Relation Extraction (1)

Bonan Min

bonanmin@gmail.com

Some slides are based on class materials from Ralph Grishman, Thien Huu Nguyen

Relation Extraction

A *relation* is a predication about a pair of entities:

- *Rodrigo works for UNED.*
- *Alfonso lives in Tarragona.*
- *Otto's father is Ferdinand.*

Typically they represent information which is permanent or of extended duration.

Rudolph William Louis Giuliani *(/ˌdʒuːliˈɑːni/,* Italian: *[dʒuˈljaːni]; born May 28, 1944) is an American politician, attorney, and public speaker who served as the 107th Mayor of New York City from 1994 to 2001. He currently acts as an attorney to President Donald Trump.[1] Politically first a Democrat, then an Independent in the 1970s, and a Republican since the 1980s, Giuliani served as United States Associate Attorney General from 1981 to 1983. That year he became the United States Attorney for the Southern District of New York, holding the position until 1989.[2]*

Rudy Giuliani

107th Mayor of New York City	
In office	
January 1, 1994 – December 31, 2001	
Preceded by	David Dinkins
Succeeded by	Michael Bloomberg
United States Attorney for the Southern District of New York	
In office	
June 3, 1983 – January 1, 1989	
President	Ronald Reagan
Preceded by	John S. Martin Jr.
Succeeded by	Benito Romano (Acting)
United States Associate Attorney General	
In office	
February 20, 1981 – June 3, 1983	
President	Ronald Reagan
Preceded by	John Shenefield
Succeeded by	D. Lowell Jensen
Personal details	
Born	Rudolph William Louis Giuliani May 28, 1944 (age 75) New York City, New York, U.S.
Political party	Republican (1980–present)
Other political affiliations	Independent (1975–1980) Democratic (before 1975)

History of Relation Extraction

Relations were introduced in MUC-7 (1997)
- 3 relations

Extensively studied in ACE (2004 – 2008)
- Lots of training data

Effectively included in KBP
- Wikipedia infobox model
- QA-style evaluation

SemEval: relations between a pair of nouns

ACE (2004-2008)

Provided large exhaustively annotated corpora

- Pre-defined types between ACE entities
- A few hundred files were provided for training/development/testing
- Several revisions of relation definitions
  - With goal of having a set of relations which can be ore consistently annotated

Both entities must be mentioned in the same sentence

- Do not get a parent-child relation from
  - *Ferdinand and Isabella were married in 1481. A son was born in 1485.*
- Or an employee relation for
  - Bank Santander replaced several executives. Alfonso was named an executive vice president.

Base for extensive research

- On supervised and semi-supervised methods

Relation type	Subtypes
Physical	Located, Near, Part-whole
Personal-social	Business, Family, Other
Employment / Membership / Subsidiary	Employ-executive, Employ-staff, Employ-undetermined, Member-of-group, Partner, Subsidiary, Other
Agent-artifact	User-or-owner, Inventor-or-manufacturer, Other
Person-org affiliation	Ethnic, Ideology, Other
GPE affiliation	Citizen-or-resident, Based-in, Other

Relation types in ACE 2004

KBP Slot Filling (2009-2017)

Slot Filling (SF): The **slot filling** task is to search the document collection to **fill** in values for specific attributes ("**slots**") for specific entities

QA style evaluation:
- *What's the age of Barack Obama?*
- *Who is the spouse of Barack Obama?*

Entities do not need to appear in the same sentence

Focus on getting the answer
- System needs to deduplicate answers

Limited training data
- Encouraged semi-supervised methods

Train from Wikipedia with distant supervision!

Person Slots			Organization Slots		
Name	**Type**	**List?**	**Name**	**Type**	**List?**
per:alternate_names	Name	Yes	org:alternate_names	Name	Yes
per:date_of_birth	Value		org:political_religious_affiliation	Name	Yes
per:age	Value		org:top_members_employees	Name	Yes
per:country_of_birth	Name		org:number_of_employees_members	Value	
per:stateorprovince_of_birth	Name		org:members	Name	Yes
per:city_of_birth	Name		org:member_of	Name	Yes
per:origin	Name	Yes	org:subsidiaries	Name	Yes
per:date_of_death	Value		org:parents	Name	Yes
per:country_of_death	Name		org:founded_by	Name	Yes
per:stateorprovince_of_death	Name		org:date_founded	Value	
per:city_of_death	Name		org:date_dissolved	Value	
per:cause_of_death	String		org:country_of_headquarters	Name	
per:countries_of_residence	Name	Yes	org:stateorprovince_of_headquarters	Name	
per:statesorprovinces_of_residence	Name	Yes	org:city_of_headquarters	Name	
per:cities_of_residence	Name	Yes	org:shareholders	Name	Yes
per:schools_attended	Name	Yes	org:website	String	
per:title	String	Yes			
per:employee_or_member_of	Name	Yes			
per:religion	String	Yes			
per:spouse	Name	Yes			
per:children	Name	Yes			
per:parents	Name	Yes			
per:siblings	Name	Yes			
per:other_family	Name	Yes			
per:charges	String	Yes			

Slots in KBP Slot Filling

Relation Extraction by

Learning methods
- Supervised learning
- Semi-supervised approach
- Unsupervised learning
- Weakly/distantly supervised

