SCRIPT ACQUISITION

Presentation of the seminar Recent Developments in Computational Semantics 2010-2011

Angeliki Lazaridou, Saarland University

What is a script?

Script is a standardized sequence of events that describes some stereotypical human activity such as going to a restaurant or visiting a doctor

(Barr and Feigenbaum, 1981)

Motivation

- Inference
- Textual undestanding
- Narrative generation

Outline

- Unsupervised Learning of Narrative Events Chains
  - Narrative Chain Model (narrative chain, narrative event, protagonist)
  - Learning Narrative Relations
  - Ordering Narrative Events
- Learning Script Knowledge with Web Experiments
  - Terminology
  - Data Acquisition
  - Constructing Temporal Script Graphs
  - Experiments and Results
- Unsupervised Learning of Narrative Scemas and their Participants
  - Typed Narrative Chains
  - Learning Argument Types
  - Narrative Scemas

Unsupervised Learning of Narrative Events Chains

Nathanael Chambers & Dan Jurafsky
ACL 2008

Unsupervised Learning of Narrative Events Chains

Nathanael Chambers & Dan Jurafsky
ACL 2008

Narrative Chain Model

John was arrested by the police two years ago.

He was accused for stealing. That ended up in convicted him and finally, he was sentenced for 2 years.

- Narrative event : Tuple of an event v (e.g verb) and the typed dependency, s.t. d ∈ {subject, object, preposition}
- Protagonist : The central actor who characterizes a narrative chain
- Narrative chain *(L,O)* : Partially ordered set of narrative events

  L = { (arrested, subj), {accused, subj), (convicted, obj), (sentenced, subj)}

  O = {(arrested, accused), (accused, convicted), (convicted, sentenced)}

Narrative Chain Model

John was arrested by the police two years ago.

He was accused for stealing. That ended up in convicted him and finally, he was sentenced for 2 years.

- **Assumption of narrative coherence**

  verbs sharing coreferring arguments are semantically connected by virtue of narrative discourse structure => verbs that share arguments are most likely to participate in the same narrative chain

- Distributional learning **vs** Narrative Learning

  Whether two verbs are related — Information about the participants

Learning Narrative Event Chains

3-step process from raw newswire text

Narrative event induction: Learn Narrative Relations

Temporal ordering: Order the connected events with respect to time

Structured Selection : Prune and cluster self-contained chains from space of events

Narrative Event Induction

- **Goal:** learn the protagonist and subevents
- Extract pairwise relation of events that share **grammatical arguments** using a similarity mesure
- **Similarity mesure:** Given e(w,d) and e(v,g) where w,v are verbs and d,g dependencies (e.g e(push, subject)) calculate

$$pmi(e(w,d),e(v,g)) = log\frac{P(e(w,d),e(v,g))}{P(e(w,d))P(e(v,g))}$$

where P(e(w,d),e(v,g)) is calculated as

$$P(e(w,d),e(v,g)) = \frac{C(e(w,d),e(v,g))}{\sum_{x,y}\sum_{d,f}C(e(x,d),e(y,f))}$$

#of times the events had the dependencies d,g filled with arguments referring to the same entity

#of times the dependencies d,f in the events were referring to the same entity

Narrative Event Induction

- For find the next most likely event:
  - Use the scores calculated from the previous process
  - Find which event from the m candidates maximizes the sum of the pmi of this event with all the other events of the chain size n

$$\max_{j:0<j<m} \sum_{i=0}^{n} pmi(e_i, f_j)$$

Narrative Event Induction Experimental Setup

- Evaluation metric:
  - Based on cloze task (Taylor, 1953)
  - **Narrative cloze:** Having sequence of narrative events, remove one and predict the **verb** and **dependency** that is missing
- Dataset: Gigaword Corpus
  - **Training corpus**: Documents from 1994-2004
  - **Development set: manually** selected to capture various topics
  - **Testing corpus**: randomly selected 69 stories(containing a narrative chain > 5) from a randomly selected year (2001)

Narrative Event Induction Experimental Setup

- Experiment:
  - For training:
    - Parse into typed dependencies and resolve entity mentions
    - Create PMI counts
  - For testing:
    - Hand-selected narrative chain, remove one event from chain and generate ranked list with the guesses
    - 740 cloze tests (69 narratives with 740 events)

