

Probabilistic Reasoning for Entity & Relation Recognition

M. Trognitz

5. Februar 2009

- 1 Einleitung und Motivation
- 2 Problem und Umsetzung
- 3 Testen und Evaluation
- 4 Diskussion

Überblick

- 1** Einleitung und Motivation
- 2 Problem und Umsetzung
- 3 Testen und Evaluation
- 4 Diskussion

- Dan Roth and Wen-tau Yih (2002). *Probabilistic Reasoning for Entity & Relation Recognition*, Proceedings of the 19th International Conference on Computational Linguistics (COLING'02), 1:1-7.

Überblick

- Methode zur Erkennung von Relationen und Entitäten in Sätzen
- Gegenseitige Abhängigkeit wird beachtet
 - kill(Johns, Oswald)
 - J.V. Oswald was murdered at JFK after his assassin, K. F. Johns...
 - *people*: Johns, Oswald
location: JFK;
relation: kill(John, Oswald)

Überblick

- Methode zur Erkennung von Relationen und Entitäten in Sätzen
- Gegenseitige Abhängigkeit wird beachtet
 - kill(Johns, Oswald)
 - J.V. Oswald was murdered at JFK after his assassin, K. F. Johns...
 - *people*: Johns, Oswald
location: JFK;
relation: kill(John, Oswald)

Überblick

- Die Relationen und Entitäten werden aus den Sätzen gelernt (*lokal*)
- Die Abhängigkeiten werden mit dieser Information und Constraints, die aus den Entitätstypen und den Relationen induziert werden, abgeleitet (*global*)

Motivation

- Information Extraction
 - Entitäten müssen extrahiert werden
- Question Answering
 - *Where was Poe born?*
(fragt nach einem Ort)
 - *Who killed Lee Harvey Oswald?*
(fragt nach Personen, die in einer bestimmten Relation zueinander stehen)
- Story comprehension

Motivation

- Bisher: zuerst Entitäten erkennen, dann Relationen herausfinden
 - Fehler aus *named entity recognizer* führen zu weiteren Fehlern beim Klassifizieren der Relationen
 - Wenn Boston als Person erkannt wird, kommt Boston nicht mehr als *Geburtsort* in Frage
- Dieser Ansatz erkennt Entitäten und Relationen gleichzeitig

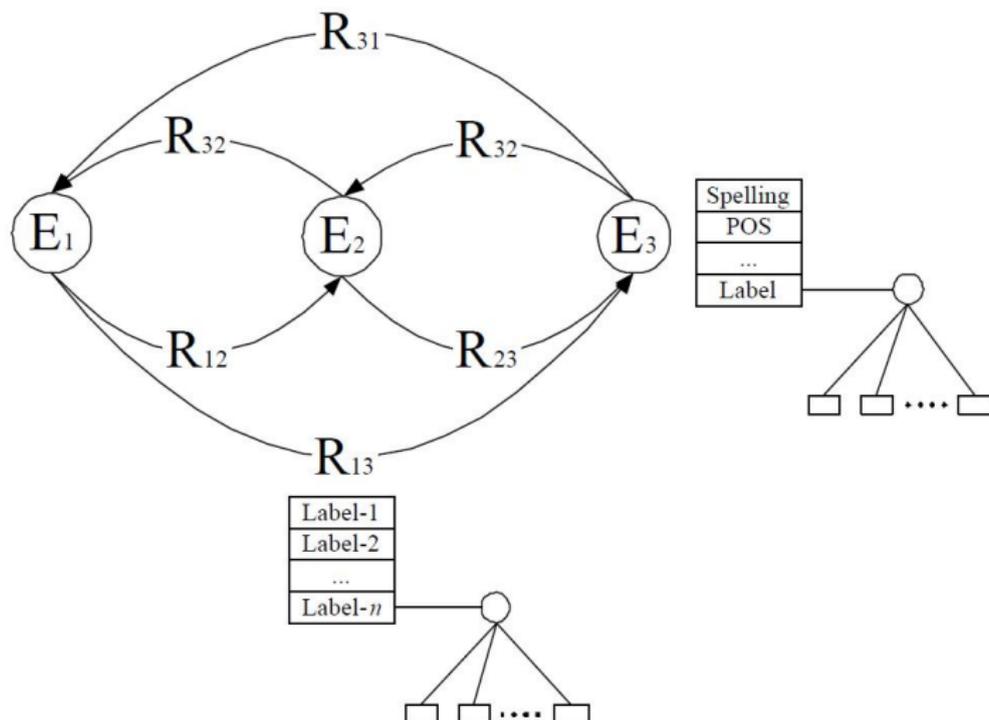
Überblick

- 1 Einleitung und Motivation
- 2 Problem und Umsetzung**
- 3 Testen und Evaluation
- 4 Diskussion