Approach
- Rule-based
- Machine Learning
- Hybrid methods

Domain
- Pre-defined domains
- Large, collaboratively constructed domains (e.g., Wikipedia)
- Open IE
- On-demand IE

Task design
- Within-sentence vs. cross-sentence
- Binary vs. n-ary
- Between pairs of entities vs. pairs of events

Let’s Started with Supervised Relation Extraction

Rule-based methods
- Write rules to capture different types of relations

Feature-based methods
- Design feature sets for RE and send them to some statistical classifiers (i.e., MaxEnt, SVM)

Kernel-based methods
- Design kernels to compute similarities relation mentions and use them in kernel-based SVM

Deep learning methods
- Let deep learning learn the features for RE from data

Rule-based Approach

Relations Appear in a Wide Range of Forms

Embedded constructs (one argument contains the other)

- **Premodifier** relations specify the proper adjective or proper noun premodifier and the following noun it modifies, e.g.: *the Seattle zoo*
- **Possessive** indicates that the first mention is in a possessive case, e.g.: *California 's Governor*
- **Preposition** indicates that the two mentions are semantically related via the existence of a preposition, e.g.: *officials in California*

Formulaic constructs

- *Tarragona, Spain*
- *Walter Cronkite, CBS News, New York*

Longer-range ('predicate-linked') constructs

- With a predicate disjoint from the arguments
  - *Fred lived in New York*
  - *Fred and Mary got married*

Hand-Crafted Patterns

Most instances of relations can be identified by the types of the entities and the words between the entities

- But not all: *Fred and Mary got married.*

Word sequence patterns (linear patterns) work well enough for short-range relations

- But problems arise for longer-range patterns: greater variety, intervening modifiers

Take advantage of parsing (e.g., PCFG parsers, dependency parsers)

- Arguments of semantic relation generally connected by a limited set of syntactic structures and lexical items
- Need not take into account the wide range of intervening words

Patterns from Parses

Take advantage of parsing (e.g., PCFG parsers, dependency parsers)

- Arguments of semantic relation generally connected by a limited set of syntactic structures and lexical items
- Need not take into account the wide range of intervening words

"Fred shot Mary."

"Fred, 61, shot Mary."

"Fred, tired of her endless lectures on parsing, shot Mary."

All have the same dependency relations:

- verb "shot"
- subject of *shot* = "Fred"
- object of *shot* = "Mary"

Dependency Structures

Root of tree is generally a (tensed) verb
- auxiliaries and modals appear as vch* [verb chain] dependents of tensed verb
- principal arguments appear as
  - nsubj [noun subject]
  - dobj [direct object]
  - iobj [indirect object]
- sentential complements appear as
  - ccomp
  - Xcomp

noun modifiers
- poss [possessive]
- amod [adjective modifiers]
- nn [compound noun]

prepositional phrases: prep and pobj

conj [conjunction]

* The dependency labels are from USC/ISI's Tratz-Hovy dependency parser

Lexicalized Dependency Paths

Path in dependency tree between two entity mentions

combines dependency types and lexical items

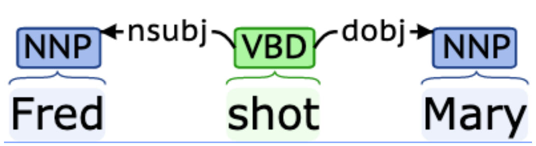
- type = edge from governor to dependent
- type-1 = edge from dependent to governor

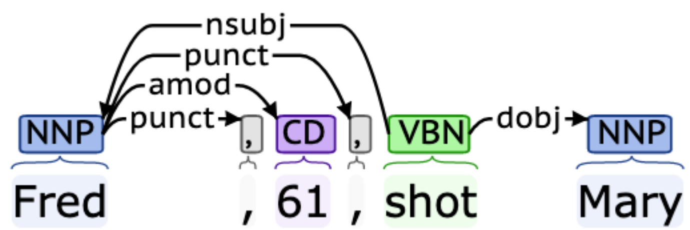
PERSON – nsubj-1:shoot:dobj -- PERSON

"Fred shot Mary."

"Fred, 61, shot Mary."

"Fred, tired of her endless lectures on parsing, shot Mary."





Transformations

Using **dependency paths** (rather than **linear patterns**) greatly increases coverage

Can further (modestly) increase coverage through transformations that connect closely related structures

- operate to simplify dependency parse
- reduce sentences to *kernel sentences* + *transformations*

Transformations

Transformations

- passive:
  - The cake was baked by Harry. → Harry baked the cake.
- relative
  - Harry, who baked the cake → Harry baked the cake
- reduced relative
  - the cake baked by Harry → the cake, which was baked by Harry
- subject control:
  - Harry planned to bake the cake → Harry planned (Harry baked the cake)

Fun project: try developing a pattern-based relation extraction leveraging NER and a dependency parser (both can be found in Stanford CoreNLP)

These can be used as features in a trainable statistical model!

