机关了。
- 理解语言通常需要背景知识和推理
  - 张三打了李四，他倒下了。
- 语言动态化、非规范化
  - 超女、非典
  - 腐败、杯具
  - 为森么

Methodology

- Statistical NLP



Chinese Corpus

Name	Resource	Size	For Tasks	InstitutionI
People Daily	98, 2000 year People Daily	10M words	Word Segment, POS Tagging	PKU
Chinese Dependency Treebank	People Daily	50K sent	Dependency Parsing	HIT
Weibo Treebank	weibo.com	50K sent	Parsing	HIT, DataTang
Chinese Propbank	Newswire	10K sent	Semantic Role Labeling	University of Pennsifania

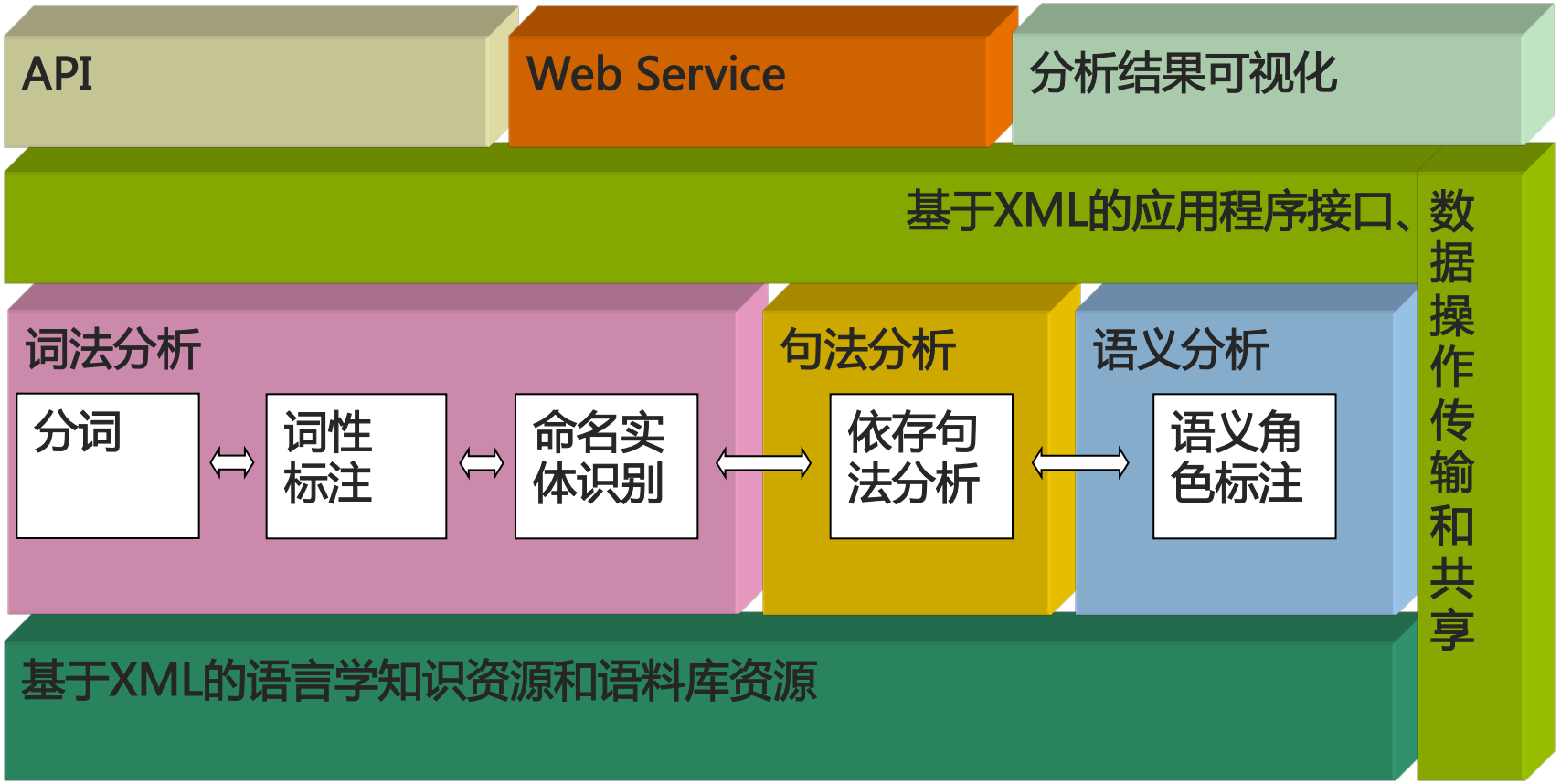
NLP Models

<table>
<tr><th>Task</th><th>Problem</th><th>Model</th><th>Optimizing</th><th>Decoding</th></tr>
<tr><td rowspan="2">Word Seg,
POS Tagging,
NER</td><td rowspan="2">Sequence
Labeling</td><td>CRF</td><td>L-BFGS,
SGD</td><td>Viterbi</td></tr>
<tr><td rowspan="2">Structure
Perceptron</td><td rowspan="2">MIRA, PA,
AP</td><td>Viterbi,
Beam Search</td></tr>
<tr><td rowspan="2">Dependency
Parsing</td><td>Graph-based</td><td>MST</td></tr>
<tr><td>Transition-
based</td><td colspan="2" rowspan="2">Logistic Regression, SVM</td><td>Shift-reduce</td></tr>
<tr><td>Semantic
Role Labeling</td><td>Classification</td><td>ILP</td></tr>
</table>

语言技术平台

- Language Technology Platform, LTP
- 意义
  - 支撑各类应用课题的研究
  - 研究成果的积累和转化
- 历程
  - 2003年：开始研制
  - 2006年9月：对外共享
  - 2011年6月：开源
  - 2013年9月：提供“语言云”服务
  - 2014年11月：与讯飞联合推出“哈工大讯飞语言云”
- 获奖
  - 2010年获钱伟长中文信息处理科学技术一等奖

LTP 系统框架


API
Web Service
分析结果可视化
基于XML的应用程序接口、数据操作传输和共享
词法分析
分词
词性标注
命名实体识别
句法分析
依存句法分析
语义分析
语义角色标注
基于XML的语言学知识资源和语料库资源

LTP 特色

- 丰富的高质量语言分析模块
- 基于XML的内部数据表示
- 基于浏览器的可视化工具
- 调用方式便捷灵活
- 开源

各模块性能

模块	算法	准确率	内存	速度
分词	Structured Perceptron	F1 = 97.2%	119M	177KB/s
词性标注		Acc = 97.8%	291M	106KB/s
命名实体识别		F1 = 93.6%	21M	150KB/s
依存句法分析	2-order MST	LAS/UAS = 81.1/84.2	974M	0.8KB/s
语义角色标注	MaxEnt(L1) + ILP	F1 = 77.9	94M	3.2KB/s

分词盲测结果（2014年9月）

- 与另外两个流行分词系统对比
- 随机抽取新闻、微博文本
- 对近2,000句分词结果不同文本进行比较
- 互联网用户盲测

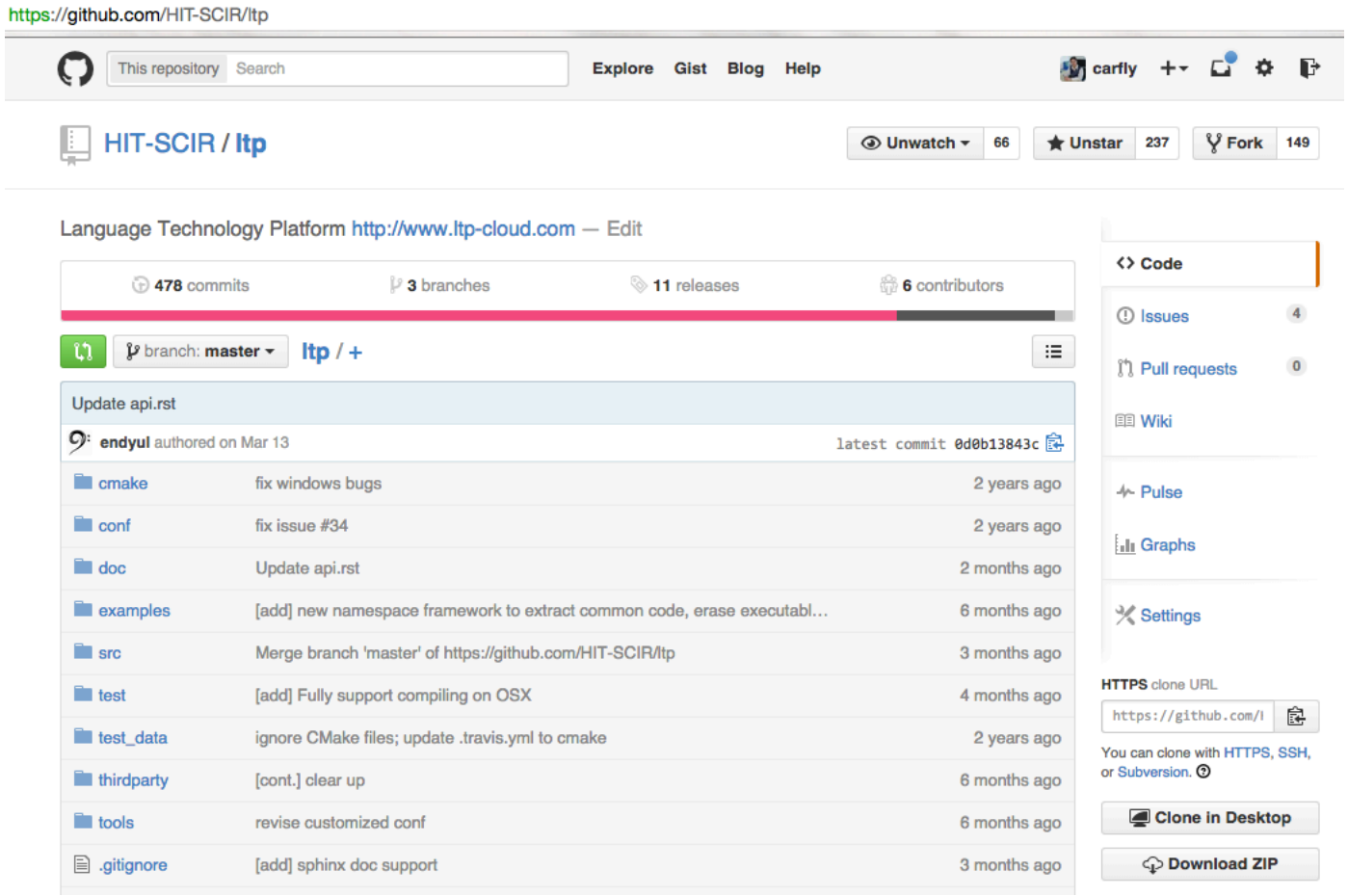
	系统一	系统二
LTP更好	66.01%	50.84%
LTP不好	17.48%	29.55%
两者都好	4.58%	5.30%
都不好	11.93%	14.31%

CoNLL 2009 国际句法、语义分析评测

Rank	System	Average	Catalan	Chinese	Czech	English	German	Japanese	Spanish
1	Che	82.64	81.84	@ 76.38	@ 83.27	87.00	@ 82.44	@ 85.65	81.90
2	Chen	82.52	@ 83.01	76.23	80.87	@ 87.69	81.22	85.28	@ 83.31
3	Merlo	82.14	82.66	76.15	83.21	86.03	79.59	84.91	82.43
4	Bohnet	80.85	80.44	75.91	79.57	85.14	81.60	82.51	80.75
5	Asahara	78.43	75.91	73.43	81.43	86.40	69.84	84.86	77.12
6	Brown	77.27	77.40	72.12	75.66	83.98	77.86	76.65	77.21
7	Zhang	76.49	75.00	73.42	76.93	82.88	73.76	78.17	75.25
8	Dai	73.98	72.09	72.72	67.14	81.89	75.00	80.89	68.14
9	Lu Li	73.97	71.32	65.53	75.85	81.92	70.93	80.49	71.72
10	Lluís	71.49	56.64	66.18	75.95	81.69	72.31	81.76	65.91
11	Vallejo	70.81	73.75	67.16	60.50	78.19	67.51	77.75	70.78
12	Ren	67.81	59.42	75.90	60.18	77.83	65.77	77.63	57.96
13	Zeman	51.07	49.61	43.50	57.95	50.27	49.57	57.69	48.90

开源（2011年6月）

- https://github.com/HIT-SCIR/ltp



首届语言技术平台用户大会

- 2014年10月31日
  - 参会人数：近100位
  - 地点：北京西郊宾馆

语言技术平台合作企业

Wanxiang Che, Zhenghua Li, Ting Liu. LTP: A Chinese Language Technology Platform. In Proceedings of the **Coling 2010**: Demonstrations. 2010.08, pp13-16, Beijing, China.

语言云（2013年9月）

- 全称“语言技术平台云”
  - 基于云计算技术的中文自然语言处理服务平台
  - http://www.ltp-cloud.com/

7行代码分析自然语言

```
import urllib2, urllib, sys
uri_base = http://api.ltp-cloud.com/analysis/?
api_key  = "YourAPIKey"
text     = urllib.quote("我爱北京天安门")
format   = sys.argv[1]
url = "{}api_key={}&text={}&format={}&pattern=all".format(uri_base, api_key, text, format)
print urllib2.urlopen(url).read()
```

- More Language and Documents
  - https://github.com/HIT-SCIR/ltp-cloud-api-tutorial

哈工大讯飞语言云（2014年11月）

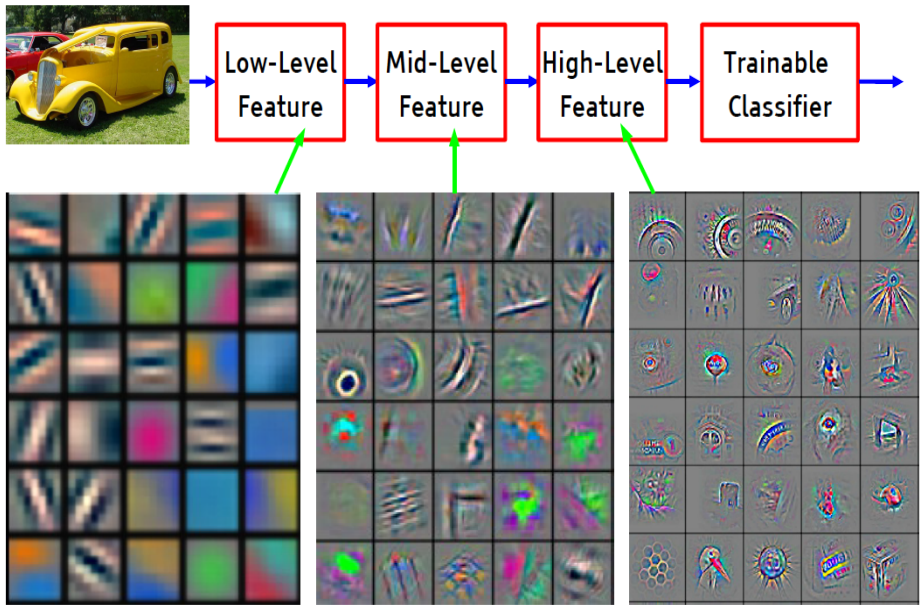
- 更稳定、高效的服务
  - 结合讯飞“语音云”云服务运维经验
- 更丰富的功能
  - 针对不同用户给出个性化的分析结果
  - 更多种类的命名实体识别
- 更好的支持移动端开发

当前NLP方法面临的问题

- 标注数据不足引起数据稀疏
- 需要细致的特征工程
- 多层处理带来的错误蔓延
- 处理速度较慢

Deep Learning

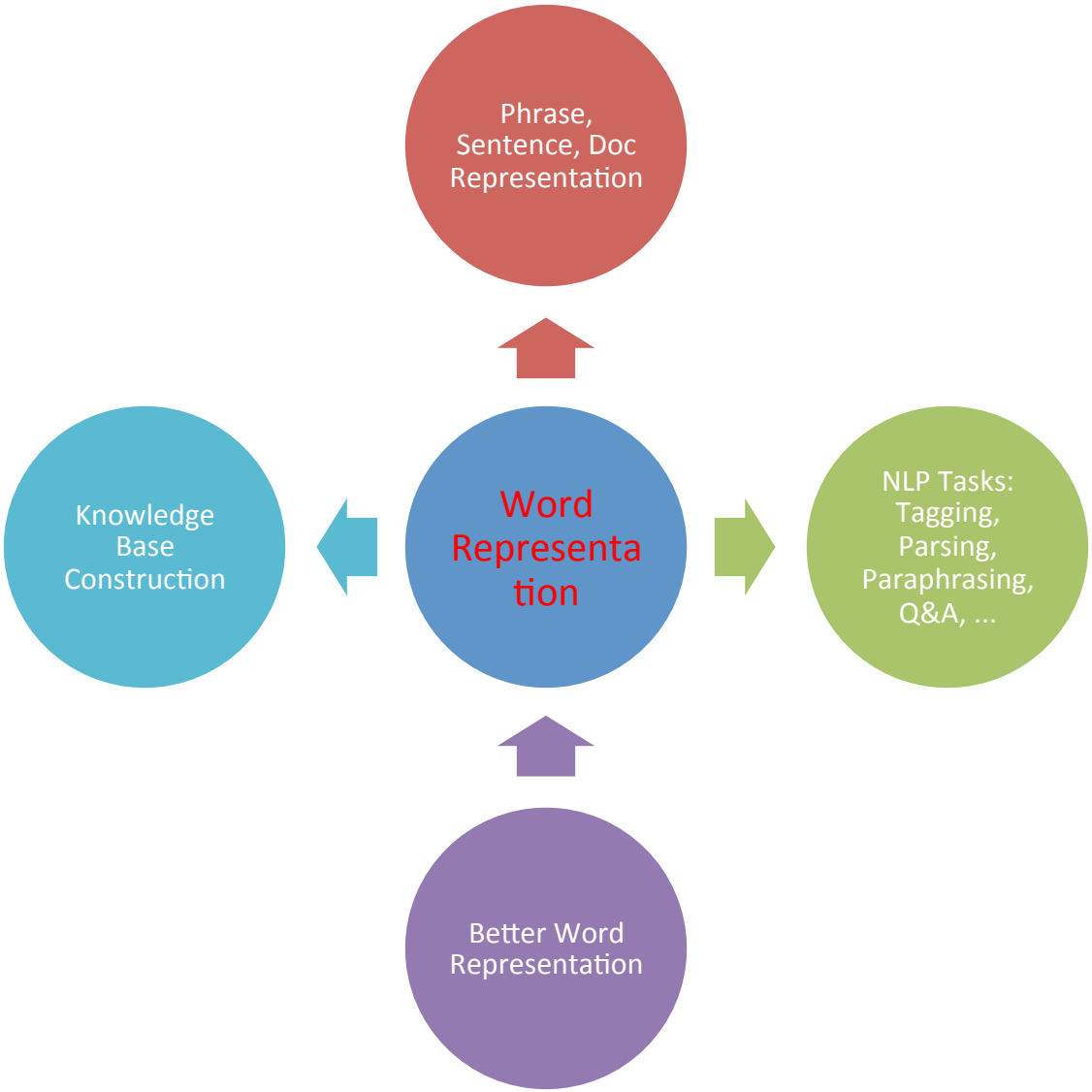
- Learning Hierarchical Representations
- It's deep if it has more than one stage of non-linear feature transformation
  - Successful in image, video and speech processing