Known events:
(pleaded subj), (admits subj), (convicted obj)

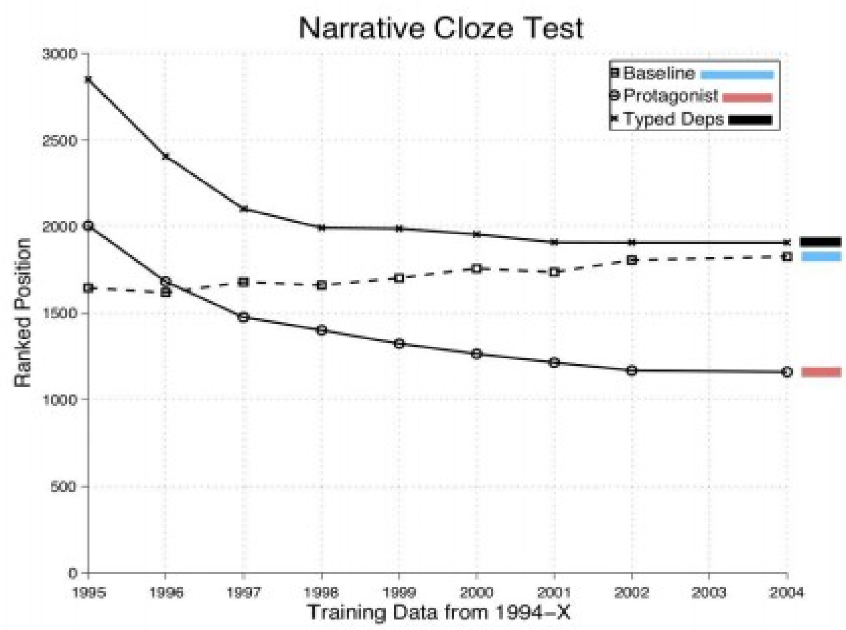
Likely Events:

sentenced obj	0.89	indicted obj	0.74
paroled obj	0.76	fined obj	0.73
fired obj	0.75	denied subj	0.73

Given the chain and the top6 ranked guesses if the correct event was **fired** then the score would be 3. Unseen events are penalized with the length of the ranked list

Narrative Event Induction Results

Use only information about common arguments.



More difficult task. Very informative in order to rebuilt the story.

Pure verb co-occurence in document level. Easier task but not informative

- Measuring average ranked performance
- Approximatly 9.000 candidates for each ranked list (716/740 removed events in the respective ranked list)
- Protagonist model & Typed Deps model: The more training data, the better for the performance
- Protagonist model: 36% improvment over baseline
- 3.5 % **decrease** of performance **without** coreference tool

Ordering Narrative Events

- 2-stage process based on previous work (Chambers et al. 2007)

  1st stage: Label temporal attributes of events (e.g tense)

  - Train an SVM using as features for the classifier neigbouring POS, auxiliaries and modals, and WordNet synsets

  2nd stage: Classify temporal relation events (*before* relation or *other*)

  - Combine SVM from first stage with linguistic features

Ordering Narrative Events Experimental Setup

- Training Dataset:
  - Timebank corpus (37519 of *before* relations)
  - Applying transitivity rules and extend Timebank to 45619 relations
- Testing Dataset:
  - Gigaworld corpus (69 documents as before)
  - **Hand-labeled** with temporal ordering (remove 6 documents because of no ordered event)
- Experiment:
  - First classify temporal features **of each event**
  - *Before* relation between **all events**
  - Every pair of events is a different classifiaction task
- Evaluation: compare the predicted order with (up to) 300 random orderings for every document.