Das Problem

- In einem gegebenen Satz sollen die Entitäten und Relationen kohärent gelabelt werden
- Einige Informationen kommen direkt aus dem Input
- Einige Informationen können leicht gelernt werden
- Die Benennung der Entitäten und Relationen ist das Ziel

Das Problem



Definitionen - Entität

- Ein Satz S ist eine verlinkte Liste, die aus Wörtern w und Entitäten E besteht
- Eine Entität kann ein einzelnes Wort oder eine Wortfolge mit einer vordefinierten Grenze sein und nimmt den Wert aus der Menge der Entitätstypen C^E an
- *Dole* 's wife, *Elizabeth*, is a native of *Salisbury, N.C.*
- $E_1 = \text{Dole}$, $E_2 = \text{Elizabeth}$, $E_3 = \text{Salisbury}$
- Die Grenzen der Entitäten werden als gegeben vorausgesetzt

Definitionen - Relation

- Eine binäre Relation $R_{ij} = (E_i, E_j)$ repräsentiert die Relation zwischen E_i und E_j
- R_{ij} ist eine Relation aus der Menge der Relationstypen C^R
- *Dole* 's wife, *Elizabeth*, is a native of *Salisbury, N.C.*
- 6 Relationen:
 - $R_{12} = (Dole, Elizabeth)$
 - $R_{21} = (Elizabeth, Dole)$
 - $R_{13} = (Dole, Salisbury, N.C.)$
 - $R_{31} = (Salisbury, N.C., Dole)$
 - $R_{23} = (Elizabeth, Salisbury, N.C.)$
 - $R_{32} = (Salisbury, N.C., Elizabeth)$

Definitionen - Klasse

- Die Menge der vorgegebenen Entitäts- und Relationsklassen werden mit C^E und C^R gekennzeichnet
- C^E enthält ein spezielles Element „other_ent“, das für nicht definierte Klassen steht
- C^R enthält ein spezielles Element „other_rel“, das für nicht definierte oder nicht relevante Klassen steht

Definitionen - Klasse

- *Dole* 's wife, *Elizabeth*, is a native of *Salisbury, N.C.*
- $E_1 = \text{Dole}$, $E_2 = \text{Elizabeth}$, $E_3 = \text{Salisbury}$
 - $C^E = \{\text{other_ent}, \text{person}, \text{location}\}$
 - $C^R = \{\text{other_rel}, \text{born_in}, \text{spouse_of}\}$
 - $E_1, E_2 : \text{person}$
 - $E_3 : \text{location}$
 - $R_{12} = (\text{Dole}, \text{Elizabeth}) : \text{spouse_of}$
 - $R_{21} = (\text{Elizabeth}, \text{Dole}) : \text{spouse_of}$
 - $R_{23} = (\text{Elizabeth}, \text{Salisbury, N.C.}) : \text{born_in}$
 - $R_{13} = (\text{Dole}, \text{Salisbury, N.C.}),$
 $R_{31} = (\text{Salisbury, N.C.}, \text{Dole}),$
 $R_{32} = (\text{Salisbury, N.C.}, \text{Elizabeth}) : \text{other_rel}$

Klassifizierung

- Klassifizierung nicht trivial, da eine Entität nicht nur von ihren Eigenschaften abhängt, sondern auch von den Eigenschaften anderer Entitäten und Relationen
- z. B. hängt E_1 von R_{12} ab, wobei R_{12} auch noch von E_1 und E_2 abhängt
- Zur Evaluationszeit sind nicht alle Daten annotiert, nur einige lokale Eigenschaften sind gegeben (Wort, POS, ...)
- Vereinfachung: wahrscheinlichkeitstheoretischer Rahmen

Klassifizierung

- Zunächst werden die Klassifizierer unabhängig voneinander trainiert
- Das Ergebnis wird als bedingte Verteilung von Entitäten und Relationen verwendet
- Daraus werden die Labels der Entitäten und Relationen gefolgert
- Die Label von Entitäten und Relationen müssen bestimmte *constraints* erfüllen
- Wenn E_1 als erstes Argument von R_{12} ein *Ort* ist, dann kann R_{12} nicht *born_in* sein, da hier das erste Argument eine *Person* sein muss

Definitionen - Constraint

- Ein Constraint \mathcal{C} ist ein Tripel $(\mathcal{R}, \mathcal{E}^1, \mathcal{E}^2)$, mit $\mathcal{R} \in \mathcal{C}^R$ und $\mathcal{E}^1, \mathcal{E}^2 \in \mathcal{C}^E$
- *(born_in, person, location), (spouse_of, person, person), (murder, person, person)*
- Sie können mit einer gemeinsamen Wahrscheinlichkeitsverteilung modelliert werden
- Es werden nur einige der bedingten Wahrscheinlichkeiten berechnet