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

Deep Learning for NLP

Deep Learning for NLP



Typical Approaches

- 1-hot representation: basis of bag-of-word model
  - high-dimensional, sparse, binary

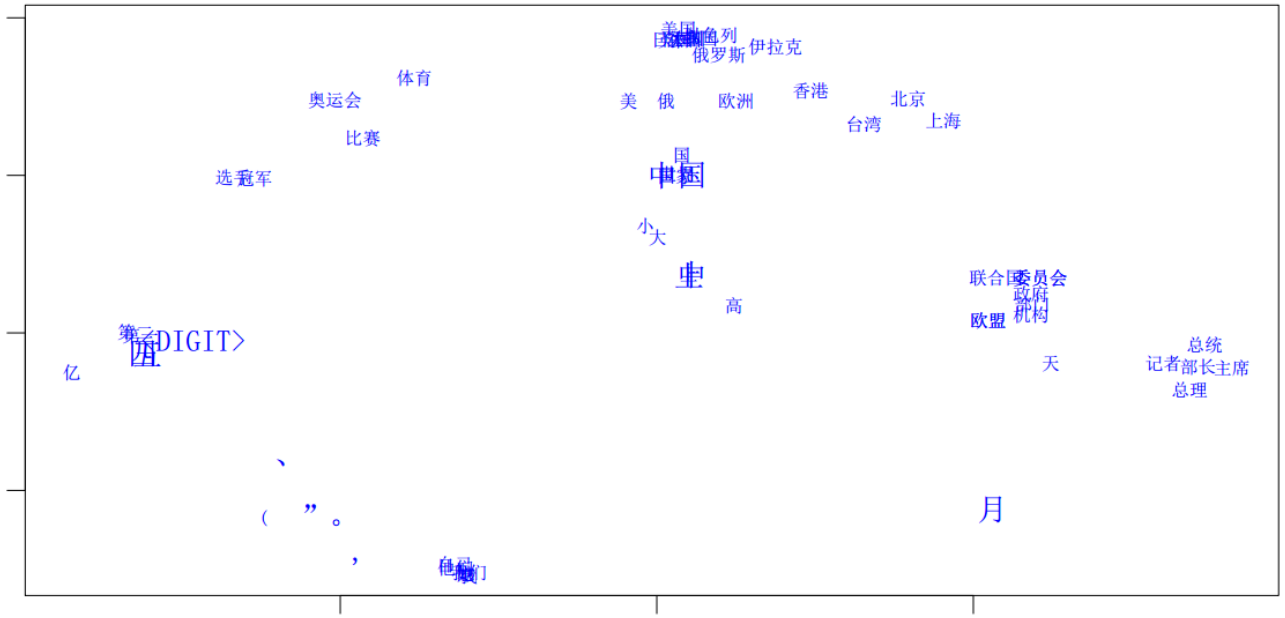
star [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, …]

sun [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, …]

sim(star, sun) = 0

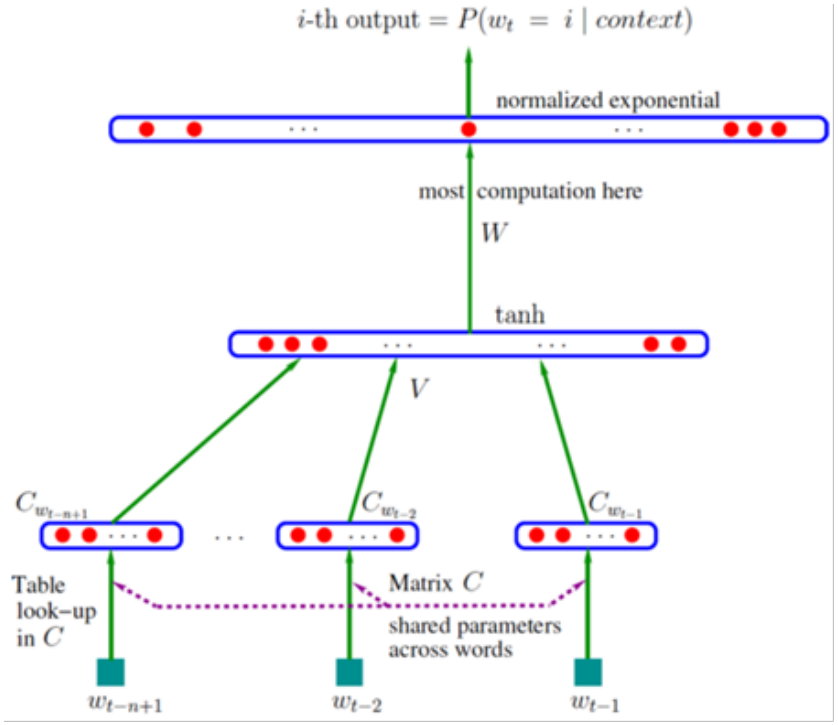
Distributed Word Representation

- Mapping words to low dimensional, dense, continuous vectors (word embedding)
- Property
  - Similar words situate clsoe positions



Neural Network Language Models (NNLM)

- Feed Forward (Bengio et al. 2003)



- **Maximum-Likelihood Estimation**
- Back-propagation
- Input: $(n-1)$ embeddings

$$P(w_t = k | w_{t-n+1}, \ldots w_{t-1}) = \frac{e^{a_k}}{\sum_{l=1}^{N} e^{a_l}}$$

$$a_k = b_k + \sum_{i=1}^{h} W_{ki} \tanh(c_i + \sum_{j=1}^{(n-1)d} V_{ij} x_j)$$

$$L(\theta) = \sum_t \log P(w_t | w_{t-n+1}, \ldots w_{t-1})$$

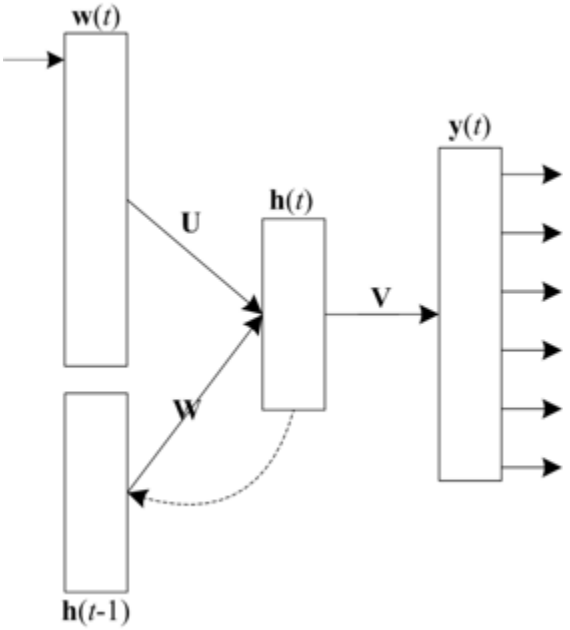
Recurrent Neural Network Language Model (RNNLM)

- Mikolov et al. (2010-2013)
  - Compute:

$$\mathbf{h}(t) = f(\mathbf{U}\mathbf{w}(t) + \mathbf{W}\mathbf{h}(t-1))$$

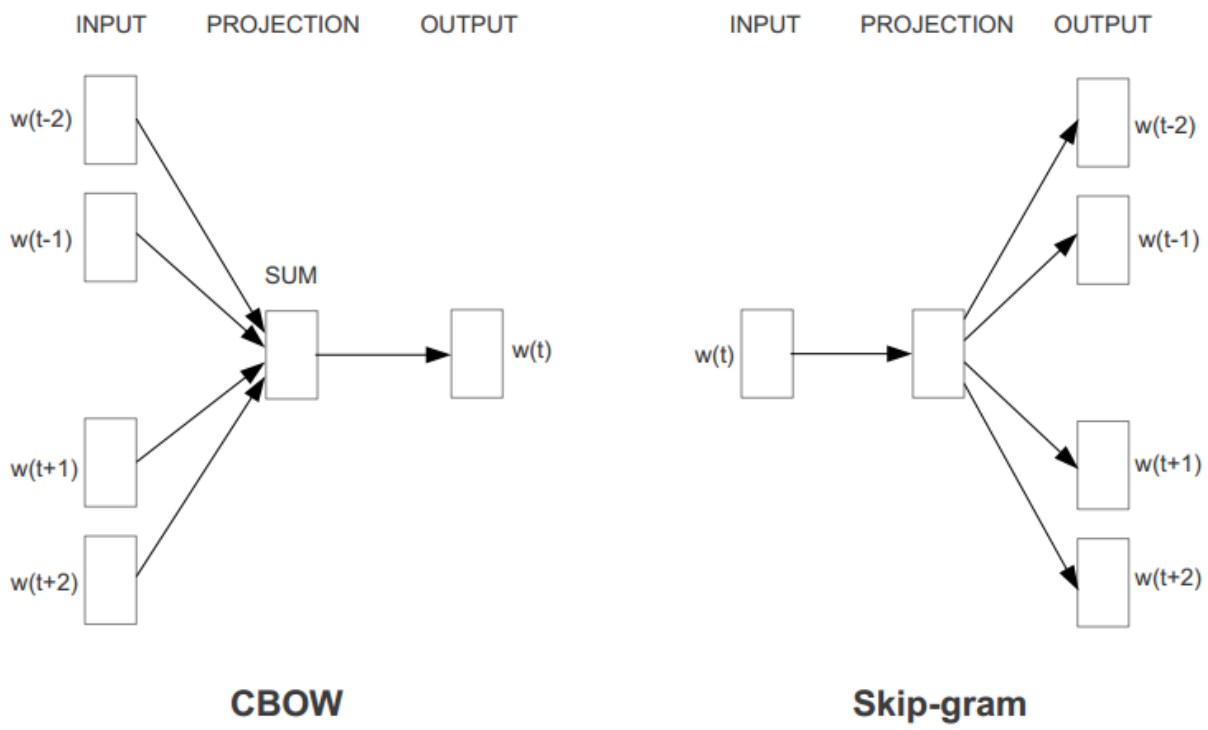
$$\mathbf{y}(t) = g(\mathbf{V}\mathbf{h}(t))$$

  - U is the Embedding Matrix

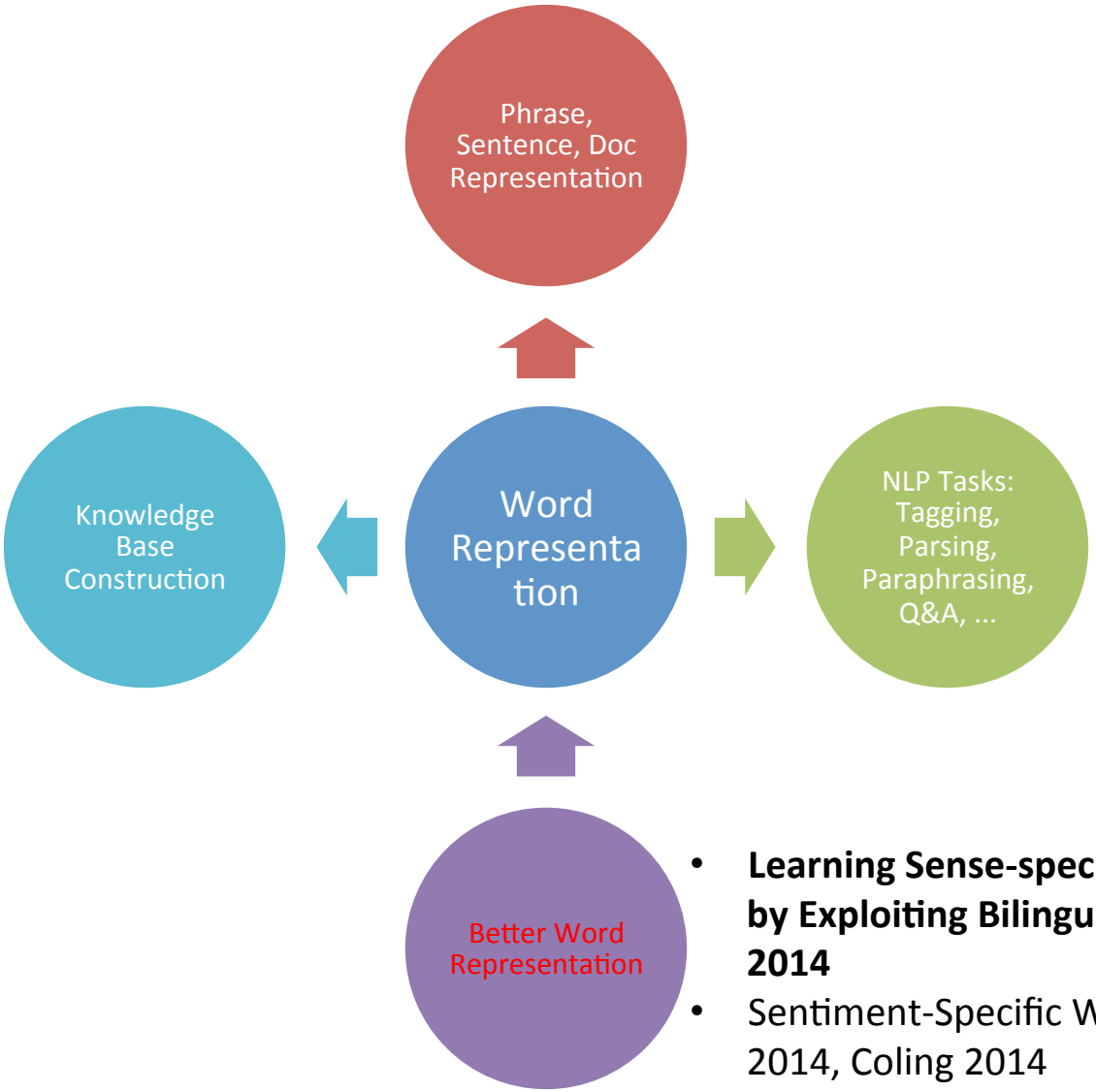


Word2vec

- CBOW and Skip-Gram (Mikolov et al. 2013)
  - https://code.google.com/p/word2vec/



Deep Learning for NLP



- **Learning Sense-specific Word Embedding by Exploiting Bilingual Resources, Coling 2014**
- Sentiment-Specific Word Embedding, ACL 2014, Coling 2014

Word and Sense Mapping

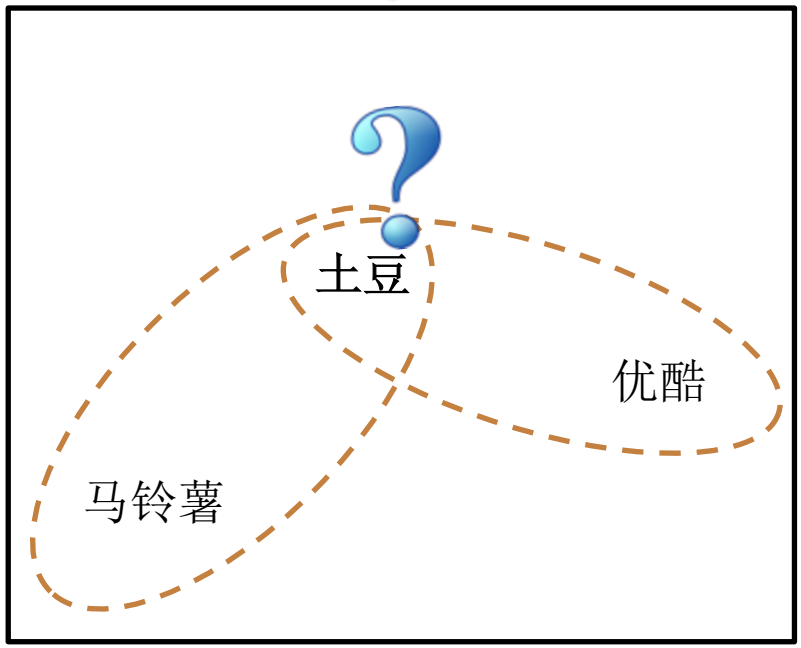
Sense space

Word space

土豆

马铃薯

Word Embedding

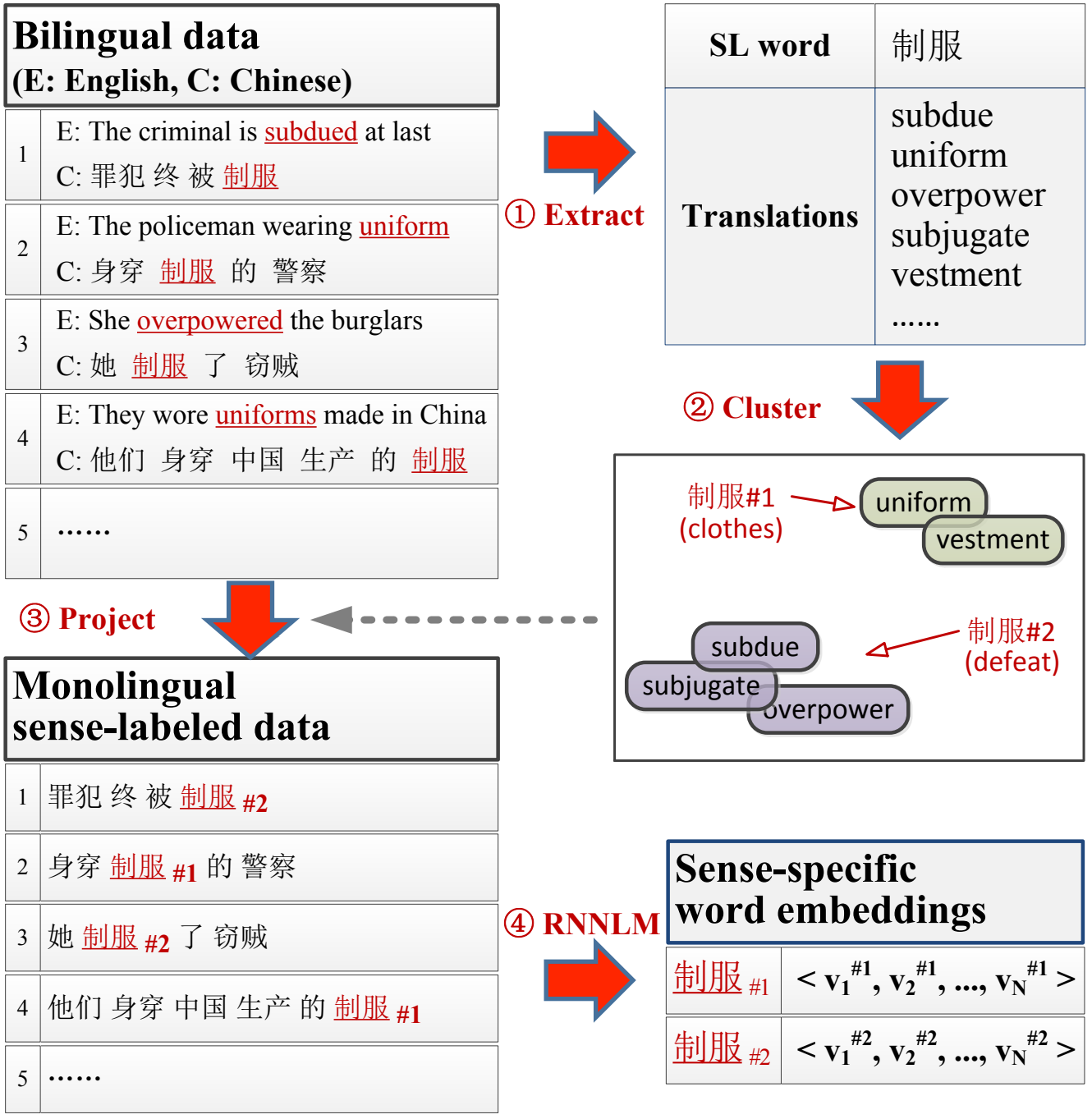


Related Work

- Huang et al. (2012), Reisinger and Mooney (2010)
  - Learning Multiple-prototype Word Embeddings
    - embeddings for each word
    - Context clustering method
  - Problem
    - Inexactly clustering
    - Ignoring the fact that the number of senses of different words are varied