Leverage Syntactic-Semantic Structures for Relation Extraction

Apply patterns to identify the syntactic-semantic structure dimension first, and leverage this in the RE process

Reported +3/5.5 F1 in relation classification, and +4/8.3 F1 in relation detection (vary by the amount of training data used)

Structure type	Pattern
Premodifier	Basic pattern: [u* [v+] w+] , where u, v, w represent words
	Each w is a noun or adjective
	If u* is not empty, then u*: JJ+ $\vee$ JJ "and" JJ? $\vee$ CD JJ* $\vee$ RB DT JJ? $\vee$ RB CD JJ $\vee$ DT (RB|JJ|VBG|VBD|VBN|CD)?
	Let w_1 = first word in w+. $w_1 \neq$ "'s" and POS tag of $w_1 \neq$ POS
	Let v_l = last word in v+. POS tag of $v_l \neq$ PRP$ nor WP$
Possessive	Basic pattern: [u? [v+] w+] , where u, v, w represent words
	Let w_1 = first word in w+. If w_1 = "'s" $\vee$ POS tag of w_1 = POS, accept mention pair
	Let v_l = last word in v+. If POS tag of v_l = PRP$ or WP$, accept mention pair
Preposition	Basic pattern: [m_i] v* [m_j], where v represent words and number of prepositions in the text span v* between them = 0, 1, or 2
	If satisfy pattern: IN [m_i][m_j], accept mention pair
	If satisfy pattern: [m_i] (IN|TO) [m_j], accept mention pair
	If all labels in $\mathcal{L}_d$ start with "prep", accept mention pair
Formulaic	If satisfy pattern: [m_i] / [m_j] $\wedge$ $E_c(m_i)$ = PER $\wedge$ $E_c(m_j)$ = ORG, accept mention pair
	If satisfy pattern: [m_i][m_j] If $E_c(m_i)$ = PER $\wedge$ $E_c(m_j)$ = ORG $\vee$ GPE, accept mention pair

Yee Seng Chan and Dan Roth. Exploiting Syntactico-Semantic Structures for Relation Extraction. ACL 2011

Feature-based Methods

Supervised Learning for RE

Collect training data

- Annotate corpus with entities and relations
- For every pair of entities in a sentence
  - If linked by a relation, treat as positive training instance with the relation type as the label
  - If not linked, treat as a negative training instance

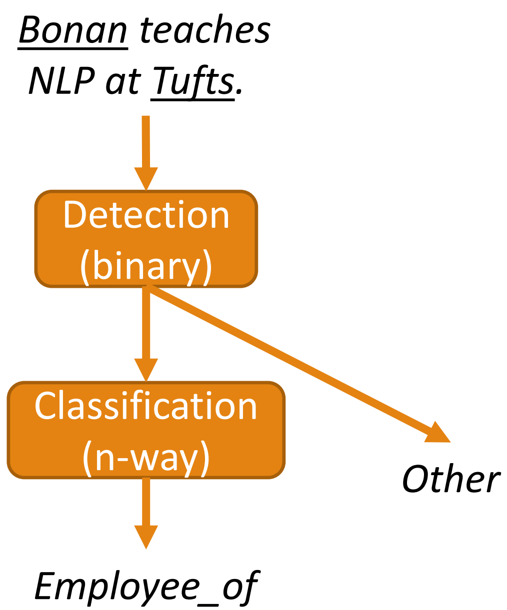
Train model

- For *n* relation types, either
  - Binary (detection) model + *n*-way classifier model (classification) or
  - Unified *n+1*-way classifier
  - Either way, the dataset is very **imbalanced** toward the negative instances ("Other")

On test data

- Apply entity classifier
- Apply relation classifier to every pair of entities in same sentence

Evaluate using Precision, Recall and F1



Supervised Learning for RE

*The spokesman, reporting on the meeting, said **IBM** hired **Fred Smith** as the president.*

Employment

Relation instances

The spokesman, reporting on the meeting, said IBM hired Fred Smith as the president. -> Other

The spokesman, reporting on the meeting, said IBM hired Fred Smith as the president. -> Other

The spokesman, reporting on the meeting, said IBM hired Fred Smith as the president. -> Other

The spokesman, reporting on the meeting, said IBM hired Fred Smith as the president. -> Employment

The spokesman, reporting on the meeting, said IBM hired Fred Smith as the president. -> Employment

The spokesman, reporting on the meeting, said IBM hired Fred Smith as the president. -> Other

Feature-based Methods for RE

Design a set of features, compute the values of such features for each instance, and send them to statistical classifiers for classification

Typical features:

- Heads of entities
- Types of entities
- Distance between entities
- Containment relations
- Word sequence between entities
- Individual words between entities
- Dependency path
- Individual words on dependency path

Zhou et al., 2005. Exploring Various Knowledge in Relation Extraction. ACL 2005

Features

Ray Young, the chief financial officer of *General Motors*, said GM could not bail out Delphi