Ordering Narrative Events Results

Length of narrative chain

	All	≥ 6	≥ 10
correct	8086 **75%**	7603 **78%**	6307 **89%**
incorrect	1738	1493	619
tie	931	627	160

- Coherence score: sum of correct relations weighted by the confidence score.
- Compute coherence score for their output and 300 random
- If their score is higher from the 300 random, they are correct
- The bigger the chain, the better for their method(unfair for random, maybe compare with more permutations)

Recap

- What we saw:
  - An unsupervised method for learning ordered narrative event chains
  - No information about type of protagonist
  - Only one protagonist
  - Some scripts like SHOPPING are shared implicit knowledge and are rarely elaborated in text

Learning Script Knowledge with Web Experiments

Regneri M, Koller A.,Pinkal M.
ACL 2010

Introduction

- Goal:
  - Learn paraphrases for the same events
  - Learn what constraints should hold on the temporal order

- Key Idea
  - Ask non-experts to describe typical event sequences in a given scenario

Some terminology

- Event sequence descriptions:
  - Set of ordered steps that are taking place in a specific scenario (e.g open the beamer, open my presentation, start presenting, go home)
- Temporal Script Graph
  - A directed graph $G_S = (E_s, T_s)$ where:
    - E_s is a set of nodes representing events in a scenario
    - T_s is a set of edges (e_i, e_k) indicating that e_i happens before e_k

Data Acquisition

- 22 scenarios of varying complexity
- collect 493 ESD's with Amazon Mechanical Turk (bullet type)
- Different level of granularity (average 9 event)
- 93% of all individual event decriptions occured only once

1. look at menu
2. decide what you want
3. order at counter
4. pay at counter
5. receive food at counter
6. take food to table
7. eat food

1. walk to the counter
2. place an order
3. pay the bill
4. wait for the ordered food
5. get the food
6. move to a table
7. eat food
8. exit the place

1. walk into restaurant
2. find the end of the line
3. stand in line
4. look at menu board
5. decide on food and drink
6. tell cashier your order
7. listen to cashier repeat order
8. listen for total price
9. swipe credit card in scanner
10. put up credit card
11. take receipt
12. look at order number
13. take your cup
14. stand off to the side
15. wait for number to be called
16. get your drink

Temporal Script Graphs

sequence $s_1,....,s_n$

gap cost c_{gap}

similarity cost function c_m

Compute MSA = Align events from different ESD's that describe the same action

Extract Temporal Script Graph

Temporal Script Graphs-Multiple Sequence Aligment

row	s_1	s_2	s_3	s_4
1	∅	walk into restaurant	∅	enter restaurant
2	∅	∅	walk to the counter	go to counter
3	∅	find the end of the line	∅	∅
4	∅	stand in line	∅	∅
5	look at menu	look at menu board	∅	∅
6	decide what you want	decide on food and drink	∅	make selection
7	order at counter	tell cashier your order	place an order	place order
8	∅	listen to cashier repeat order	∅	∅
9	pay at counter	∅	pay the bill	pay for food
10	∅	listen for total price	∅	∅
11	∅	swipe credit card in scanner	∅	∅
12	∅	put up credit card	∅	∅
13	∅	take receipt	∅	∅
14	∅	look at order number	∅	∅
15	∅	take your cup	∅	∅
16	∅	stand off to the side	∅	∅
17	∅	wait for number to be called	wait for the ordered food	∅
18	receive food at counter	get your drink	get the food	pick up order
19	∅	∅	∅	pick up condiments
20	take food to table	∅	move to a table	go to table
21	eat food	∅	eat food	consume food
22	∅	∅	∅	clear tray
22	∅	∅	exit the place	∅

Temporal Script Graphs- Multiple Sequence Aligment

- Compute

$$c(A) = c_{gap} \cdot \Sigma_{\oslash} + \sum_{i=1}^{n} \sum_{\substack{j=1, \\ a_{ji} \neq \oslash}}^{m} \sum_{\substack{k=j+1, \\ a_{ki} \neq \oslash}}^{m} c_m(a_{ji}, a_{ki})$$

- Choose this alignment A with the **minimum cost**
- m>2 is NP-Complete. Approximation algorithm: align two sequences, take this as one sequence of pairs and then add the third one etc.. (always m=2)
- Cost function = semantic dissimilarity
  - ESD's short, elliptic -> only pseudo-parsing possible
  - Sim = α* pred + β * subj + γ * obj
  - Pred, subj, obj scores(100 for synonyms, 0 for no relation, intermediate numbers for other kind)
  - Optimize intermediate numbers of pred, subj, obj and weights from development set

Temporal Script Graphs- Building Temporal Graph

- Initial graph-> one node for every row of MSA output (paraphrases)
- Add edges to initial graph
- Post-process
  - eliminate nodes that have only one ESD
  - Merge nodes with two filters:
    - First clustering
    - Then for preventing introduction of new temporal relations not supported by input apply graph-structural constraints.
    - Important : SHOULDN'T MERGE NODES THAT COME FROM SAME SEQUENCE!!