Our Approach

- Represent words with **sense-specific** embeddings
- Word sense induction using a **bilingual approach**
- Train embeddings on the sense-tagged corpus


Bilingual data
(E: English, C: Chinese)
1
E: The criminal is subdued at last
C: 罪犯 终 被 制服
2
E: The policeman wearing uniform
C: 身穿 制服 的 警察
3
E: She overpowered the burglars
C: 她 制服 了 窃贼
4
E: They wore uniforms made in China
C: 他们 身穿 中国 生产 的 制服
5
……
① Extract
SL word
制服
Translations
subdue
uniform
overpower
subjugate
vestment
……
② Cluster
制服#1
(clothes)
uniform
vestment
制服#2
(defeat)
subdue
subjugate
overpower
③ Project
Monolingual
sense-labeled data
1
罪犯 终 被 制服 #2
2
身穿 制服 #1 的 警察
3
她 制服 #2 了 窃贼
4
他们 身穿 中国 生产 的 制服 #1
5
……
④ RNNLM
Sense-specific
word embeddings
制服 #1
< v1#1, v2#1, …, vN#1 >
制服 #2
< v1#2, v2#2, …, vN#2 >

Word Similarity Evaluation

- A manually constructed Polysemous Word Similarity Dataset
- Measurement
  - Spearman Correlation
  - Kendall Correlation
- Quantitative Evaluation

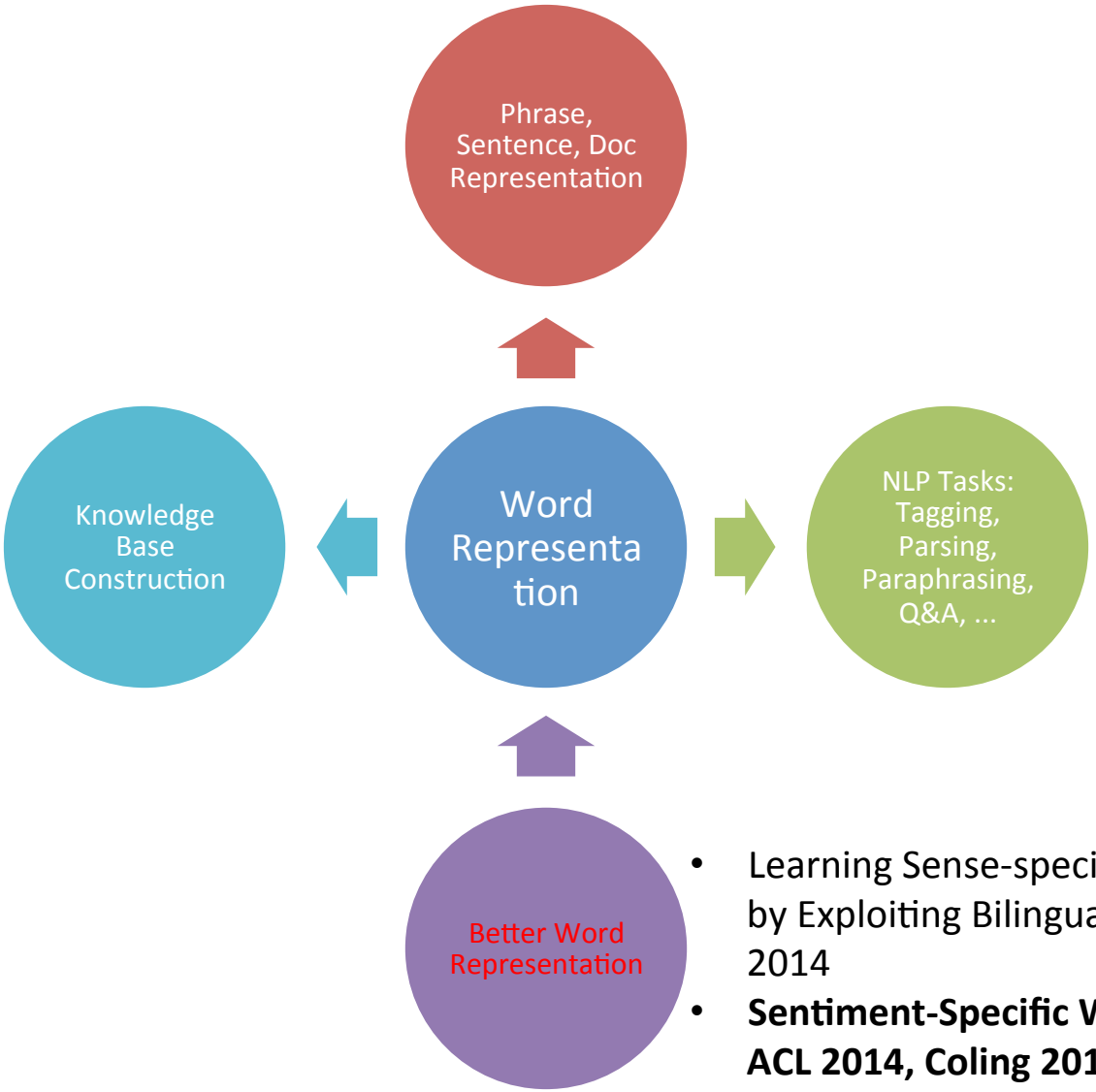
System	MaxSim		AvgSim	
	$\rho \times 100$	$\tau \times 100$	$\rho \times 100$	$\tau \times 100$
Ours	**55.4**	**40.9**	49.3	35.2
SingleEmb	42.8	30.6	42.8	30.6
Multi-prototype	40.7	29.1	38.3	27.4

Word Similarity Evaluation

- Qualitative Evaluation: K-nearest neighbors

Word	Nearest Neighboors
制服$_{uniform}$	穿着$_{dress}$, 警服$_{policeman\ uniform}$
制服$_{subdue}$	打败$_{defeat}$, 击败$_{beat}$, 征服$_{conquer}$
花$_{spend}$	花费$_{cost}$, 节省$_{save}$, 剩下$_{rest}$
花$_{flower}$	菜$_{greens}$, 叶$_{leaf}$, 果实$_{fruit}$
法$_{law}$	法令$_{ordinance}$, 法案$_{bill}$, 法规$_{rule}$
法$_{method}$	概念$_{concept}$, 方案$_{scheme}$
法$_{French}$	德$_{Germany}$, 俄$_{Russia}$, 英$_{Britain}$
领导$_{lead}$	监督$_{supervise}$, 决策$_{decision}$
领导$_{leader}$	主管$_{chief}$, 上司$_{boss}$

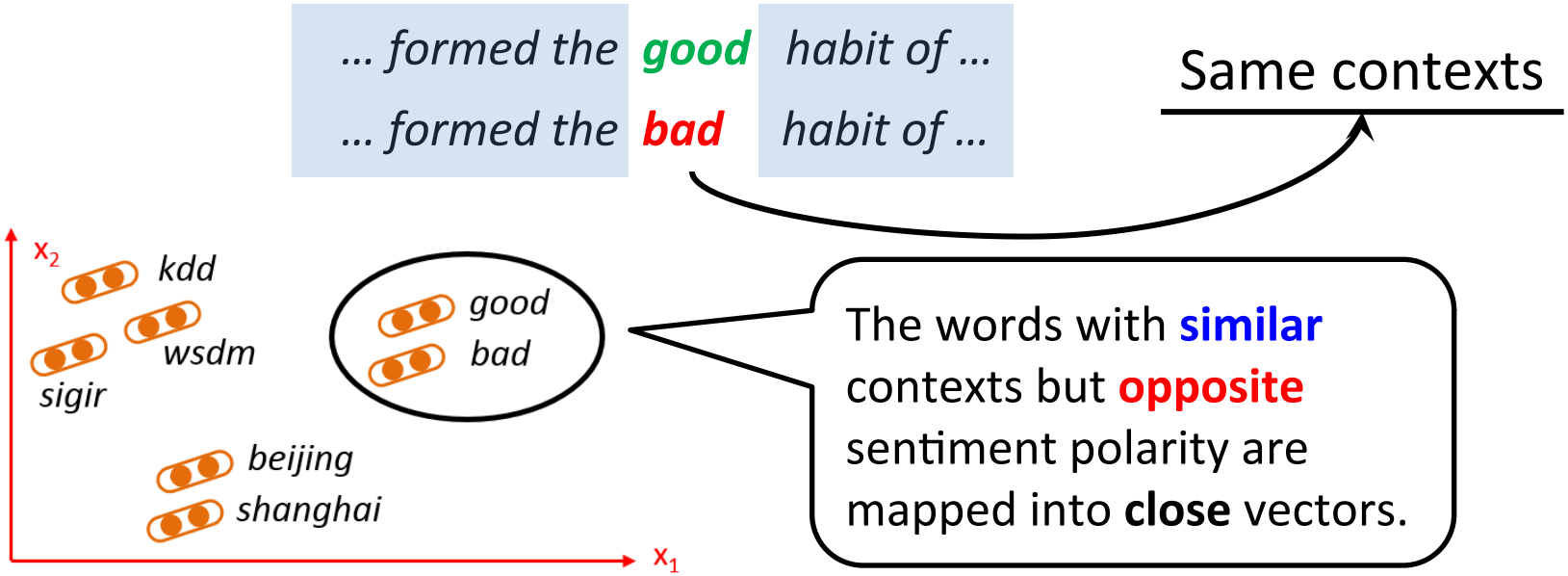
Deep Learning for NLP



- Learning Sense-specific Word Embedding by Exploiting Bilingual Resources, Coling 2014
- **Sentiment-Specific Word Embedding, ACL 2014, Coling 2014**

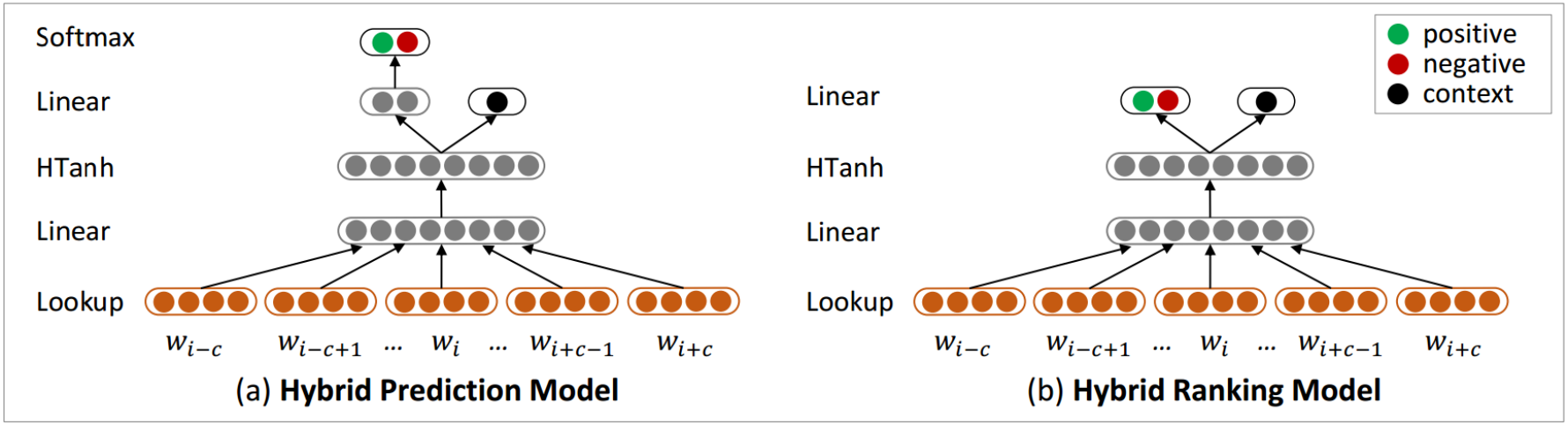
Why Sentiment Embedding

- Existing **context-based** embedding learning algorithms are not effective enough for sentiment analysis
- For example



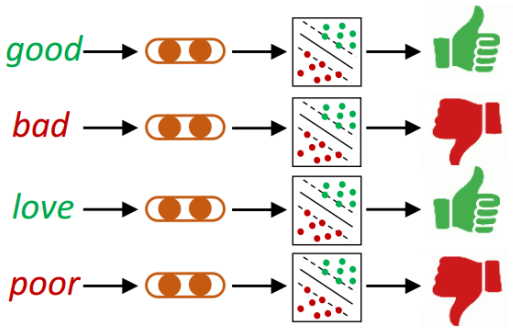
An illustration about models

- Basic idea
  - Consider the **contexts** of words and **sentiment** of sentences
  - We introduce two hybrid models



Application 1: Word Level Sentiment Classification

- Task description
  - Given a word, we classify its sentiment polarity as positive or negative
  - We use word embedding as word feature



Sentiment Lexicon	Size	#Positive	#Negative
BL-Lexicon	6,786	2,006	4,780
MPQA	6,451	2,301	4,150
NRC-Lexicon	5,555	2,231	3,324

- Experimental Setting
  - Supervised learning with cross validation, accuracy.

Application 1: Word Level Sentiment Classification

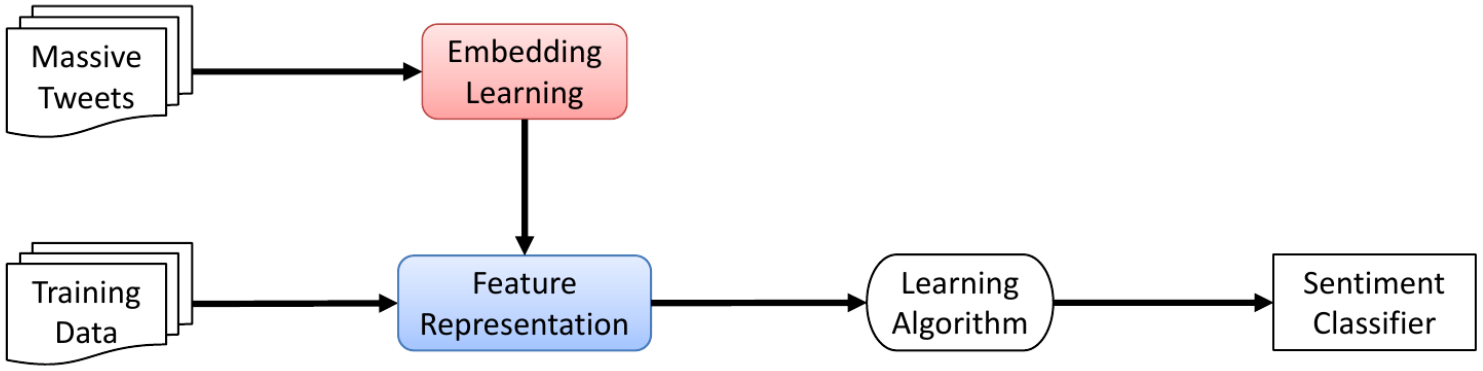
- Experimental Results

Embedding	N-Fold = 5		
	BL	MPQA	NRC
C&W	76.2	70.5	68.1
SE-SPred	80.8	78.2	73.3
SE-SRank	78.6	76.2	71.3
SE-HyPred	**86.0**	**85.3**	**77.5**
SE-HyRank	83.9	80.1	75.7

Hybrid models that capture both context-level and sentiment-level information perform best

Application 2: Twitter Sentiment Classification

- Task description
  - Given a tweet, we classify its polarity as Positive, Negative OR Neutral
- Apply sentiment embedding as features
  - Advantage: learning features automatically without feature engineering



Application 2: Twitter Sentiment Classification

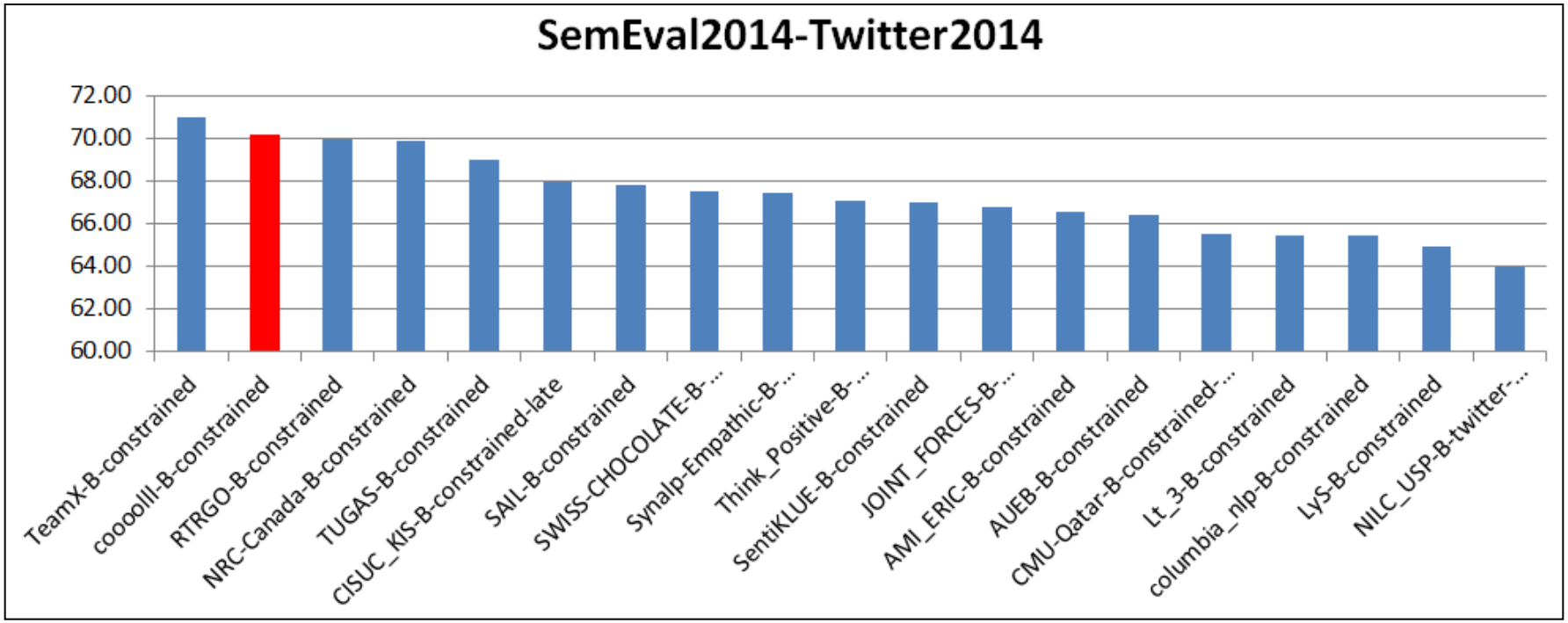
- Experimental Settings
  - We train supervised classifier on SemEval dataset.
  - We use min/max/avg pooling (simple and effective for tweets) as composition function.