Designed Features	Values	Designed Features	Values
head word of M1	Ray_Young	last word in between	of
head word of M2	General_ Motors	middle token sequence	, the chief financial officer of
first word before M1	nil	Shortest path connecting M1 and M2 in the dependency parsing tree	PERSON_appos_officer prep_of_ORGANIZATION
second word before M1	nil		
first word after M2	,	entity type of M1	PERSON
second word after M2	said	enity type of M2	ORGANIZATION
first word in between	,	...	….


```
BagWM1_mark=TRUE BagWM1_webster=TRUE BagWM2_itn=TRUE WBFL=of BM1F=first. BM1L=at AM2F=has AM2L=an NUMWB=1 TPatternET=PERSON_of
---PP--of----NP--itn CPatternET=PERSON_of_ORGANIZATION CPHAM2F=has CPHAM2L=update DPathET=PERSON_prep_of_ORGANIZATION ET1DW1=P
alse ET12SamePP=PERSON--ORGANIZATION--false ET12SameVP=PERSON--ORGANIZATION--false orderM=1 HM1=Webster HM2=Itn HM12=Webster--
=NAM--NAM NUMMB=0 ET12M1inM2=PERSON--ORGANIZATION--false ET12M2inM1=PERSON--ORGANIZATION--false HM12M1inM2=Webster--Itn--false
BagWM1_ali=TRUE BagWM2_hospital=TRUE WBFL=and BM1F=in BM1L=weeks AM2F=is AM2L=recovering NUMWB=1 TPatternET=FACILITY_and_PERSO
CPHAM2L=fast DPathET=FACILITY_conj_and_PERSON ET1DW1=PERSON--in ET2DW2=FACILITY--ali H1DW1=ali--in H2DW2=hospital--ali ET12Sam
=2 HM1=Ali HM2=hospital HM12=Ali--hospital ET1=PERSON ET2=FACILITY ET12=PERSON--FACILITY EST12=Individual--Building-Grounds ML
lse HM12M1inM2=Ali--hospital--false HM12M2inM1=Ali--hospital--false detectorLabel=1 classLabel=PHYS--Located
```

Features: Brown Word Clustering

The Brown algorithm (a hierarchical clustering algorithm):

- Initially assigns each word to its own cluster
- Repeatedly merges the two clusters which cause the least loss in average mutual information between adjacent clusters based on bigram statistics
- By tracing the pairwise merging steps, one can obtain a word hierarchy which can be represented as a binary tree

Use prefixes of the bit strings of the heads of the entity mentions as the features (i.e., HM1_WC2, HM2_WC4)

Bit string	Examples
111011011100	***US*** …
1110110111011	***U.S.*** …
1110110110000	***American*** …
1110110111110110	***Cuban***, *Pakistani, Russian* …
11111110010111	***Germany***, *Poland, Greece* …
110111110100	***businessman***, *journalist, reporter*
1101111101111	***president,*** *governor, premier*…
1101111101100	***senator***, *soldier, ambassador* …
11011101110	***spokesman***, *spokeswoman,* …
11001100	***people***, *persons, miners, Haitians*
110110111011111	***base***, *compound, camps, camp* …
110010111	***helicopters***, *tanks, Marines* …

Type	P		R		F	
	Baseline	PC4 (△)	Baseline	PC4 (△)	Baseline	PC4 (△)
EMP-ORG	75.4	77.2(+1.8)	79.8	81.5(+1.7)	77.6	79.3(+1.7)
PHYS	73.2	71.2(-2.0)	61.6	60.2(-1.4)	66.9	65.3(-1.7)
GPE-AFF	67.1	69.0(+1.9)	60.0	63.2(+3.2)	63.3	65.9(+2.6)
PER-SOC	88.2	83.9(-4.3)	58.4	61.0(+2.6)	70.3	70.7(+0.4)
DISC	79.4	80.6(+1.2)	42.9	46.0(+3.2)	55.7	58.6(+2.9)
ART	87.9	96.9(+9.0)	63.0	67.4(+4.4)	73.4	79.3(+5.9)
OTHER-AFF	70.6	80.0(+9.4)	41.4	41.4(0.0)	52.2	54.6(+2.4)

Sun at al., 2011: Semi-supervised Relation Extraction with Large-scale Word Clustering (ACL)

Features: Word Embeddings

Generalizing the head words of the entity mentions seems to be very helpful for RE

Use word embeddings to achieve such generalization (i.e., using the word embeddings of the heads as the features)

Without regularization:

System	In-domain	bc	cts	wl
Baseline(B)	51.4	49.7	41.5	36.6
B+WC10	52.3(+0.9)	50.8(+1.1)	45.7(+4.2)	39.6(+3)
B+WC	53.7(+2.3)	52.8(+3.1)	46.8(+5.3)	41.7(+5.1)
B+ED	54.1(+2.7)	52.4(+2.7)	46.2(+4.7)	42.5(+5.9)
B+WC+ED	**55.5(+4.1)**	**53.8(+4.1)**	**47.4(+5.9)**	**44.7(+8.1)**

With regularization:

System	In-domain	bc	cts	wl
Baseline(B)	56.2	55.5	48.7	42.2
B+WC10	57.5(+1.3)	57.3(+1.8)	52.3(+3.6)	45.0(+2.8)
B+WC	58.9(+2.7)	58.4(+2.9)	52.8(+4.1)	47.3(+5.1)
B+ED	58.9(+2.7)	59.5(+4.0)	52.6(+3.9)	48.6(+6.4)
B+WC+ED	**59.4(+3.2)**	**59.8(+4.3)**	**52.9(+4.2)**	**49.7(+7.5)**