Experimental Setup

- Select 10 scenarios
- Baselines:
  - Clustering Baseline: (the granularity of clustering is optimized with repsect to desnity and cluster distance)
  - Levenshtein Baseline: in the proposed method replace (dis)similarity mesure with Levenshtein distance)
- 1st Task: Evaluate Paraphrases
  - Create parahrase set (30 from system and 30 random pairs)
  - Clustering Baseline: if e1, and e2 in the same cluster
- 2nd Task: Evaluate temporal constraints
  - Create happens-before set (30 from system, 30 random and the reverse of all)
  - Clustering Baseline: e comes before f if some phrase in e's cluster precedes some phrase in f's cluster
- Gold standard from Mechanical Turk (Does e1 and e2 describe the same thing in SCENARIO? Does e1 comes before e2 in SCENARIO?

Results
Paraphrasing Task

	Scenario	Precision			Recall			F-Score			
		sys	base$_{cl}$	base$_{lev}$	sys	base$_{cl}$	base$_{lev}$	sys	base$_{cl}$	base$_{lev}$	upper
MTurk	pay with credit card	0.52	0.43	0.50	0.84	0.89	0.11	**0.64**	0.58	• 0.17	0.60
	eat in restaurant	0.70	0.42	0.75	0.88	1.00	0.25	**0.78**	• 0.59	• 0.38	• 0.92
	iron clothes I	0.52	0.32	1.00	0.94	1.00	0.12	**0.67**	• 0.48	• 0.21	• 0.82
	cook scrambled eggs	0.58	0.34	0.50	0.86	0.95	0.10	**0.69**	• 0.50	• 0.16	• 0.91
	take a bus	0.65	0.42	0.40	0.87	1.00	0.09	**0.74**	• 0.59	• 0.14	• 0.88
OMICS	answer the phone	0.93	0.45	0.70	0.85	1.00	0.21	**0.89**	• 0.71	• 0.33	0.79
	buy from vending machine	0.59	0.43	0.59	0.83	1.00	0.54	**0.69**	0.60	0.57	0.80
	iron clothes II	0.57	0.30	0.33	0.94	1.00	0.22	**0.71**	• 0.46	• 0.27	0.77
	make coffee	0.50	0.27	0.56	0.94	1.00	0.31	**0.65**	• 0.42	◦ 0.40	• 0.82
	make omelette	0.75	0.54	0.67	0.92	0.96	0.23	**0.83**	• 0.69	• 0.34	0.85
	Average	0.63	0.40	0.60	0.89	0.98	0.22	**0.73**	0.56	0.30	0.82

- System outperfoms both baselines in F-score.
- Good recall for clustering but not for precision. Vice verca for Levenshtein.
- Upper bound (random selection of one of the annotations)

Results
Happens-Before Task

	Scenario	Precision			Recall			F-Score			
		sys	$base_{cl}$	$base_{lev}$	sys	$base_{cl}$	$base_{lev}$	sys	$base_{cl}$	$base_{lev}$	upper
MTurk	pay with credit card	0.86	0.49	0.65	0.84	0.74	0.45	**0.85**	• 0.59	• 0.53	0.92
	eat in restaurant	0.78	0.48	0.68	0.84	0.98	0.75	**0.81**	• 0.64	0.71	• 0.95
	iron clothes I	0.78	0.54	0.75	0.72	0.95	0.53	**0.75**	0.69	• 0.62	• 0.92
	cook scrambled eggs	0.67	0.54	0.55	0.64	0.98	0.69	0.66	*0.70*	0.61	• 0.88
	take a bus	0.80	0.49	0.68	0.80	1.00	0.37	**0.80**	• 0.66	• 0.48	• 0.96
OMICS	answer the phone	0.83	0.48	0.79	0.86	1.00	0.96	0.84	• 0.64	*0.87*	0.90
	buy from vending machine	0.84	0.51	0.69	0.85	0.90	0.75	**0.84**	• 0.66	◦ 0.71	0.83
	iron clothes II	0.78	0.48	0.75	0.80	0.96	0.66	**0.79**	• 0.64	0.70	0.84
	make coffee	0.70	0.55	0.50	0.78	1.00	0.55	**0.74**	0.71	◦ 0.53	◦ 0.83
	make omelette	0.70	0.55	0.79	0.83	0.93	0.82	0.76	◦ 0.69	*0.81*	• 0.92
	Average	0.77	0.51	0.68	0.80	0.95	0.65	**0.78**	0.66	0.66	0.90