- Sentiment embedding features perform comparable with state-of-the-art hand-crafted features.

SVM + unigram	74.50
SVM + uni/bi/tri-gram	75.06
NBSVM	75.28
RAE	75.12
NRC (**Top System** in SemEval)	**84.73**
SE-HyRank	**84.98**
SE-HyRank+NRC	**86.58**

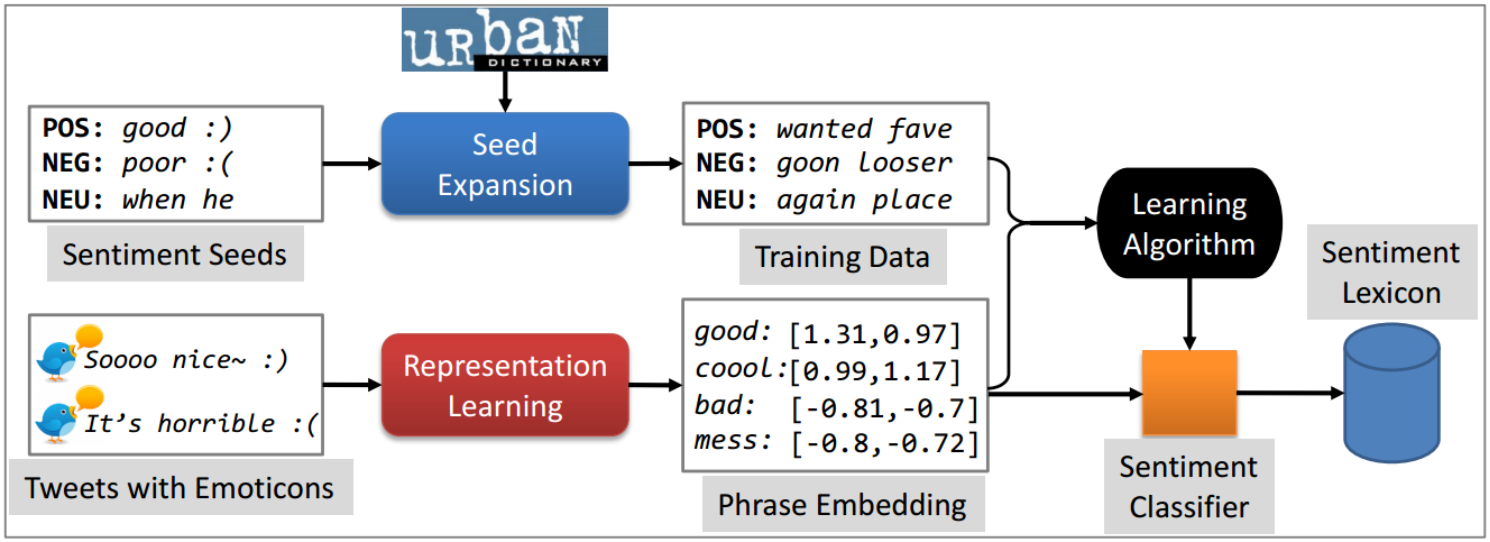
Application 2: Twitter Sentiment Classification

- Our system yields the **second** place among 45 systems. (Red bar is our system.)



Application 3: Building Sentiment Lexicon

- We regard building sentiment lexicon as a phrase-level sentiment classification task
  - Train a phrase-level sentiment classifier with supervised learning, regarding sentiment embedding as features.



Application 3: Building Sentiment Lexicon

- Experimental setting
  - Apply sentiment lexicon as features in existing supervised pipeline for Twitter sentiment

Although the generated sentiment lexicon is not the largest one in literature, it outperforms previous ones when used for Twitter sentiment classification.

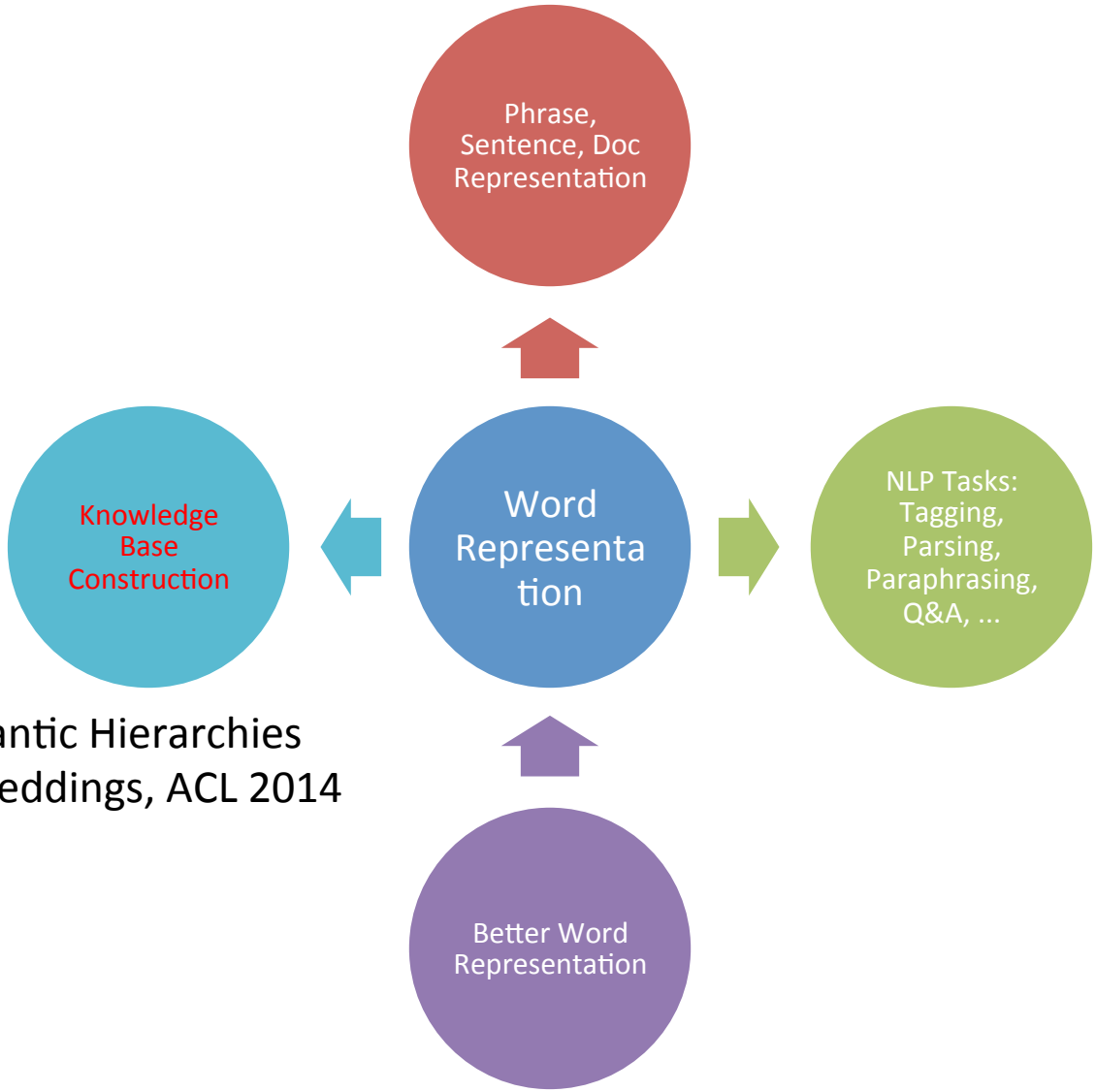
Lexicon	Positive	Negative	Total
HL	2,006	4,780	6,786
MPQA	2,301	4,150	6,451
NRC-Emotion	2,231	3,324	5,555
TS-Lex	**178,781**	**168,845**	**347,626**
HashtagLex	216,791	153,869	370,660
Sentiment140Lex	480,008	260,158	740,166

Lexicon	Unique	Appended
HL	60.49	79.40
MPQA	59.15	76.54
NRC-Emotion	54.81	76.79
HashtagLex	65.30	76.67
Sentiment140Lex	72.51	80.68
TS-Lex	**78.07**	**82.36**

Conclusion

- We learn sentiment embedding to capture both contexts of words and sentiment of sentences.
- We apply sentiment embedding to three applications in sentiment analysis, and show its effectiveness.

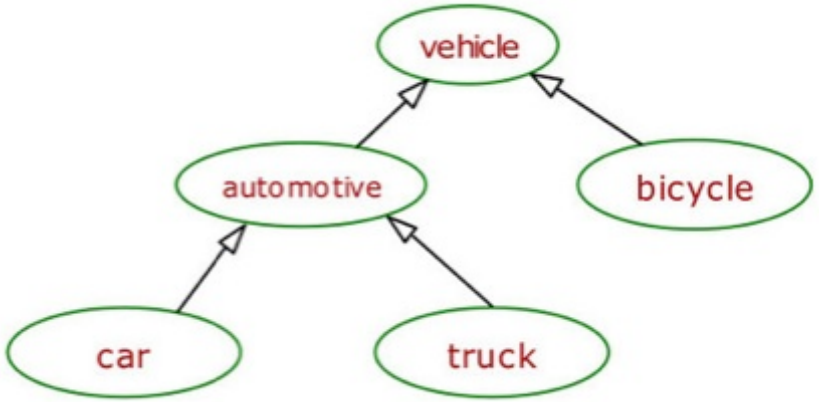
Deep Learning for NLP



Learning Semantic Hierarchies via Word Embeddings, ACL 2014

Semantic Hierarchies

- Learning Semantic Hierarchies via Word Embeddings
  - car → automotive
    - hypernym: automotive
    - hyponym: car



  - manually-built semantic hierarchies
    - WordNet
    - HowNet
    - CilinE (Tongyi Cilin - Extended version)

Previous Work

- Pattern-based method
  - e.g. “such NP1 as NP2”
    - Hearst (1992) ; Snow et al. (2005)

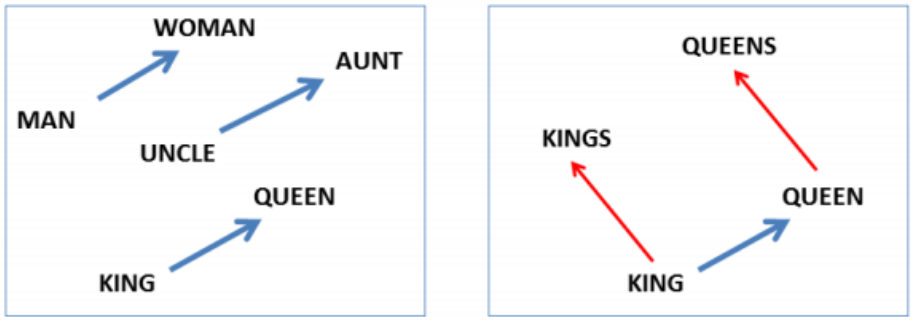
Pattern	Translation
w 是[一个|一种] h	w is a [a kind of] h
w [、] 等 h	w[,] and other h
h [，] 叫[做] w	h[,] called w
h [，] [像]如 w	h[,] such as w
h [，] 特别是 w	h[,] especially w

- Methods based on web mining
  - assuming that the hypernyms of an entity co-occur with it frequently
  - extracting hypernym candidates from multiple sources and learning to rank
    - Fu et al. (2013)

Word Embeddings

- Learning Semantic Hierarchies via Word Embeddings

$$v(\texttt{king}) - v(\texttt{queen}) \approx v(\texttt{man}) - v(\texttt{woman})$$



Mikolov et al. (2013a)

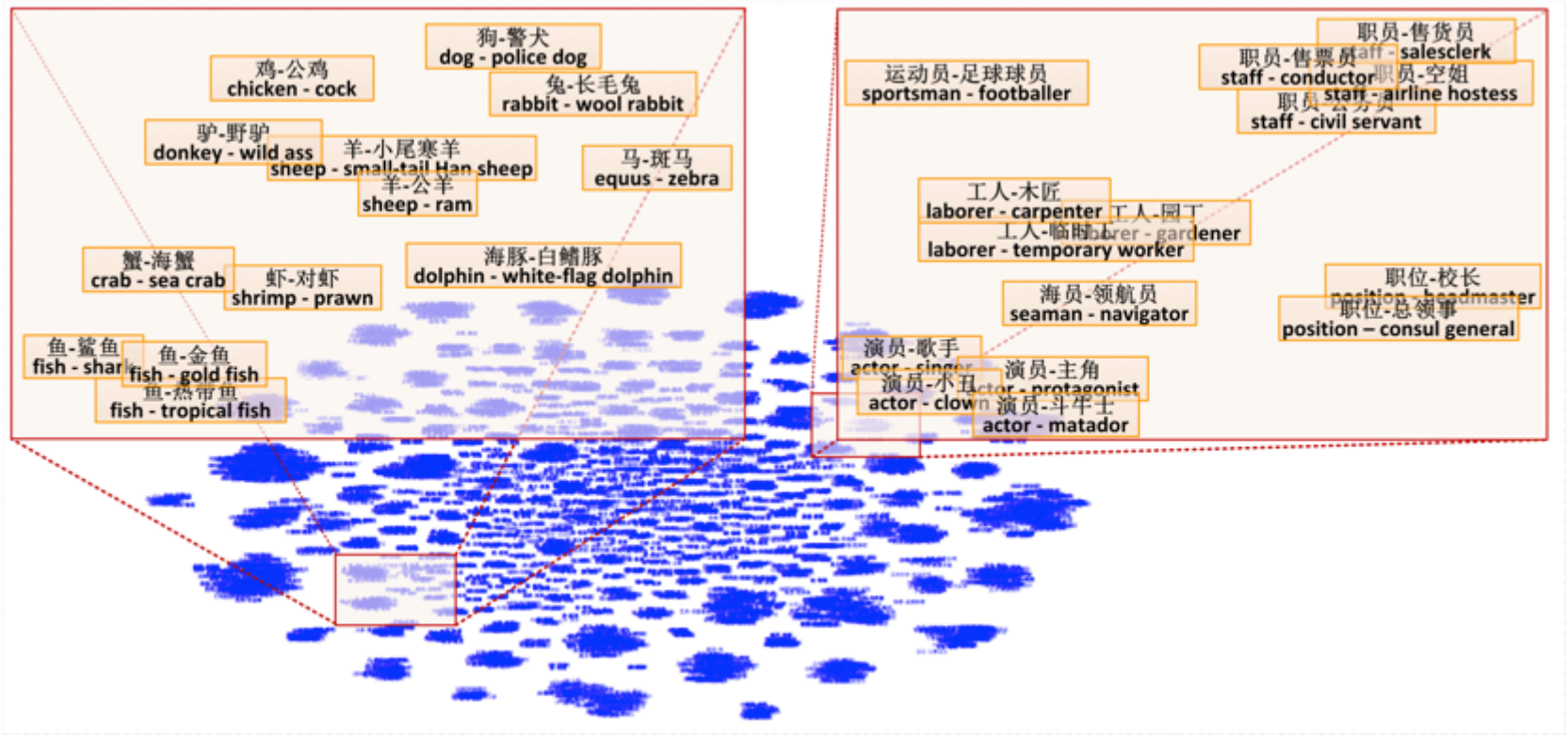
Motivation

- Does the embedding offset work well in hypernym–hyponym relations ?