Thien Huu Nguyen and Ralph Grishman. Employing Word Representations and Regularization for Domain Adaptation of Relation Extraction. ACL 2014

Kernel-based Methods

Kernel-based Methods for RE

Goal is to find training examples similar to test case

- Need similarity metrics between pairs of relation instances
- Determining similarity through features is awkward
  - Feature engineering is laborious
- Better to define a similarity measure directly: a kernel function

Kernels can be used directly by

- SVMs
- Memory-based learners (k-nearest-neighbor)

For RE, kernels defined over

- Strings, marked with left and right mentions
- Parse or Dependency Trees, marked with left and right mentions

String Kernels

Two strings are more similar if they share more substrings

Linear combination parameter

Decaying factor
$0 < \lambda \leq 1$

$$k(s_i, s_j) = \sum_n c_n k_n(s_i, s_j)$$

$$k_n(s_i, s_j) = \sum_{u \in \Sigma^n} \sum_{u = \mathbf{o}_{s_i}} \sum_{u = \mathbf{p}_{s_j}} \lambda^{l(\mathbf{o}_{s_i})} \lambda^{l(\mathbf{p}_{s_j})}$$

Sets of strings of length *n*

Length of the string

Many variants are possible

String Subsequence Kernels

Patterns of words/sequences that involved in relations

- **[FB] F**ore–**B**etween: words before and between the two entity mentions are simultaneously used to express the relationship. Examples: 'interaction of $\langle P_1 \rangle$ with $\langle P_2 \rangle$', 'activation of $\langle P_1 \rangle$ by $\langle P_2 \rangle$'.
- **[B] B**etween: only words between the two entities are essential for asserting the relationship. Examples: '$\langle P_1 \rangle$ interacts with $\langle P_2 \rangle$', '$\langle P_1 \rangle$ is activated by $\langle P_2 \rangle$'.
- **[BA] B**etween–**A**fter: words between and after the two entity mentions are simultaneously used to express the relationship. Examples: '$\langle P_1 \rangle$ – $\langle P_2 \rangle$ complex', '$\langle P_1 \rangle$ and $\langle P_2 \rangle$ interact'.

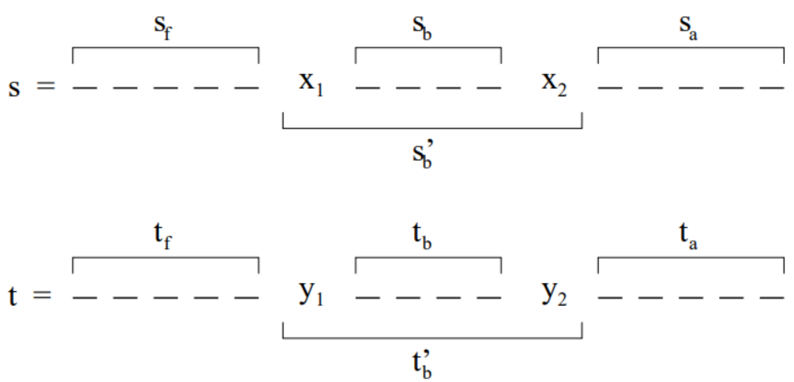


Figure 2: Sentence segments.

$$
\begin{aligned}
rK(s,t) &= fbK(s,t) + bK(s,t) + baK(s,t) \\
bK_i(s,t) &= K_i(s_b, t_b, 1) \cdot c(x_1, y_1) \cdot c(x_2, y_2) \cdot \lambda^{l(s_b^{'}) + l(t_b^{'})} \\
fbK(s,t) &= \sum_{i,j} bK_i(s,t) \cdot K_j^{'}(s_f, t_f), \quad 1 \le \mathrm{i},\ 1 \le \mathrm{j},\ \mathrm{i} + \mathrm{j} < \mathrm{fb}_{\max} \\
bK(s,t) &= \sum_{i} bK_i(s,t), \quad 1 \le \mathrm{i} \le \mathrm{b}_{\max} \\
baK(s,t) &= \sum_{i,j} bK_i(s,t) \cdot K_j^{'}(s_a^{-}, t_a^{-}), \quad 1 \le \mathrm{i},\ 1 \le \mathrm{j},\ \mathrm{i} + \mathrm{j} < \mathrm{ba}_{\max}
\end{aligned}
$$

Subsequence kernel for RE

Razvan C. Bunescu and Raymond J. Mooney. 2006. Subsequence Kernels for Relation Extraction. NIPS.

Tree Kernels

Compute the number of common subtrees:

let N_1 and N_2 be the set of nodes in T_1 and T_2 respectively, then

$$TK_\sigma(T_1, T_2) = \sum_{n_1 \in N_1, n_2 \in N_2} \Delta(n_1, n_2)$$

where $\Delta(n_1, n_2)$ is computed by:

i) if n_1 and n_2 have different productions: $\Delta(n_1, n_2) = 0$; else
ii) if n_1 and n_2 are pre-terminals: $\Delta(n_1, n_2) = \lambda$; else
iii) $\Delta(n_1, n_2) = \lambda \prod_{j=1}^{nc(n_1)} (1 + \Delta(ch(n_1, j), ch(n_2, j)))$

T_1, T_2 can be either constituent or dependency trees. The trees can be pruned to minimally cover the two entity mention of interest.