- ANSWER THE PHONE: trivial lexicon, more word knowledge do not contribute
- COOK SRAMBLED EGGS: System can't represent events that can happen in arbitrary order. (in recipes in general)

Recap

- What we saw
  - Unsupervised method for learning events along with their order
  - Can capture everyday scenaria that are implicit knowledge and thus are not elaborated explicitly in documents
  - They way of gathering data is not automated and it is limited to the knowledge of the annotators

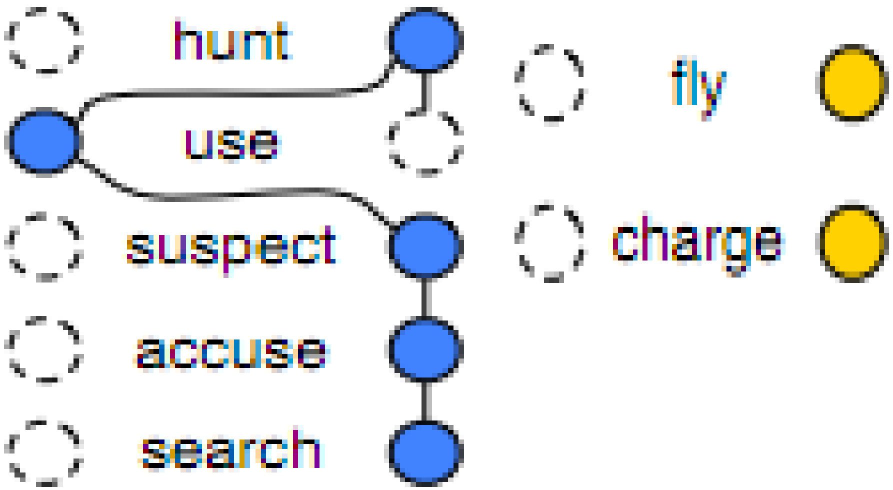
From Chambers & Jurafsky 2008

- What we saw:
  - No information about type of protagonist
  - Only one protagonist
  - Some scripts like SHOPPING are shared implicit knowledge and are rarely elaborated in text

Unsupervised Learning of Narrative Schemas and their Participants

Nathanael Chambers & Dan Jurafsky
ACL 2009

Motivation



·Top scoring is (fly, X) cause it's often seen with all 5 WRONG
➢(charge,X) shares many arguments (e.g criminal, suspect) with accuse, suspect and search

- Given *arrest* which verb is more relevant? *Convict* or *capture*?
- Only look at objects-> convict WRONG because (police *arrest* vs judge *convict*)
- Look both objects and subjets ->

  arrest and *capture* share *police* and *criminal*

- **Model argument overlap across all pairs**

- **All entities and slots in the space of events should be jointly considered**

Typed Narrative Chains

- **Narrative Chains**

  L = { (hunt, X), (X, use), (suspect, X), (accuse, X), (search,X)}

  O = {(use, hunt), (suspect, search), (suspect, accuse)...}

- **Include arguments -> Typed Narrative Chains**

  + P = {person, government, company, criminal}

Learning Argument Types

- Record counts of arguments with each pair of event slot
- Build referential set
- Represent each observed argument by the most frequent head word
- Example

> *But for a growing proportion of U.S. workers, the troubles really set in when they apply for unemployment benefits. Many workers find their benefits challenged.*

Event Slot Similarity with Arguments

- From previous approach for a new event against a chain

$$chainsim(C, \langle f, g \rangle) = \sum_{i=1}^{n} sim(\langle e_i, d_i \rangle, \langle f, g \rangle)$$