No.	Examples
1	$v(\text{虾}) - v(\text{对虾}) \approx v(\text{鱼}) - v(\text{金鱼})$ $v(\text{shrimp}) - v(\text{prawn}) \approx v(\text{fish}) - v(\text{gold fish})$
2	$v(\text{工人}) - v(\text{木匠}) \approx v(\text{演员}) - v(\text{小丑})$ $v(\text{laborer}) - v(\text{carpenter}) \approx v(\text{actor}) - v(\text{clown})$
3	$v(\text{工人}) - v(\text{木匠}) \not\approx v(\text{鱼}) - v(\text{金鱼})$ $v(\text{laborer}) - v(\text{carpenter}) \not\approx v(\text{fish}) - v(\text{gold fish})$

Motivation

- Clusters of the vector offsets in training data



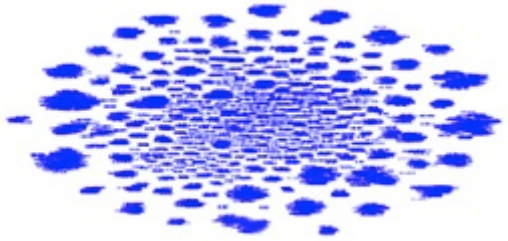
Projection Learning

- A uniform Linear Projection
  - Given a word ***x*** and its hypernym ***y***, there exists a matrix ***Φ*** so that ***y*** = ***Φx***.

$$\Phi^* = \arg\min_{\Phi} \frac{1}{N} \sum_{(x,y)} \| \Phi x - y \|^2$$

Projection Learning

- Piecewise Linear Projections
  - clustering y - x

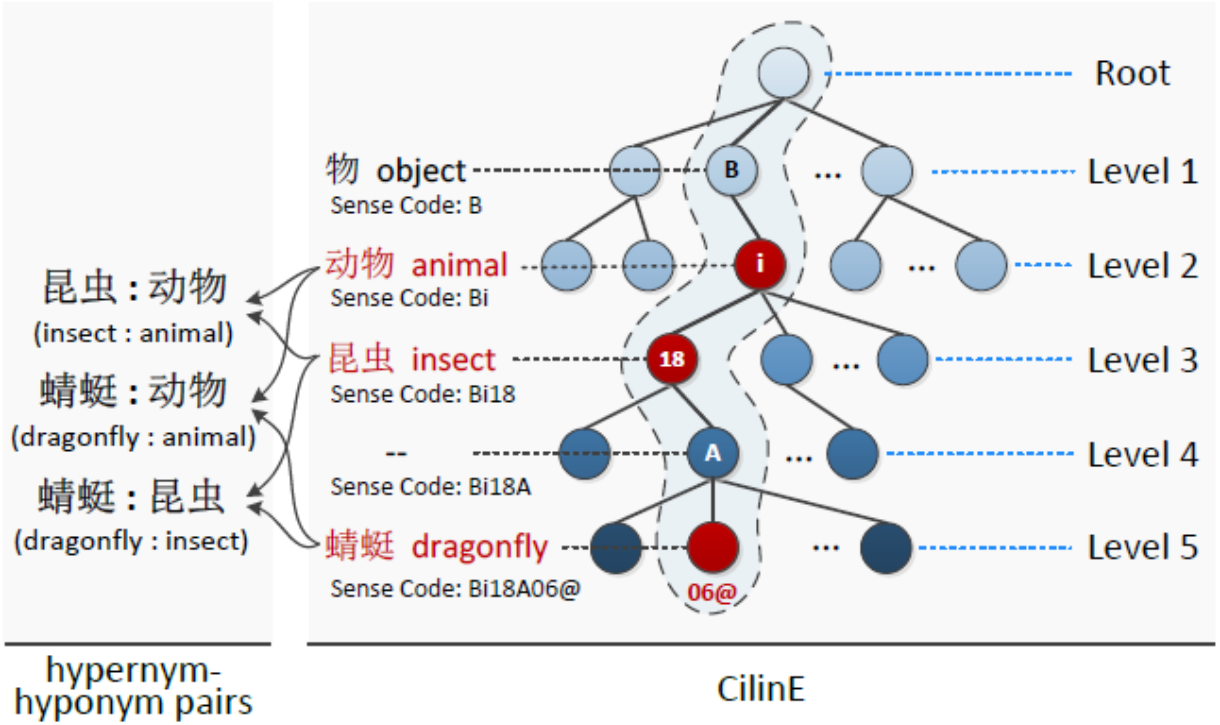


  - learning a separate projection for each cluster

$$\Phi_k^* = \arg\min_{\Phi_k} \frac{1}{N_k} \sum_{(x,y)\in C_k} \| \Phi_k x - y \|^2$$

Projection Learning

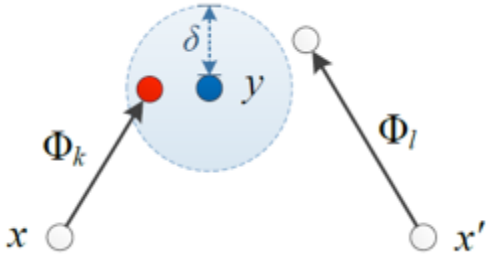
- Training data



is-a Relation Identification

- Given two words x and y
  - If y is determined as a hypernym of x, either of the two conditions must be satisfied.

Condition 1:



$$d(\Phi_k x, y) = \| \Phi_k x - y \|^2 < \delta$$

Condition 2:

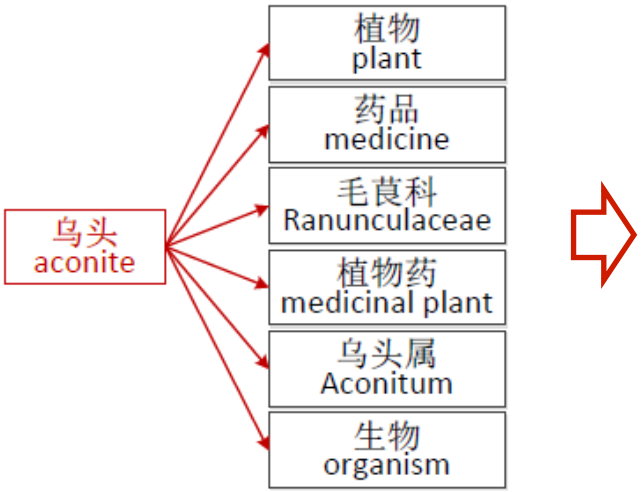
$$x \xrightarrow{H} z \text{ and } z \xrightarrow{H} y$$

Experimental Data

- Word embedding training
  - Corpus from Baidubaike
    - ~30 million sentences (~780 million words)
- Projection learning
  - CilinE
    - 15,247 is-a pairs

Experimental Data

- For evaluation



Relation	# of word pairs	
	Dev.	Test
hypernym–hyponym	312	1,079
hyponym–hypernym*	312	1,079
unrelated	1,044	3,250
Total	1,668	5,408

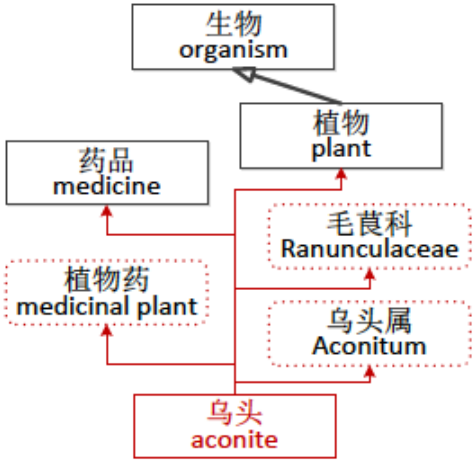
Fu et al. (2013)

Results and Analysis

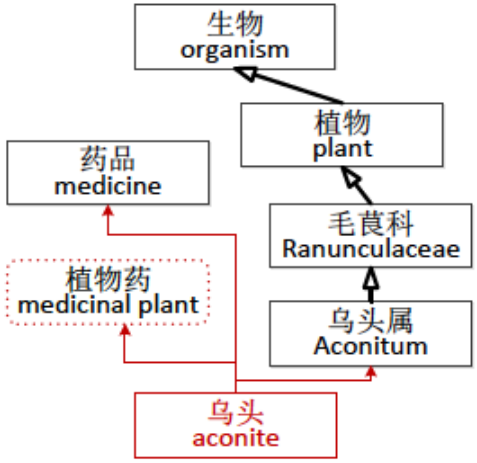
- Comparison with existing methods

	P(%)	R(%)	F(%)	
M_{CilinE}	98.21	50.88	67.03	
$M_{Wiki+CilinE}$	92.41	60.61	73.20	Suchanek et al. (2008)
$M_{Pattern}$	97.47	21.41	35.11	Hearst (1992)
M_{Snow}	60.88	25.67	36.11	Snow et al. (2005)
$M_{balApinc}$	54.96	53.38	54.16	Kotlerman et al. (2010)
M_{invCL}	49.63	62.84	55.46	Lenci and Benotto (2012)
M_{Fu}	71.64	52.92	60.87	Fu et al. (2013)
M_{offset}	59.26	63.19	61.16	
M_{Emb}	80.54	67.99	**73.74**	
$M_{Emb+CilinE}$	80.59	72.42	76.29	
$M_{Emb+Wiki+CilinE}$	79.78	80.81	**80.29**	

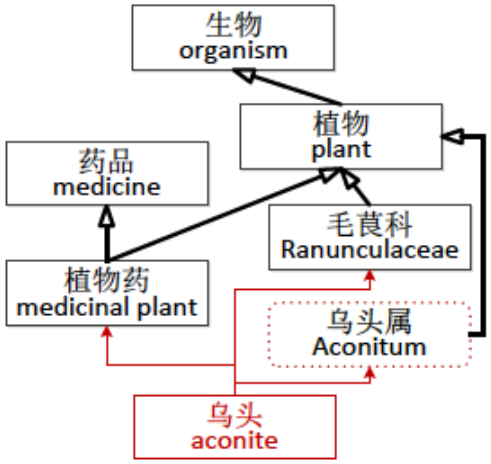
Results and Analysis



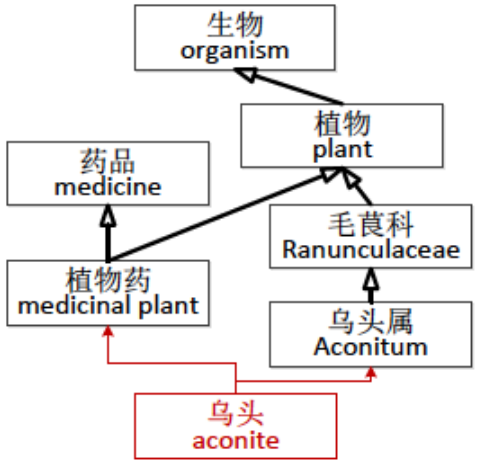
(a) CilinE



(b) Wikipedia+CilinE



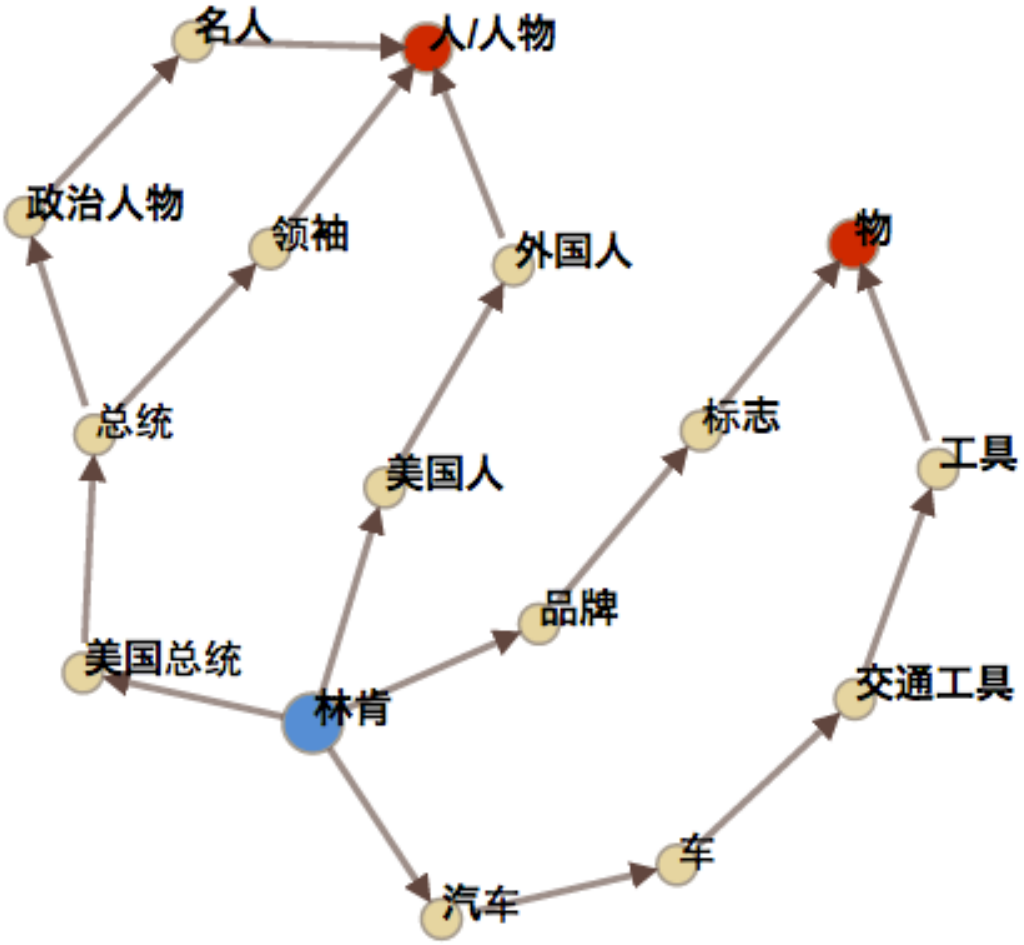
(c) Embedding



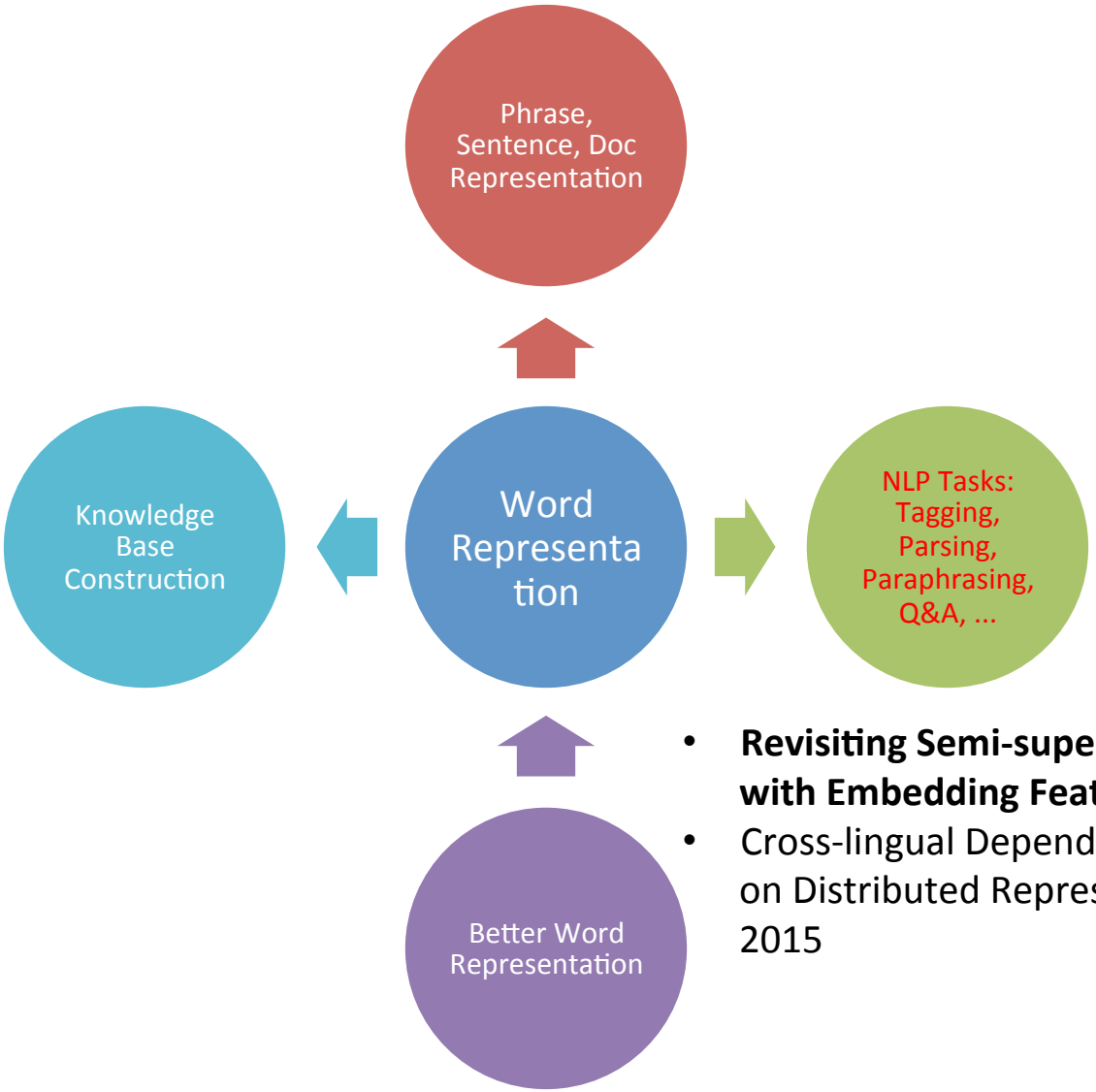
(d) Embedding+Wikipedia+CilinE

Demo

- http://www.bigcilin.com



Deep Learning for NLP



- **Revisiting Semi-supervised Learning with Embedding Features, EMNLP 2014**
- Cross-lingual Dependency Parsing Base on Distributed Representations, ACL 2015

Motivation

- Are the continuous embedding features fit for the generalized linear models (e.g., CRF) which are most widely adopted in NLP?

- How can the generalized linear models better utilize the embedding features?