Definitionen - Constraint

- $P(R_{ij}|E_i, E_j)$ mit gegebenen Labels
- $P(R_{ij} = other_rel|E_i = e^1, E_j = e^2) = 1$, wenn kein r , so dass (r, e^1, e^2) ein constraint ist
- $P(R_{ij} = r|E_i = e^1, E_j = e^2) = 0$, wenn kein constraint c , so dass $c = (r, e^1, e^2)$
- Die bedingten Wahrscheinlichkeiten können von einem annotierten Trainingsset gelernt werden

Problem - Formulierung

- Es muss also nach den wahrscheinlichsten Besetzungen für E und R gesucht werden
- $(e_1, e_2, \dots, e_n, r_{12}, r_{21}, \dots, r_{n(n-1)}) = \arg \max_{e_i, r_{jk}} \text{Prob}(E_1, \dots, E_n, R_{12}, R_{21}, \dots, R_{n(n-1)})$

Umsetzung

- Zunächst werden die Klassifizierer für die Entitäten und Relationen gelernt
- *named entity classifier*: für Entitäten, die Grenzen sind vorgegeben
- *relation classifier*: erhält die Entitäten, die die Zielrelation kennzeichnen
- Es werden *weak classifiers* durch *shallow text processing* erzeugt, da sonst zu aufwändig
- Die Labels werden mit lokalen Eigenschaften erzeugt (z.B. mit Hilfe von WordNet)

Umsetzung

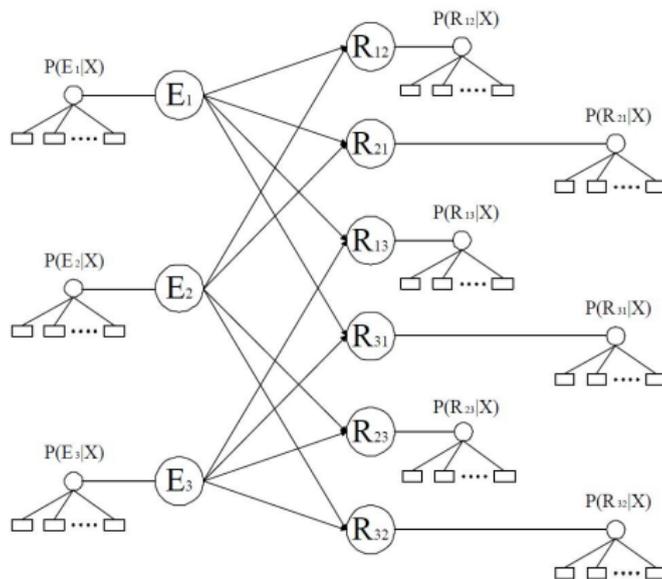
- SNoW
 - *multi-class classifier*
 - Für große Lernaufgaben gemacht
 - Verwendung eines Netzwerks aus linearen Funktionen über verschiedene Eigenschaften
 - Es wird nur der *activation value* verwendet, um die folgenden Eigenschaften abzuschätzen
 - $$p_i = \frac{e^{act_i}}{\sum_{1 \leq j \leq n} e^{act_j}}$$

Bayesian Inference Model

- Mit den Klassifizierern werden $Prob(E|observation)$ und $Prob(R|observation)$ erzeugt
- Mit Hilfe eines Bayes'schen Netzes werden dann die wahrscheinlichsten Vorhersagen für die Labels getroffen
- Bayes'sches Netz
 - Gerichteter azyklischer Graph
 - Knoten sind Zufallsvariablen
 - Jeder Knoten ist mit einer Tabelle der bedingten Wahrscheinlichkeit belegt, die von den Elterknoten abhängig ist
 - Hier: bipartiter Graph

Bayes'sches Netz

- Graph für drei Entitäten und sechs Relationen



Bayesian Inference Model

- Um die wahrscheinlichsten Labels zu finden, reicht es, die Variablen im Netz zu finden, die die gemeinsame Wahrscheinlichkeitsverteilung maximieren
- Das Netzwerk enthält ungerichtete Zyklen, weshalb der *belief propagation algorithm* von Pearl verwendet wird
- Durch Iteration wird das Ergebnis meistens erreicht

Überblick

- 1 Einleitung und Motivation
- 2 Problem und Umsetzung
- 3 Testen und Evaluation**
- 4 Diskussion

Daten

- Sätze aus TREC, hauptsächlich journalistische Texte
- Davon enthalten 245 die Relation *kill* und 179 die Relation *born_in*
- 502 Sätze ohne Relationen sind ebenfalls enthalten