Can incorporate with word clusters and word embeddings

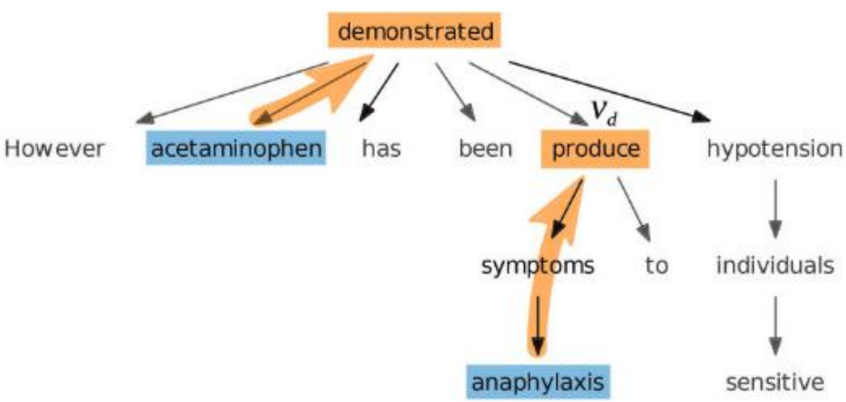
$\Phi\{$  ,  $\} = 0.95$

Tree Kernels in SVN-light:

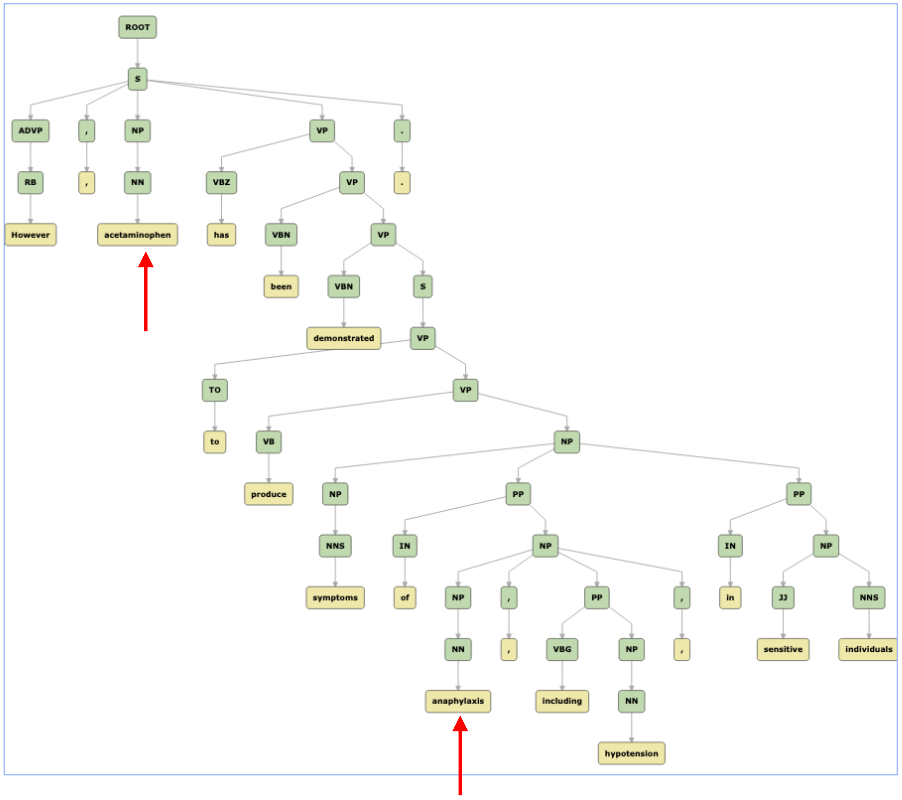
http://disi.unitn.it/moschitti/Tree-Kernel.htm

Tree Kernels

However, acetaminophen has been demonstrated to produce symptoms of anaphylaxis, including hypotension, in sensitive individuals.