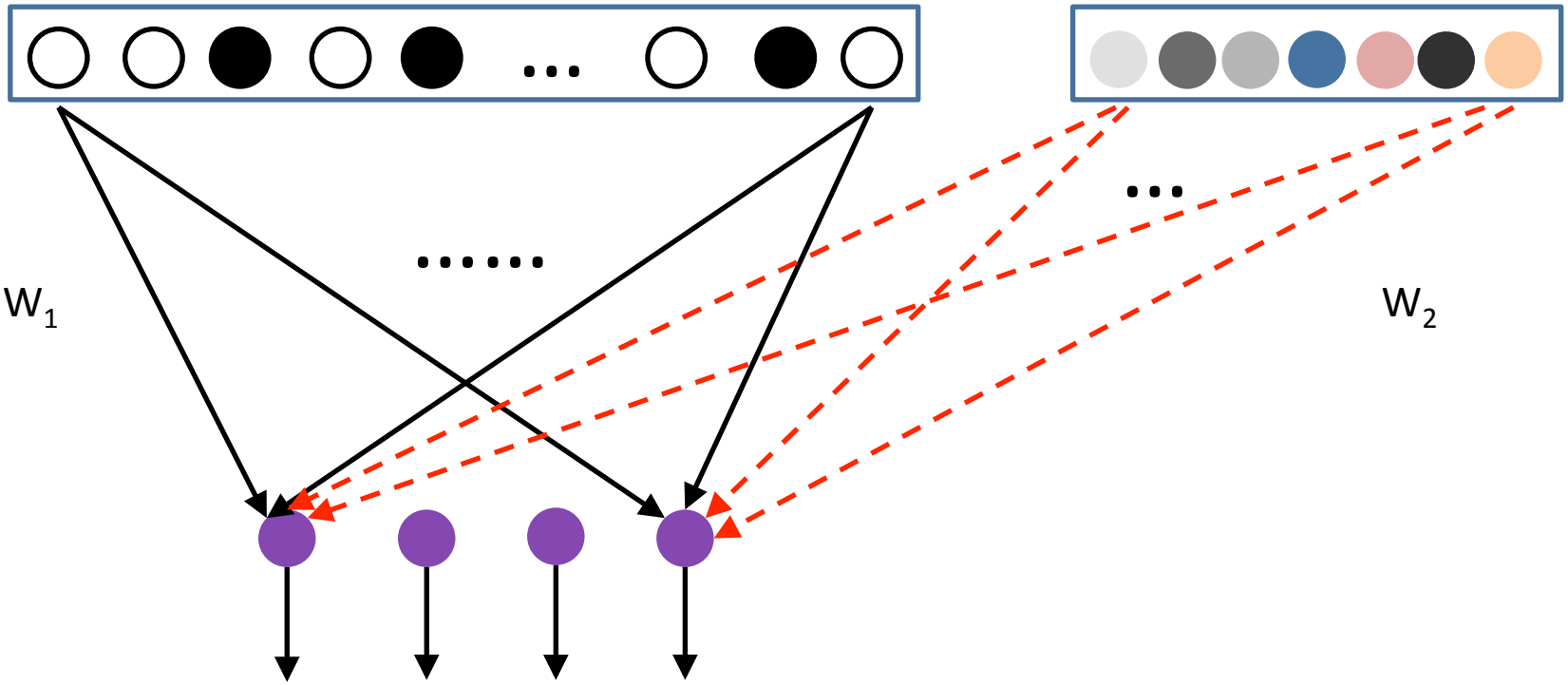
Main related studies

- Turian et al. 2010
  - The embedding features brought significant less improvement than Brown clusters
- Wang et al. 2012
  - Non-linear models benefit more with low-dimensional continuous feature.
  - Linear models are more effective in high-dimensional discrete space.
- Yu et al. 2013
  - Introduce the compound cluster feature.

Word Embeddings Applied to Generalized Linear Models

Sparse, High-dimensional lexicalized features

Dense, Continuous Word Embedding features



Our work

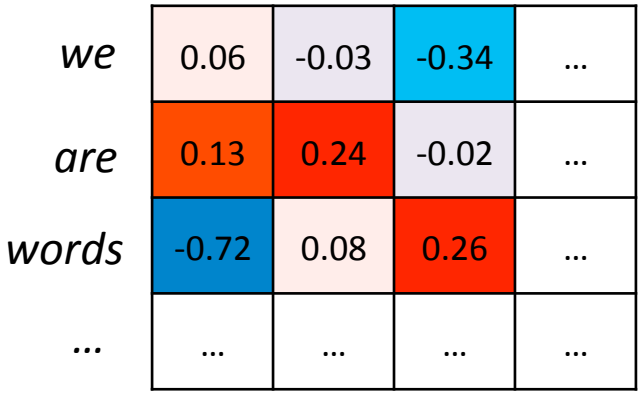
We investigate and carefully compare three approaches for utilizing embedding features, empirically (rather than theoretically)

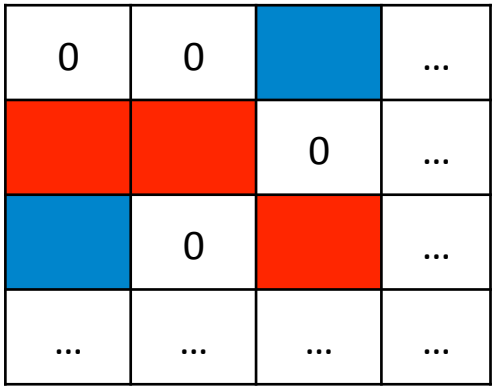
1. Binarization of embeddings
2. Clustering of embeddings
3. Distributional prototype features

1. Binarization of Embeddings

Dimension of word embeddings (e.g. 200)

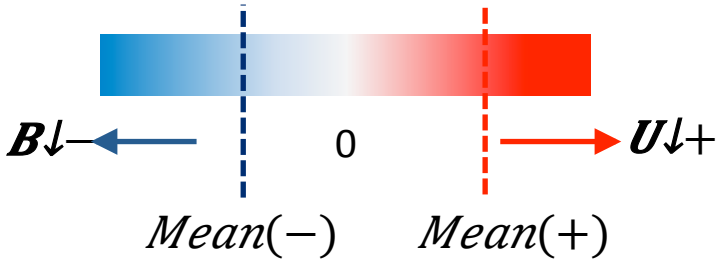
Vocabulary





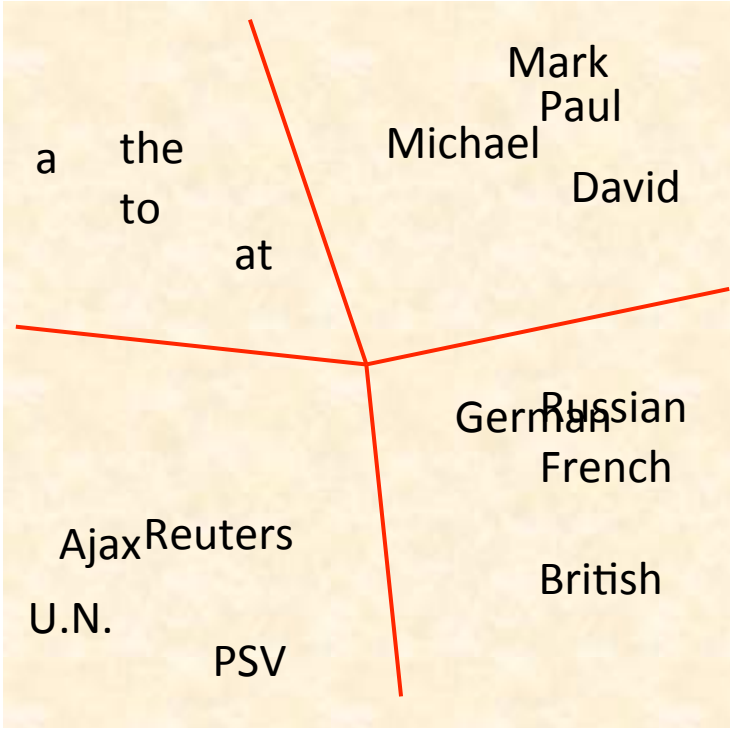
Conversion function:

$$M_{ij} = \phi(C_{ij}) = \begin{cases} U_+, & if\ C_{ij} \geq mean(C_{i+}) \\ B_-, & if\ C_{ij} \leq mean(C_{i-}) \\ 0, & otherwise \end{cases}$$



2. Clustering of Embeddings

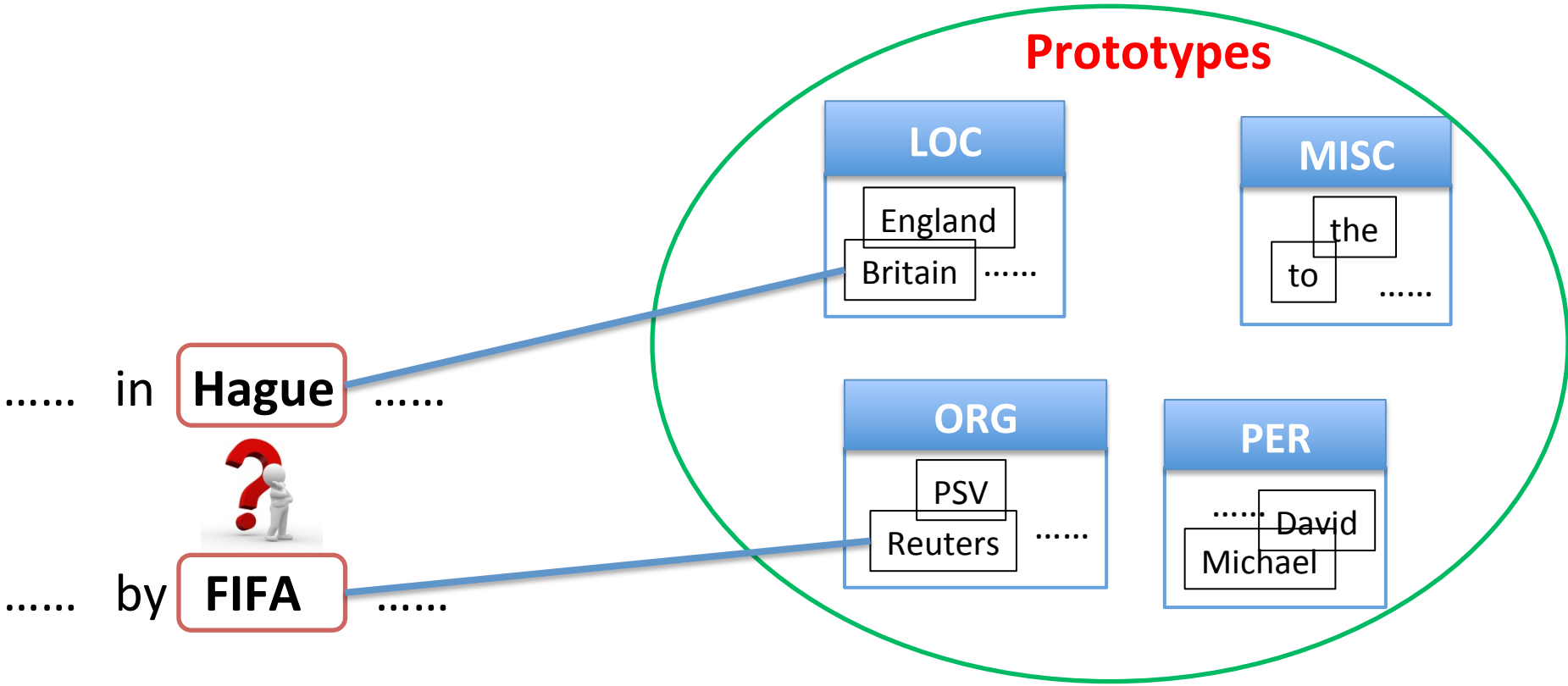
- Means clustering based on word embeddings
  - Cluster ids of words are applied as features
- Different number of clusters indicates different granularities
  - Combine cluster features of different s



3. Distributional Prototype Features

Motivation:

- prototype-driven learning (Haghighi and Klein, 2006 NAACL)
- take NER as a case study. Suppose we have some prototypical examples for each target label (prototypes)



3. Distributional Prototype Features

Prototypes Extraction

- Normalized pointwise mutual information (NPMI)

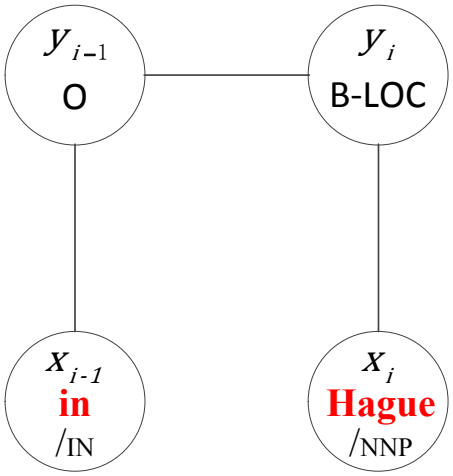
NE Type	Prototypes
B-PER	Mark, Michael, David, Paul
I-PER	Akram, Ahmed, Khan, Younis
B-ORG	Reuters, U.N., Ajax, PSV
I-ORG	Newsroom, Inc, Corp, Party
B-LOC	U.S., Germany, Britain, Australia
I-LOC	States, Republic, Africa, Lanka
B-MISC	Russian, German, French, British
I-MISC	Cup, Open, League, OPEN
O	., ,, the, to

3. Distributional Prototype Features

An example

– **NER** as sequence labeling

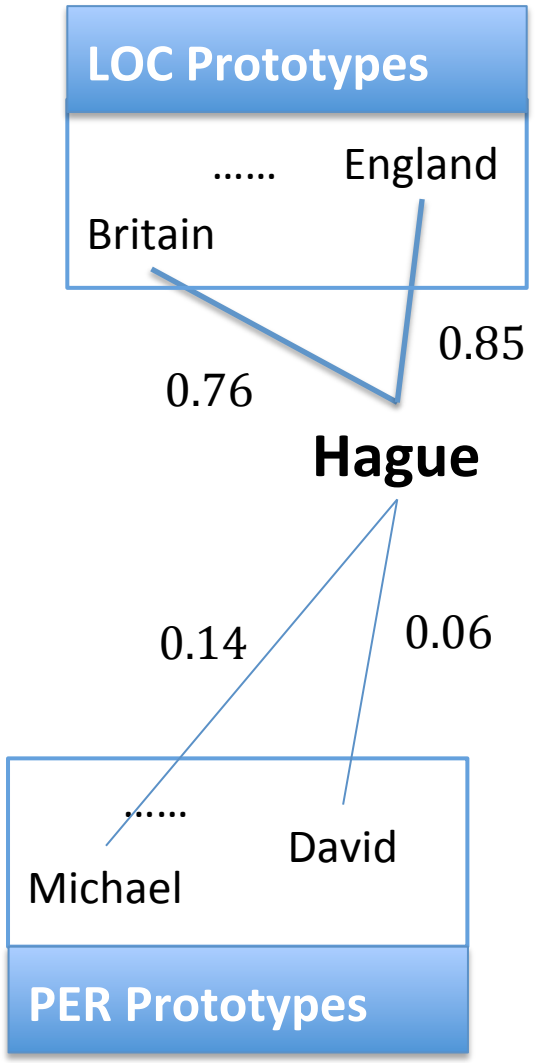
She	lives	in	**Hague**
PRP	*VBZ*	*IN*	*NNP*



$$f(y_{i-1}, y_i) = \text{O} \wedge \text{B-LOC}$$

$$f(x_i, y_i) =$$

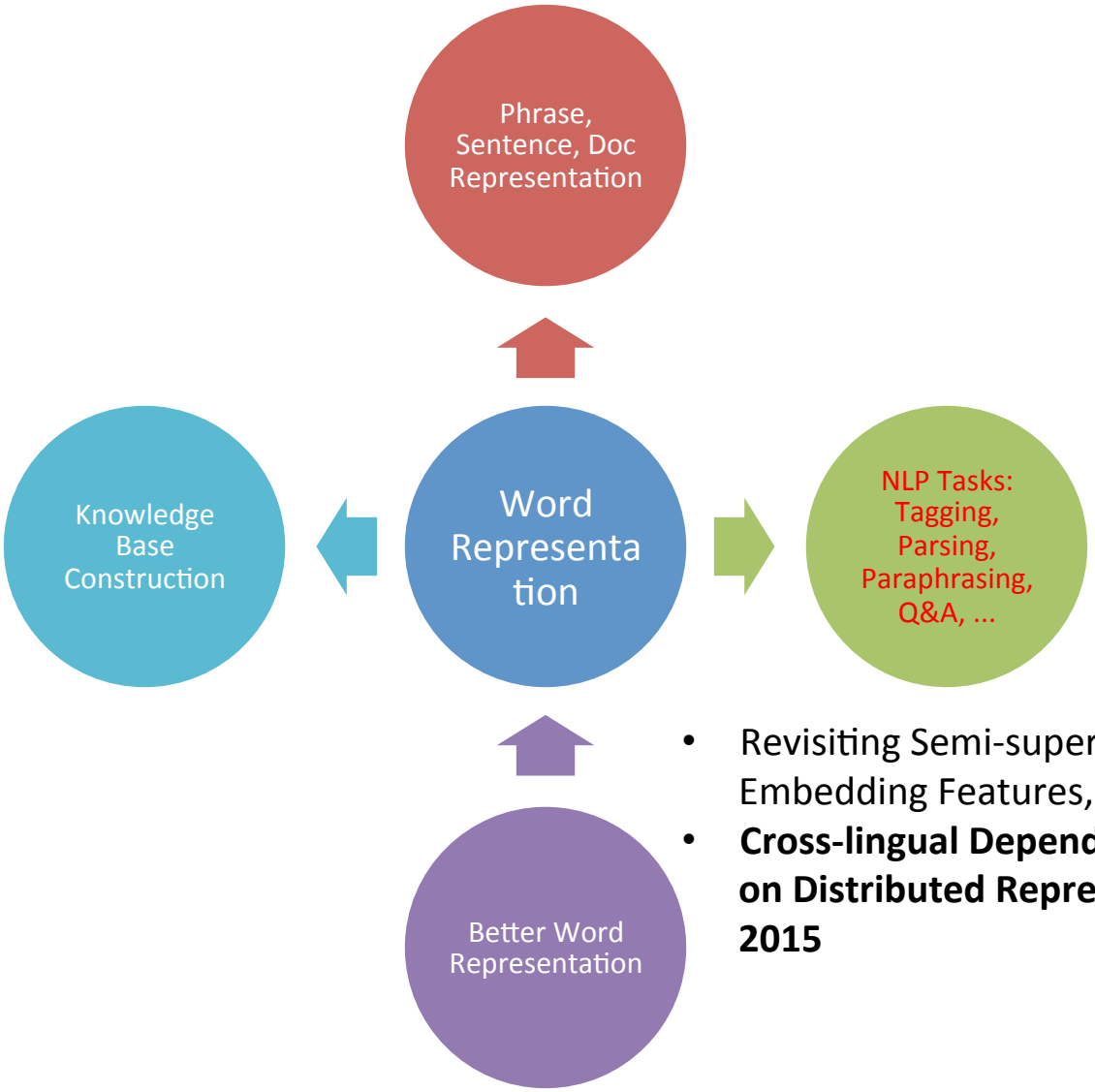
$$\left\{ \begin{array}{l} \text{word} = \text{Hague} \\ \text{pos} = \text{NNP} \\ \text{proto} = \text{Britain} \\ \text{proto} = \text{England} \\ \cdots \end{array} \right\} \wedge \text{B-LOC}$$