- Extend *sim* to include argument types

$$sim(\langle e, d \rangle, \langle e', d' \rangle, a) = pmi(\langle e, d \rangle, \langle e', d' \rangle) + \lambda \log freq(\langle e, d \rangle, \langle e', d' \rangle, a)$$

- Then score entire chain for a particular argument

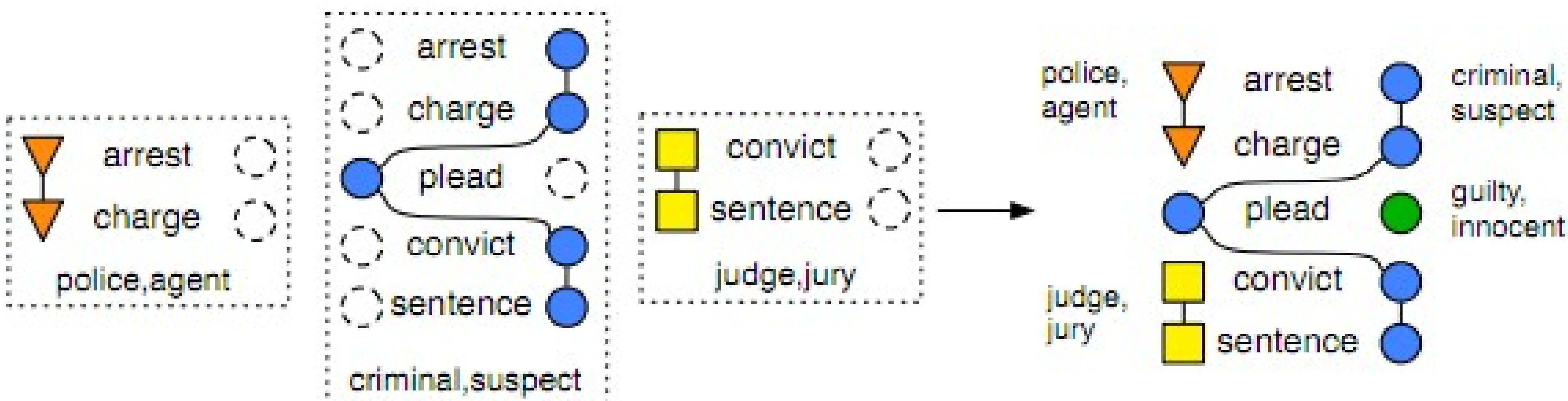
$$score(C, a) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} sim(\langle e_i, d_i \rangle, \langle e_j, d_j \rangle, a)$$

- Finally score a new event based on the argument tha maximizes the entire's chain's score

$$chainsim'(C, \langle f, g \rangle) = \max_{a}(score(C, a) + \sum_{i=1}^{n} sim(\langle e_i, d_i \rangle, \langle f, g \rangle, a))$$

Narrative Schema

- A 2-tupple N = (E,C)
  - E is set of events (v,d) where $d \in D_v$
  - C is set of events chains
  - Every event slot (v,d) belongs to a chain a chain $c \in C$



Learning Narrative Schemas

- Like the previous example with *arrest* in a chain and candidates *convict* and *capture,* we want to favor *capture* cause it shares more arguments
- Instead of asking *which event (v,d) is a best fit* ask *if v is best considering* ***all*** *slots possible*.

$$narsim(N, v) = \sum_{d \in D_v} \max(\beta, \max_{c \in C_N} chainsim'(c, \langle v, d \rangle))$$