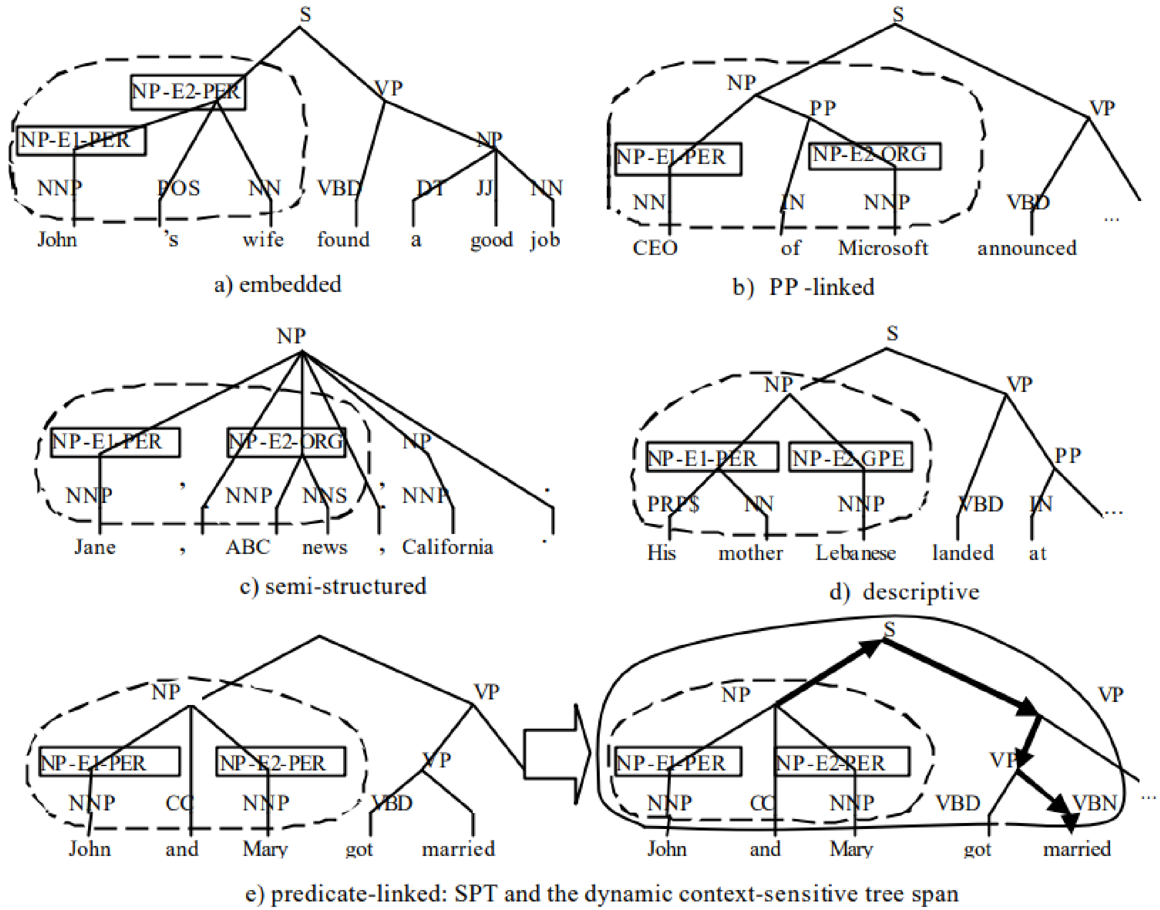


The dependency tree



The constituent tree

Tree Kernels

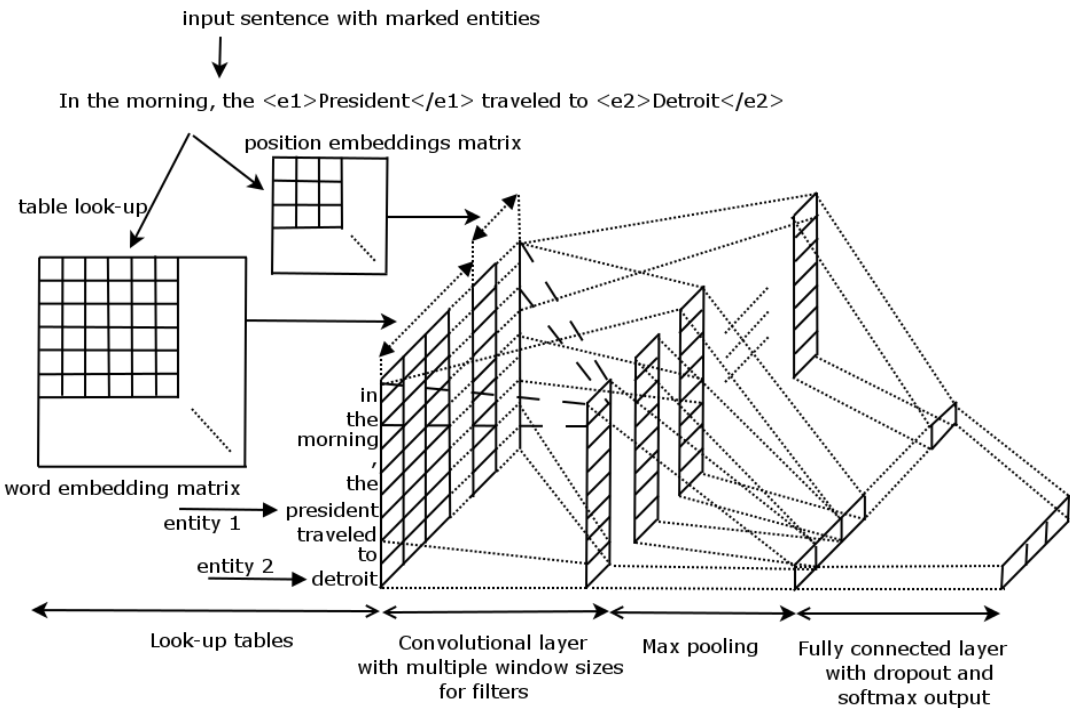


Zhou et al. Tree Kernel-based Relation Extraction with Context-Sensitive Structured Parse Tree Information. EMNLP-CoNLL 2007.