Results

Setting	**F1**
Baseline	83.43
+DenseEmb	86.21
+Brown	87.49
+BinarizedEmb	86.75
+ClusterEmb	86.90
+DistPrototype	**87.44**
+ClusterEmb+DistPrototype	**88.11**
+Brown+ClusterEmb+DistPrototype	**88.58**
Finkel et al. (2005)	86.86
Krishnan and Manning (2006)	87.24
Ando and Zhang (2005)	89.31
Collobert et al. (2011)	88.67

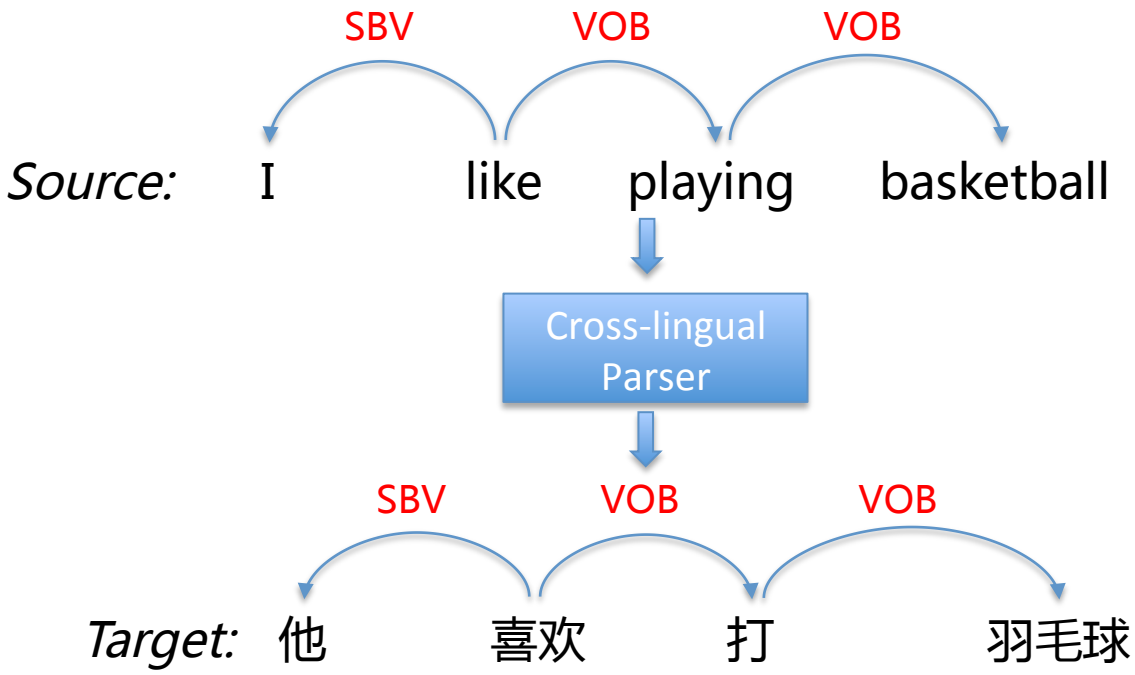
Deep Learning for NLP



- Revisiting Semi-supervised Learning with Embedding Features, EMNLP 2014
- **Cross-lingual Dependency Parsing Base on Distributed Representations, ACL 2015**

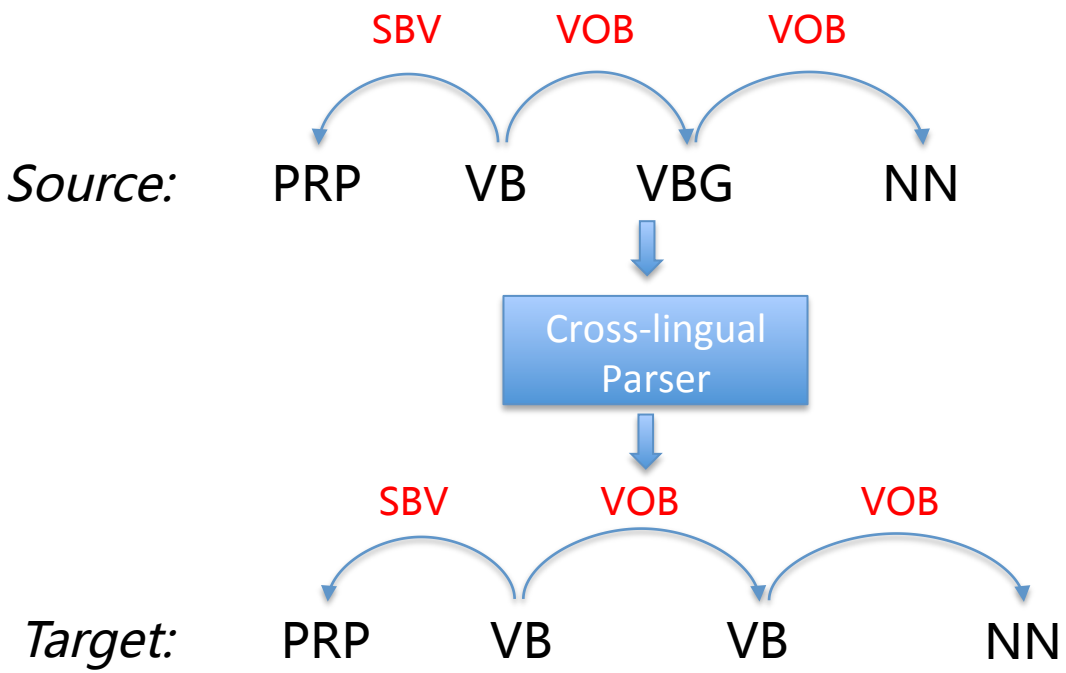
Cross-lingual Transfer DP

- No TreeBanks for low-resource (target) languages
- Transfer the parser of a rich-resource (source) language (e.g. English) to a low-resource language



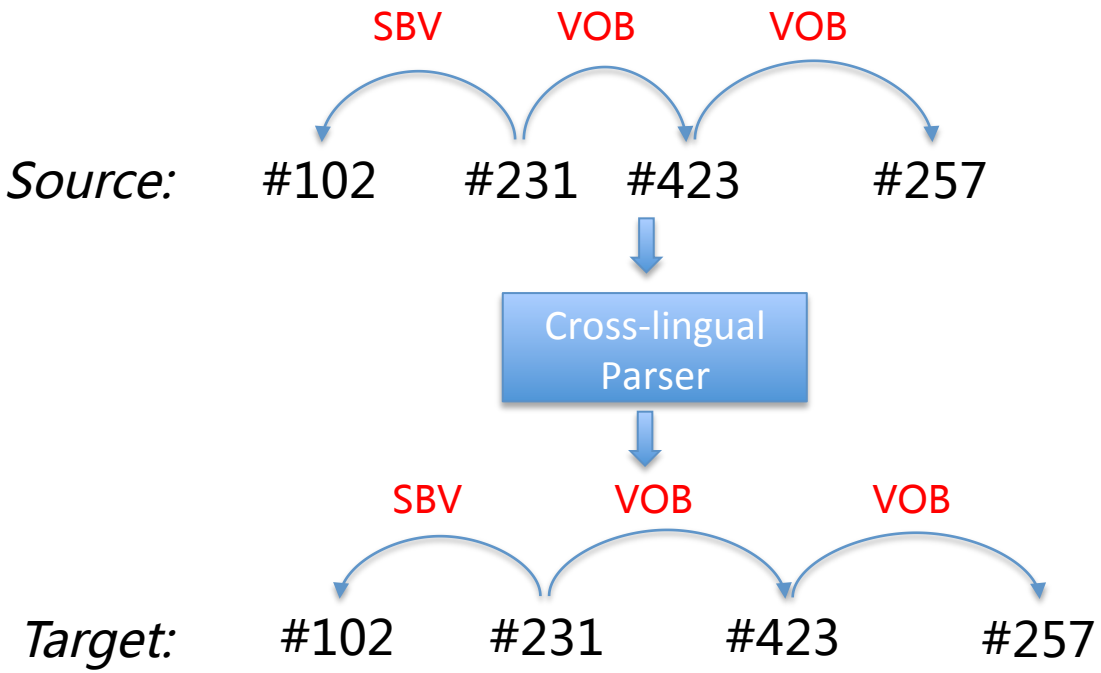
Previous Work

- Delexicalized Parser (McDonald et al. 2011)
  - Only use non-lexical features



Previous Work

- Cross-lingual Word Clustering (Tackstrom et al. 2012)
  - Coarse-grained word representation, which partially fills the *lexical feature gap*

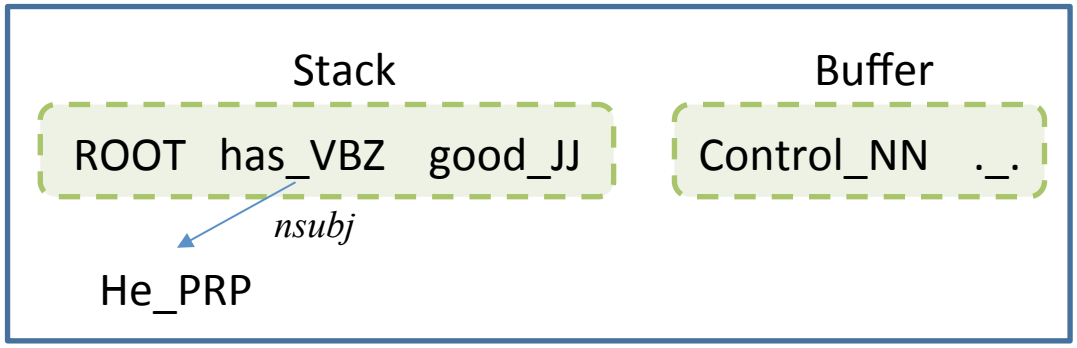


Our Motivation

- Bridging the *lexical feature gap* with distributed features representations (Embeddings)
  - Cross-lingual words
  - POS, Label histories
  - Word clusters

Transition-based Dependency Parsing

- Greedily predict a transition sequence from an initial parser state to some terminal states
- State (configuration)
  = Stack + Buffer + Dependency Arcs



Configuration

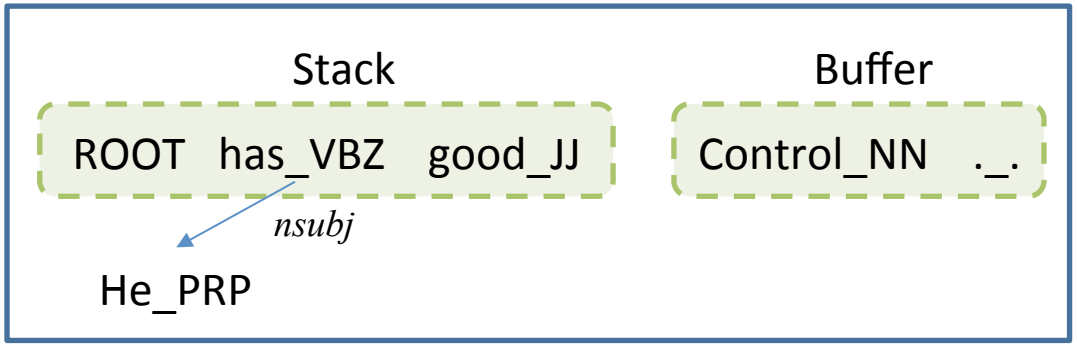
classifier

LEFT-ARC(l)
RIGHT-ARC(l)
SHIFT

ArcStandard algorithm

Traditional Features

Configuration



Feature Vector:

- Binary
- Sparse
- High-dimensional

Feature templates: a combination of $1 \sim 3$ elements from the configuration.

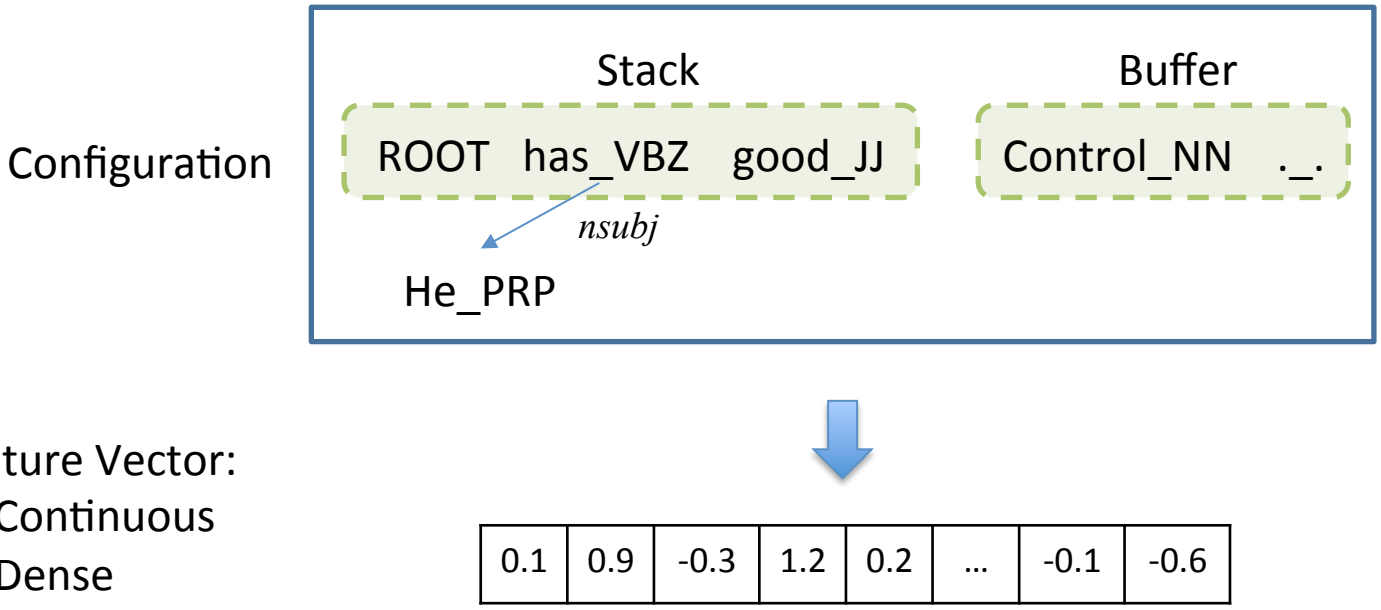
- For example: (Zhang and Nivre, 2011)

Problems of Traditional Features

- Sparse
  - Lexicalized features
  - High-order interaction features
- Incomplete
  - Unavoidable in hand-crafted feature templates
- Computationally expensive
  - More than 95% of parsing time is consumed by feature computation

Neural Network Classifier

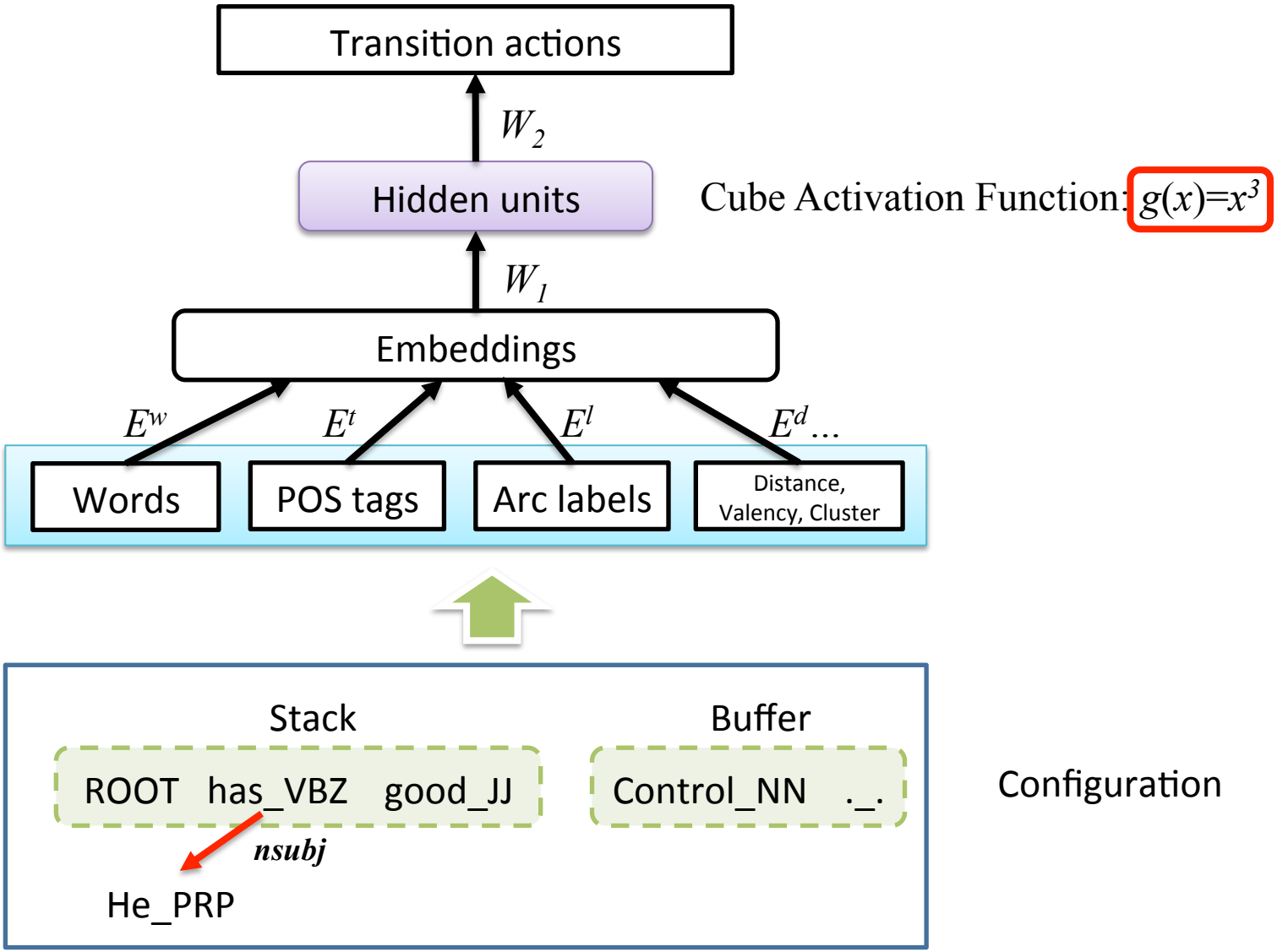
- Learn a **dense** and **compact** feature representation (Chen and Manning, 2014)



Feature Vector:

- Continuous
- Dense
- Low-dimensional

Model Architecture

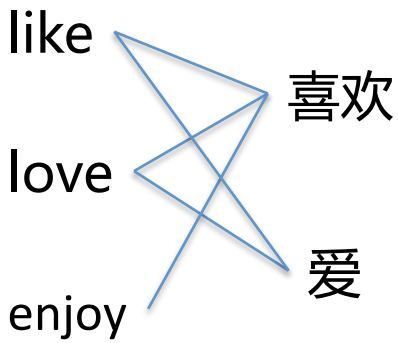


Indicator vs. Dense Features

- Problem #1: sparse
  - Distributed representations can capture similarities
- Problem #2: incomplete
  - Cube non-linearity can learn combinations automatically
- Problem #3: computationally expensive
  - String concatenation + look-up in a big table ➔ matrix operations

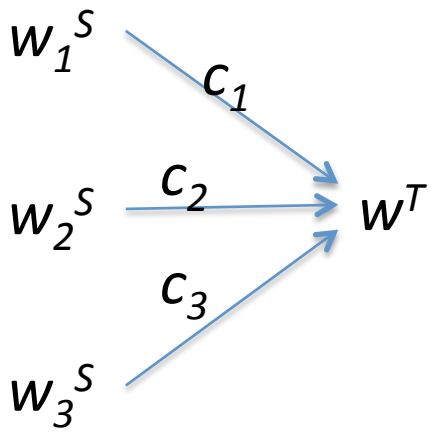
Cross-lingual Representation Transfer

- Non-lexical Features
  - POS, Label, Distance, …
  - One to one mapping: directly transfer
- Lexical Features
  - Word
  - Many to many mapping: ?