Building Scemas

- Best candidate for Chambers & Jurafsky 2008

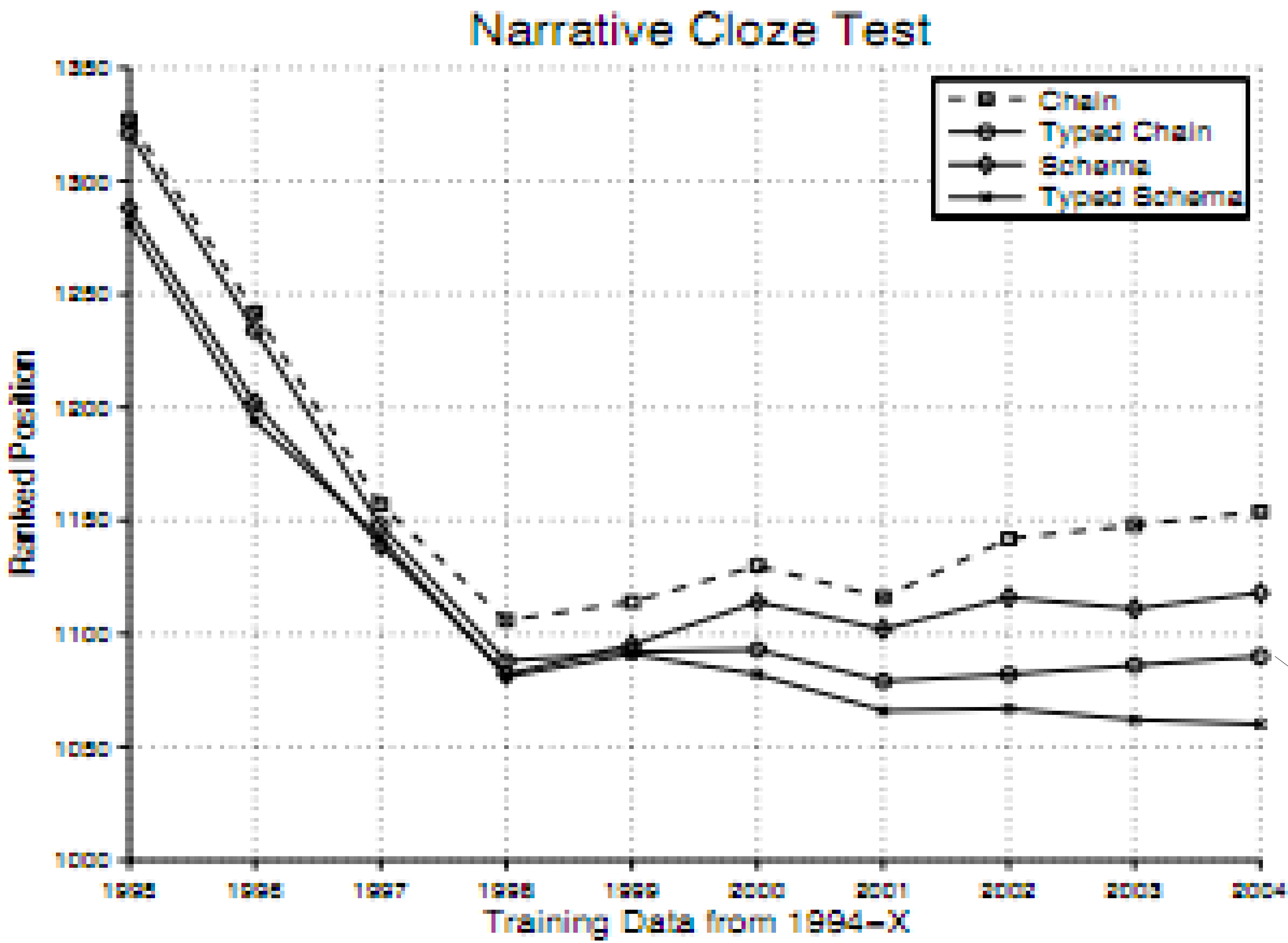
$$\max_{j:0<j<m} chainsim(c, \langle v_j, g_j \rangle)$$

- Best candidate for the proposed method

.

$$\max_{j:0<j<|v|} narsim(N, v_j)$$

Results



Chambers and Jurafsky 08

Typed chains

Conclusion

- We saw:
  - Different ways for unsupervised script learning
  - Different granularity (one or more protagonists)
  - Script learning can profit when they use for training explit scenarios
  - Different kinds of evaluation (cloze tasks, web experiments)
  - How we can leverage all the information from the document and enrich chains with arguments leading to more rich information that can help for Semantic Role Labeling.

References

Nathanael Chambers and Dan Jurafsky *Unsupervised Learning of Narrative Event Chains.* In Proceedings of ACL/HLT 2008.

Michaela Regneri, Alexander Koller and Manfred Pinkal *Learning Script Knowledge with Web Experiments.* In Proceedings of ACL 2010.

Nathanael Chambers and Dan Jurafsky *Unsupervised Learning of Narrative Schemas and their Participants.* Proceedings of ACL-IJCNLP 2009.