Daten - Beispiel

0	0	0	NP	DT	The	NOFUNC	x	0			
0	B-Unknown	1	NP	NNP/NNP	Warren/Commission				NOFUNC	x	0
0	0	2	0	VBD	determined	NOFUNC	x	0	0		
0	0	3	0	IN	that	NOFUNC	x	0			
Arg1	B-Peop	4	NP	NNP/NNP/NNP	Lee/Harvey/Oswald				NOFUNC	x	0
0	0	5	0	VBD	fired	NOFUNC	x	0			
0	0	6	NP	DT	a	NOFUNC	x	0			
0	0	7	NP	JJ	high-powered	NOFUNC	x	0	0		
0	0	8	NP	NN	rifle	NOFUNC	x	0			
0	0	9	0	IN	at	NOFUNC	x	0			
Arg2	B-Peop	10	NP	NNP	Kennedy	NOFUNC	x	0			
0	0	11	0	IN	from	NOFUNC	x	0			
0	0	12	NP	DT	the	NOFUNC	x	0			
0	0	13	NP	JJ	sixth	NOFUNC	x	0			
0	0	14	NP	NN	floor	NOFUNC	x	0			
0	0	15	0	IN	of	NOFUNC	x	0			
0	0	16	NP	DT	the	NOFUNC	x	0			
0	0	17	NP	NN	building	NOFUNC	x	0	0		
0	0	18	0	WRB	where	NOFUNC	x	0			
0	0	19	NP	PRP	he	NOFUNC	x	0			
0	0	20	0	VBD	worked	NOFUNC	x	0			
0	0	21	0	IN	on	NOFUNC	x	0			
0	B-Unknown	22	NP	NNP	Nov.	NOFUNC	x	0	0		
0	0	23	NP	CD	22	NOFUNC	x	0			
0	0	24	NP	,	,	NOFUNC	x	0			
0	0	25	NP	CD	1963	NOFUNC	x	0			
0	0	26	0	.	.	NOFUNC	x	0			
4	10	kill									



Tests

- Drei Testmethoden
 - basic: testet die Baseline und die Leistung der *basic classifiers*
 - omniscient: ähnlich wie basic, wobei die gefundenen Labels dem jeweils anderen Klassifizierer mitgeteilt werden
 - BN: testet die Fähigkeit globale Inferenzen zu machen
- Mit 5-fold validation

Ergebnisse

Approach	person			location		
	Rec	Prec	F_1	Rec	Prec	F_1
Basic	96.6	92.3	94.4	76.3	91.9	83.1
BN	89.0	96.1	92.4	78.8	86.3	82.1
Omniscient	96.4	92.6	94.5	75.4	90.2	81.9

Approach	kill		
	Rec	Prec	F_1
Basic	61.8	57.2	58.6
BN	49.8	85.4	62.2
Omniscient	67.7	63.6	64.8

- Test auf dem *kill*-Datensatz

Ergebnisse

Approach	person			location		
	Rec	Prec	F_1	Rec	Prec	F_1
Basic	85.5	90.7	87.8	89.5	93.2	91.1
BN	87.0	90.9	88.8	87.5	93.4	90.3
Omniscient	90.6	93.4	91.7	90.7	96.5	93.4

Approach	born_in		
	Rec	Prec	F_1
Basic	81.4	63.4	70.9
BN	87.6	70.7	78.0
Omniscient	86.9	71.8	78.0

- Test auf dem *born_in*-Datensatz

Ergebnisse

Approach	person			location		
	Rec	Prec	F_1	Rec	Prec	F_1
Basic	92.1	87.0	89.4	83.2	81.1	82.0
BN	78.8	94.7	86.0	83.0	81.3	82.1
Omniscient	93.4	87.3	90.2	83.5	83.1	83.2
Approach	kill			born_in		
	Rec	Prec	F_1	Rec	Prec	F_1
Basic	43.8	78.6	55.0	69.0	72.9	70.5
BN	47.2	86.8	60.7	68.4	87.5	76.6
Omniscient	52.8	79.5	62.1	76.1	71.3	73.2

- Test auf dem *all*-Datensatz

Besonderheiten

- Der *belief network* Ansatz verringert den Recallwert ein wenig, erhöht den Precisionwert aber um so mehr
 - Wenn eine Relation nicht mit *oter_rel* gelabelt wird, dann müssen die Entitäten deren *constraints* erfüllen
- Die Klassifizierung der Relationen hilft dem Klassifizierer der Entitäten nicht viel
 - Nur wenige der Entitäten eines Satzes sind in einigen Relationen enthalten

Überblick

- 1 Einleitung und Motivation
- 2 Problem und Umsetzung
- 3 Testen und Evaluation
- 4 Diskussion**

Erweiterungsmöglichkeiten

- Eine Art *boot-strapping* mit diesem Ansatz ausführen, um neue Daten zu annotieren
- Verbesserungen
 - Zyklen erlauben
 - andere probabilistische Modelle austesten