Robust Alignment-based Projection

- w_i^S aligns with w^T in c_i times



$$v(w^T) = \sum_i \frac{c_i}{|c|} v(w_i^S)$$

$$v(w^{OOV}) = \underset{w' \in C}{Avg}(v(w'))$$

$$C = \{w \mid EditDist(w^{OOV}, w) = 1\}$$

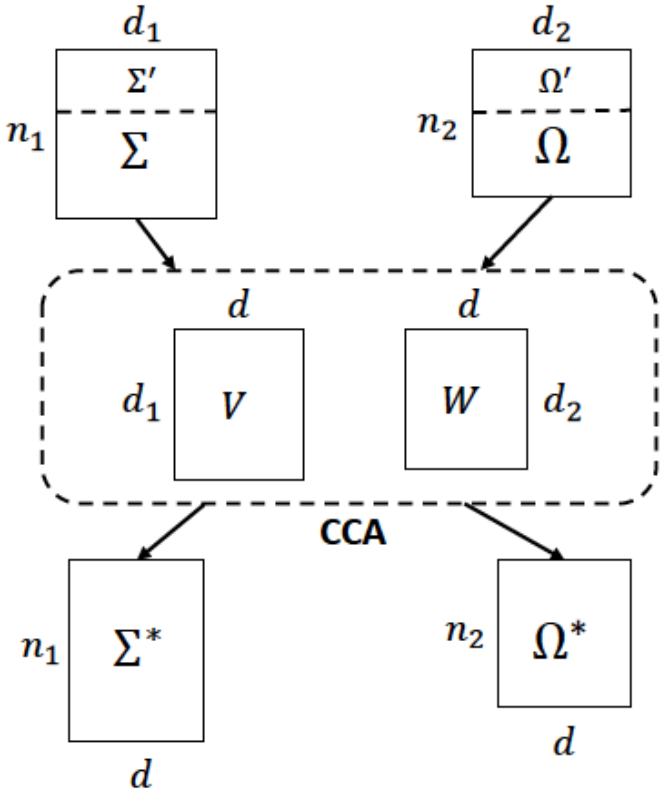
Canonical Correlation Analysis

- Canonical Correlation Analysis (CCA)
  - Measuring the linear relationship between multidimensional variables

$$V, W = CCA(\Sigma', \Omega')$$

$$\Sigma^* = \Sigma V, \quad \Omega^* = \Omega W$$

- Advantages
  - High word coverage
  - Encode the information of target language



Experiments

- Universal Dependency Treebanks v1 (Google)
  - Languages
    - Source: English (EN)
    - Target: German (DE), Spanish (ES), Franch (FR)
  - Universal Dependencies (42 relations)
  - Universal POS (12 tags)

Main Results

	Unlabeled Attachment Score (UAS)					Labeled Attachment Score (LAS)				
	EN	DE	ES	FR	AVG	EN	DE	ES	FR	AVG
Delexicalized	83.67	57.01	68.05	68.85	*64.64*	79.42	47.12	56.99	57.78	*53.96*
PROJ	91.96	60.07	71.42	71.36	*67.62*	90.48	49.94	61.76	61.55	*57.75*
PROJ+Cluster	92.33	60.35	**71.90**	**72.93**	***68.39***	90.91	**51.54**	**62.28**	**63.12**	***58.98***
CCA	90.62†	59.42	68.87	69.58	*65.96*	88.88†	49.32	59.65	59.50	*56.16*
CCA+Cluster	92.03†	**60.66**	71.33	70.87	*67.62*	90.49†	51.29	61.69	61.50	*58.16*
McD13	83.33	58.50	68.07	70.14	*65.57*	78.54	48.11	56.86	58.20	*54.39*
McD13*	84.44	57.30	68.15	69.91	*65.12*	80.30	47.34	57.12	58.80	*54.42*
McD13*+Cluster	90.21	60.55	70.43	72.01	*67.66*	88.28	50.20	60.96	61.96	*57.71*

Effect of Robust Projection

- Edit Distance for OOV words

		Simple	Robust	Δ
DE	coverage	91.37	94.70	+3.33
	UAS	59.74	60.35	+0.61
	LAS	50.84	51.54	+0.70
ES	coverage	94.51	96.67	+2.16
	UAS	70.97	71.90	+0.93
	LAS	61.34	62.28	+0.94
FR	coverage	90.83	97.60	+6.77
	UAS	71.17	72.93	+1.76
	LAS	61.72	63.12	+1.40

Effect of Fine-tuning Word Embedding

- **Projection** method over CCA lies in the fine-tuning of word embeddings while training the parser

		Fix	Fine-tune	Δ
DE	UAS	59.74	60.07	+0.33
	LAS	49.44	49.94	+0.50
ES	UAS	70.10	71.42	+1.32
	LAS	61.31	61.76	+0.45
FR	UAS	70.65	71.36	+0.71
	LAS	60.69	61.50	+0.81

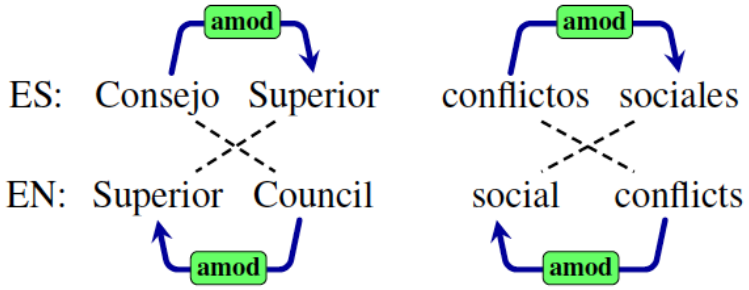
Target Minimal Supervision

- Cross-lingual approaches can only learn the **common** dependency structures shared between the source and target languages

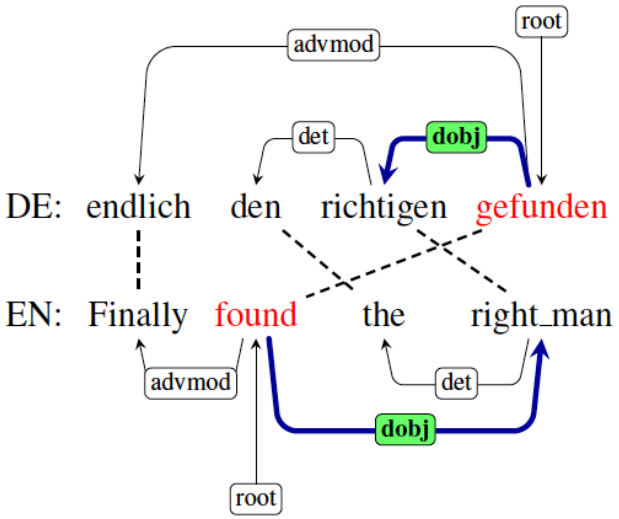
- For many languages, there are some **special** syntactic characteristics that are can only be learned from data in the target language

Target Minimal Supervision

- For example



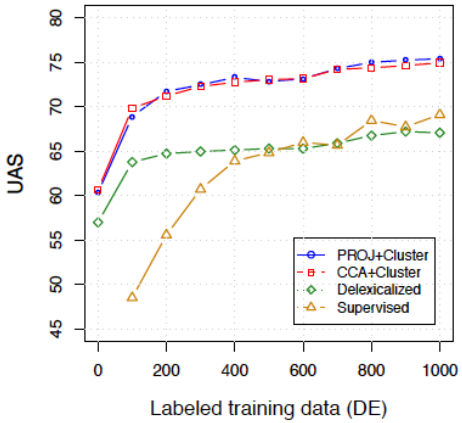
Relation: *amod*; Language: EN *vs.* ES, FR			
	$amod_{\searrow}$	$amod_{\nwarrow}$	ratio
EN	1,667	57,864	1 : 34.7
ES	14,876	5,205	2.9 : 1
FR	12,919	4,910	2.6 : 1

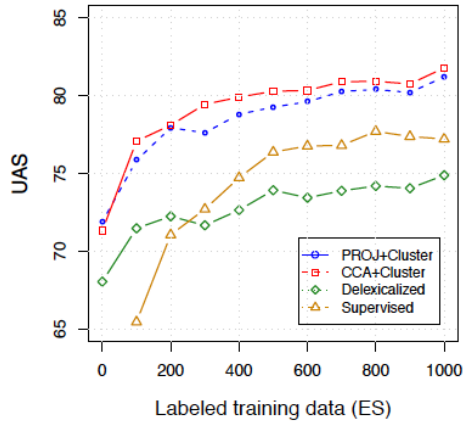


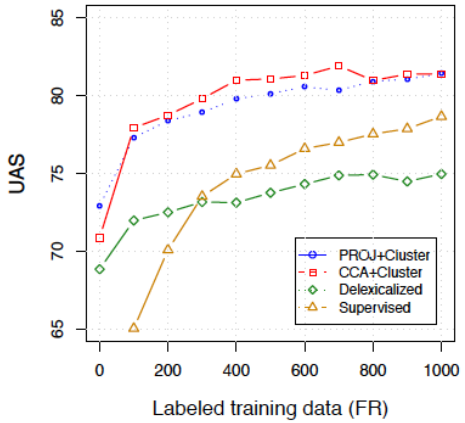
Relation: *dobj*; Language: EN *vs.* DE			
	$dobj_{\searrow}$	$dobj_{\nwarrow}$	ratio
EN	38,395	764	50.3 : 1
DE	4,277	3,457	1.2 : 1

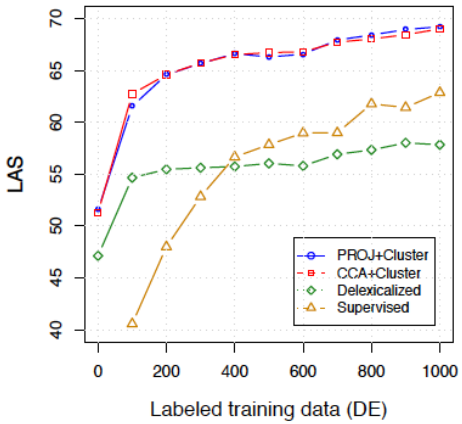
Results of Minimal Supervision

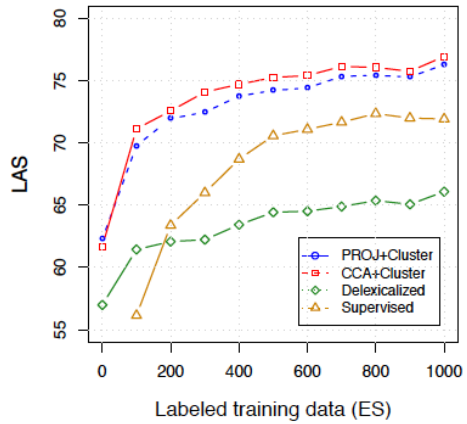
Solution: Use small labeled dependency trees (100 → 1000) from the target language to fine-tune the parsing model

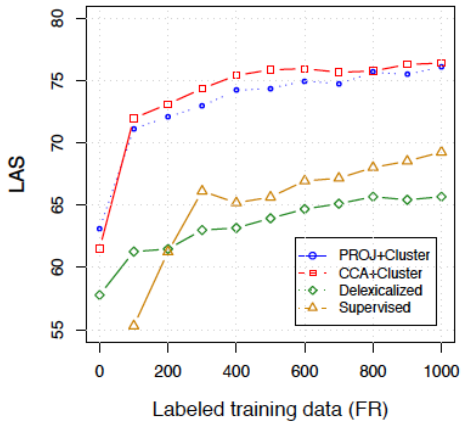




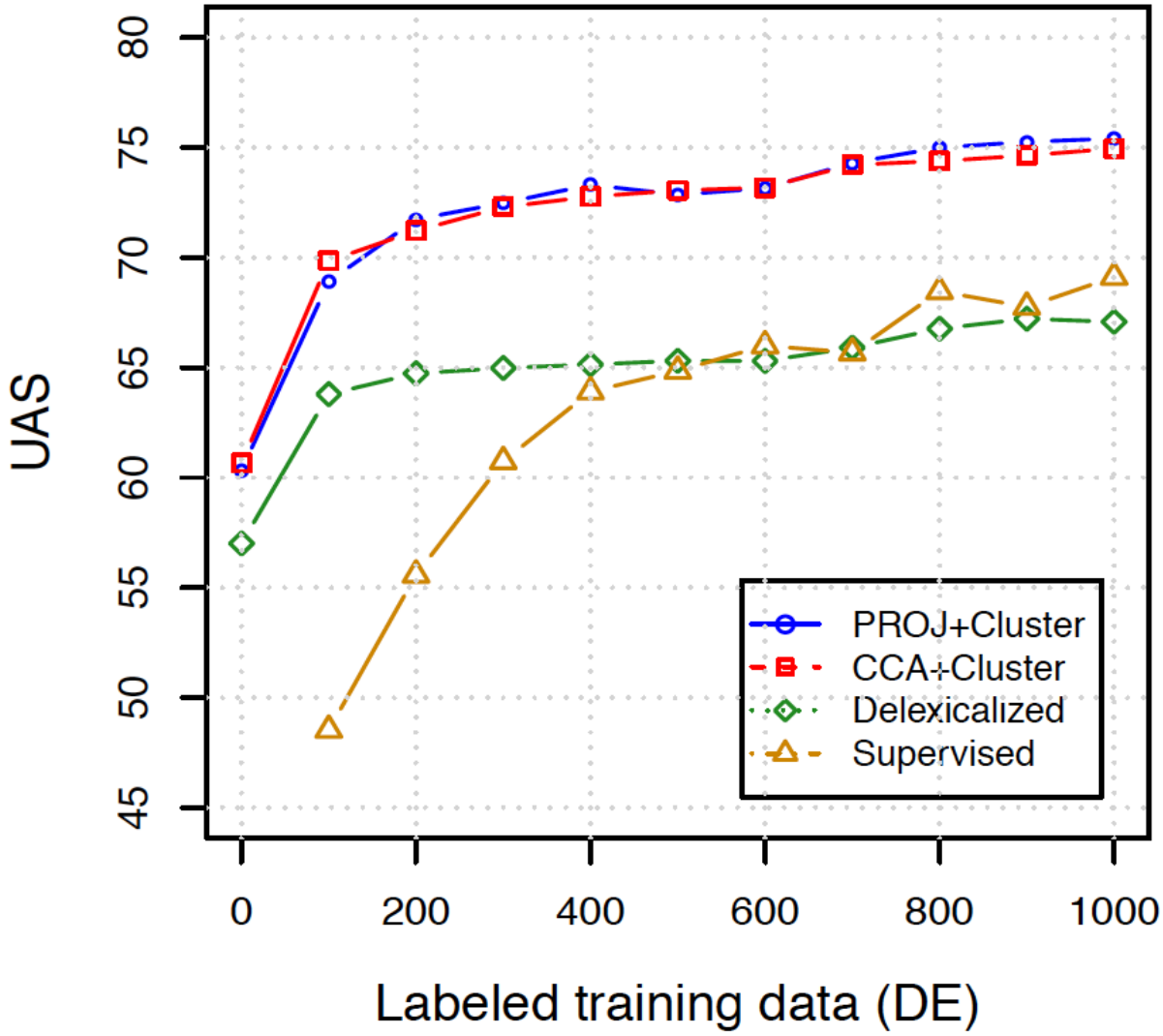








Results Zoom-in



Effect of Minimal Supervision (100 sent)

- Case studies
  - *dobj* (EN *vs.* DE)
  - *amod* (EN *vs.* ES, FR)

Relation: *dobj*; Language: DE		
	P	R
PROJ+Cluster	41.45	31.09
+100	41.90	51.40
Δ	↑ 0.45	↑ **20.31**
CCA+Cluster	39.47	31.74
+100	43.59	57.57
Δ	↑ 4.12	↑ **25.83**

Relation: *amod*; Language: ES, FR				
	ES		FR	
	P	R	P	R
PROJ+Cluster	94.97	80.05	92.94	81.70
+100	91.60	92.52	93.61	95.75
Δ	↓ 3.37	↑ **12.47**	↑ 0.67	↑ **14.05**
CCA+Cluster	93.37	77.31	92.08	72.22
+100	91.85	92.77	92.77	96.41
Δ	↓ 1.52	↑ **15.46**	↑ 0.69	↑ **24.19**

结论

- Deep Learning能缓解当前NLP面临的问题
  - 数据稀疏
    - Word Embedding
  - 需要细致的特征工程
    - Non-linear Hidden Layers
  - 多层处理带来的错误蔓延
    - End-to-end Learning
  - 处理速度较慢
    - Matrix Operations

Thanks Q&A

http://ir.hit.edu.cn