Deep Learning for RE

Deep Learning for RE

Avoid feature or kernel design for RE



A Convolutional Neural Network (CNN) for Relation Extraction (Nguyen and Grishman, 2015)

Classifier	Features	F
MaxEnt	POS, WordNet, morphological features, noun compound sys- tem, thesauri, Google n -grams	77.6
SVM	POS, WordNet, prefixes and other morphological fea- tures, dependency parse, Levin classes, PropBank, FrameNet, NomLex-Plus, Google n - grams, paraphrases, TextRunner	82.2
CNN (Zeng et al., 2014)	WordNet	82.7
CNN (Nguyen and Grishman, 2015a)	-	82.8

Performance on SemEval 2010

Position Embeddings

To inform the models about the two entity mentions of interest, we introduce (relative) position embeddings (randomly initialized and updated during training)

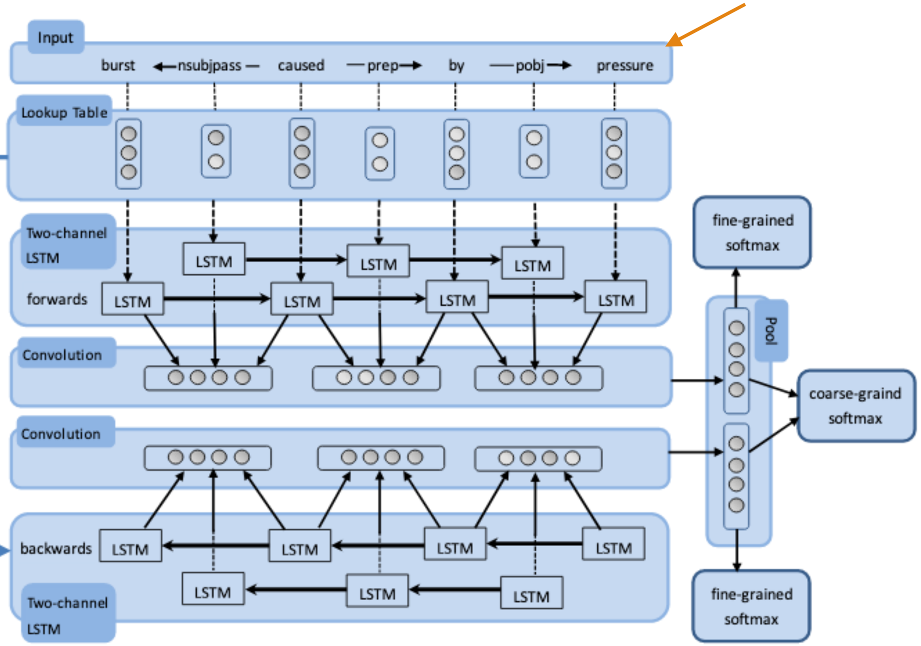
Dist from M1	0	1	2	3	4
Dist from M2	-4	-3	-2	-1	0
	[Bonan Min]	*teaches*	*NLP*	*at*	*[Tufts]*

-4	2	-0.5	1.1	0.3	0.4	-0.5
-3	-1.4	0.4	-0.2	-0.9	0.5	0.9
-2	-1.1	-0.2	-0.5	0.2	-0.8	0
-1	0.7	-0.3	1.5	-0.3	-0.4	0.1
0	-0.8	1.2	1	-0.7	-1	-0.4
1	0	0.3	-0.3	-0.9	0.2	1.4
2	0.8	0.8	-0.4	-1.4	1.2	-0.9
3	1.6	0.4	-1.1	0.7	0.1	1.6
4	1.2	-0.2	1.3	-0.4	0.3	-1.0

Deep Learning for RE

Can also incorporate syntax into deep learning models for RE: to identify important context words (i.e., via the dependency paths) or to guide the computational flows of the neural network models.

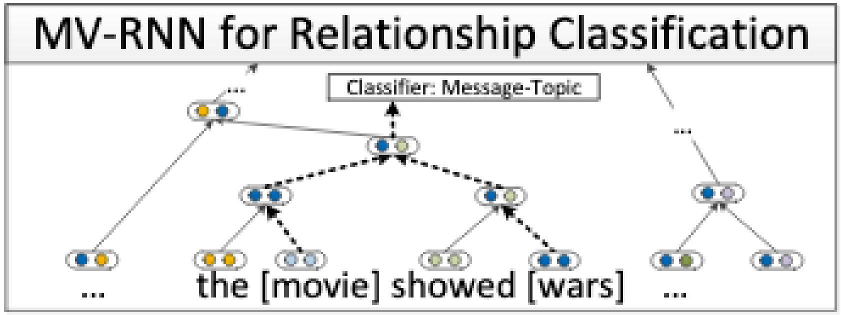
the shortest dependency path between two entity mentions



Cat et al., Bidirectional Recurrent Convolutional Neural Network for Relation Classification (ACL 2016)

Recursive neural networks: building the networks based on the constituent trees

the binarized constituent subtree



Socher et al., Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank (EMNLP 2013)

Syntactic Structures for Relation Extraction

Graph Convolutional Neural Network (GCN) over dependency trees for RE (a recent state-of-the-art approach for RE) (Zhang et al., 2018)