Master’s Thesis Defense

Matthew Jeremy Michelson

University of Southern California

June 15, 2005

Building Queryable Datasets from Ungrammatical and Unstructured Sources

Matthew Jeremy Michelson
University of Southern California
June 15, 2005

Outline

1. Introduction
2. Alignment
3. Extraction
4. Results
5. Discussion
6. Related Work
7. Conclusion

Ungrammatical & Unstructured Text

Page 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

	Topic	Replies	Last Comment	Started B
	SACRAMENTO HOTEL LIST	0	11/21/04 9:56 pm	westcoastma
	3* Rancho Cordova Holiday Inn $35, 1 nite (12/11)	1	12/9/04 12:37 am	future canadi
	3* Doubletree Sacto Arden 12/11 1 Night $34	1	12/7/04 4:46 pm	OCTraveler
	4* Sacramento Failed Bid $85 12/7	1	12/6/04 6:29 pm	Sheryl
	Failed bid Sacramento Downtown 12/6 for 1 night, 4*	13	12/6/04 6:25 pm	emaij
	2.5* Wingate Inn Rancho Cordova 5/10-5/13/05 $32	0	12/4/04 7:11 pm	ego68
	3* DoubleTree Sacramento $35 (12/04/04)	0	11/30/04 11:34 pm	shizzolator
	2.5* Rancho Cordova Wingate Inn $32 (11/23-25)	1	11/27/04 12:19 pm	Profiler
	4* DT Hyatt 11/21 $60 11/23 $60; Sheraton Grand 11/25 $55	0	11/22/04 1:22 pm	bonish
	3* Doubletree Arden/Sacramento $37 11/19	1	11/20/04 1:53 am	ahallez
	2.5* Wingate Inn Rancho Cordova $33 11/13	2	11/19/04 1:44 am	cykick42
	2.5* DT Hawthorne Suites $40 (11/18-20)	0	11/18/04 10:08 pm	Colfax30
	Roseville 2.5*Larkspur $72(11/22-24) 2* Fairfield $80(11/24)	2	11/17/04 4:38 pm	mcrinca
	3* Rancho Cordova Holiday Inn $32 (11/17)	0	11/16/04 10:20 pm	Colfax30
	3* Doubletree Sacramento $40 (11/11)	2	11/16/04 11:05 am	OCTraveler
	3* Doubletree Sacramento Arden $36 11/24	0	11/15/04 1:04 am	bomawin

Ungrammatical & Unstructured Text

For simplicity → “posts”

Goal:

Beware 2* at the airport!!!!	2	7/18/00 1:25 am
$25 winning bid at holiday inn sel. univ. ctr.	1	6/26/00 1:48 pm
3* Holiday Inn North-McKnight Rd, $10+20, 1/19	3	1/27/01 6:34 pm

No wrapper based IE (e.g. Stalker [1], RoadRunner [2])

No NLP based IE (e.g. Rapier [3], Whisk [4])

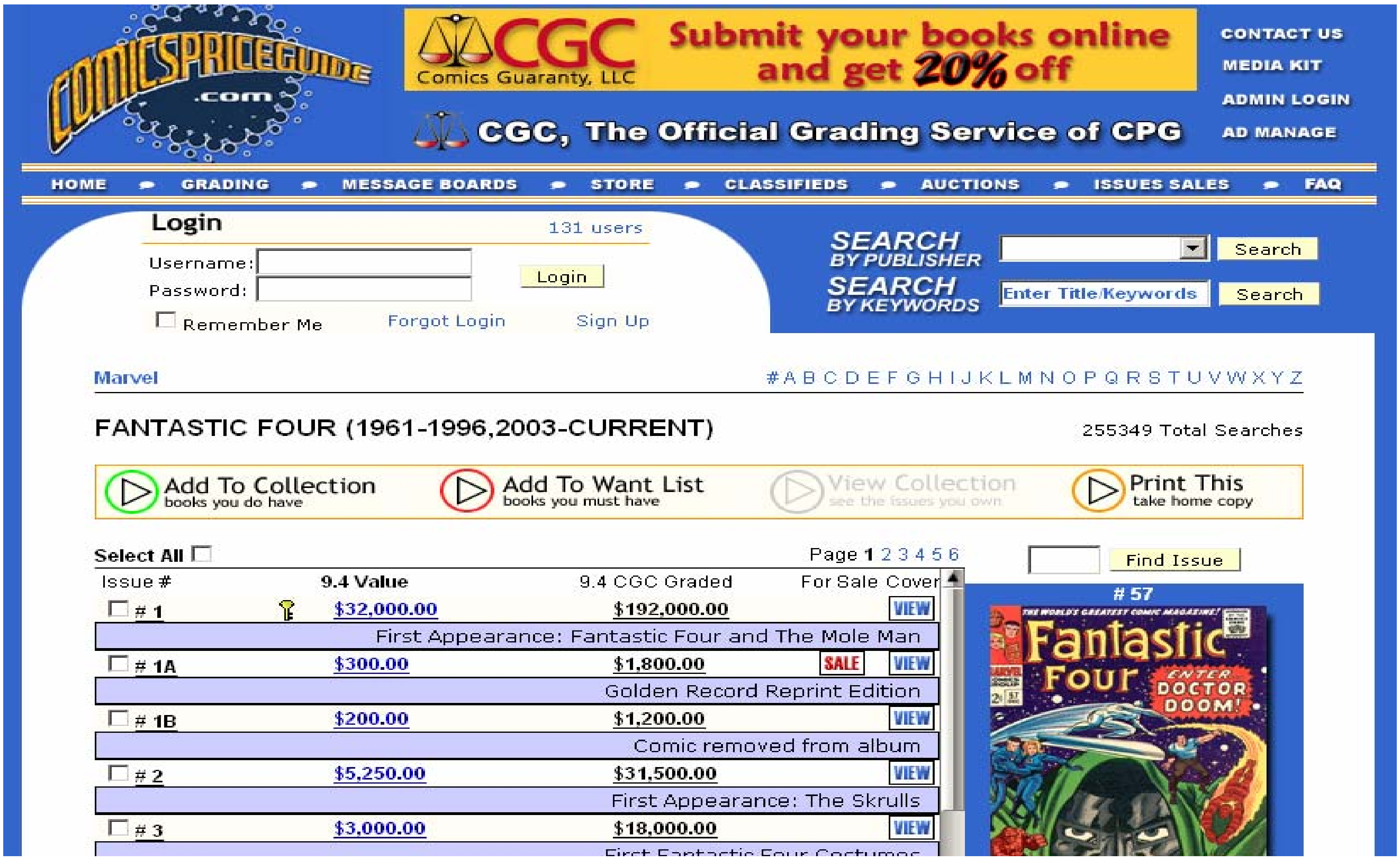
Reference Sets

IE infused with outside knowledge

"Reference Sets"

- Collections of known entities and the associated attributes
- Online (offline) set of docs
  - CIA World Fact Book
- Online (offline) database
  - Comics Price Guide, Edmunds, etc.
- Build from ontologies on Semantic Web

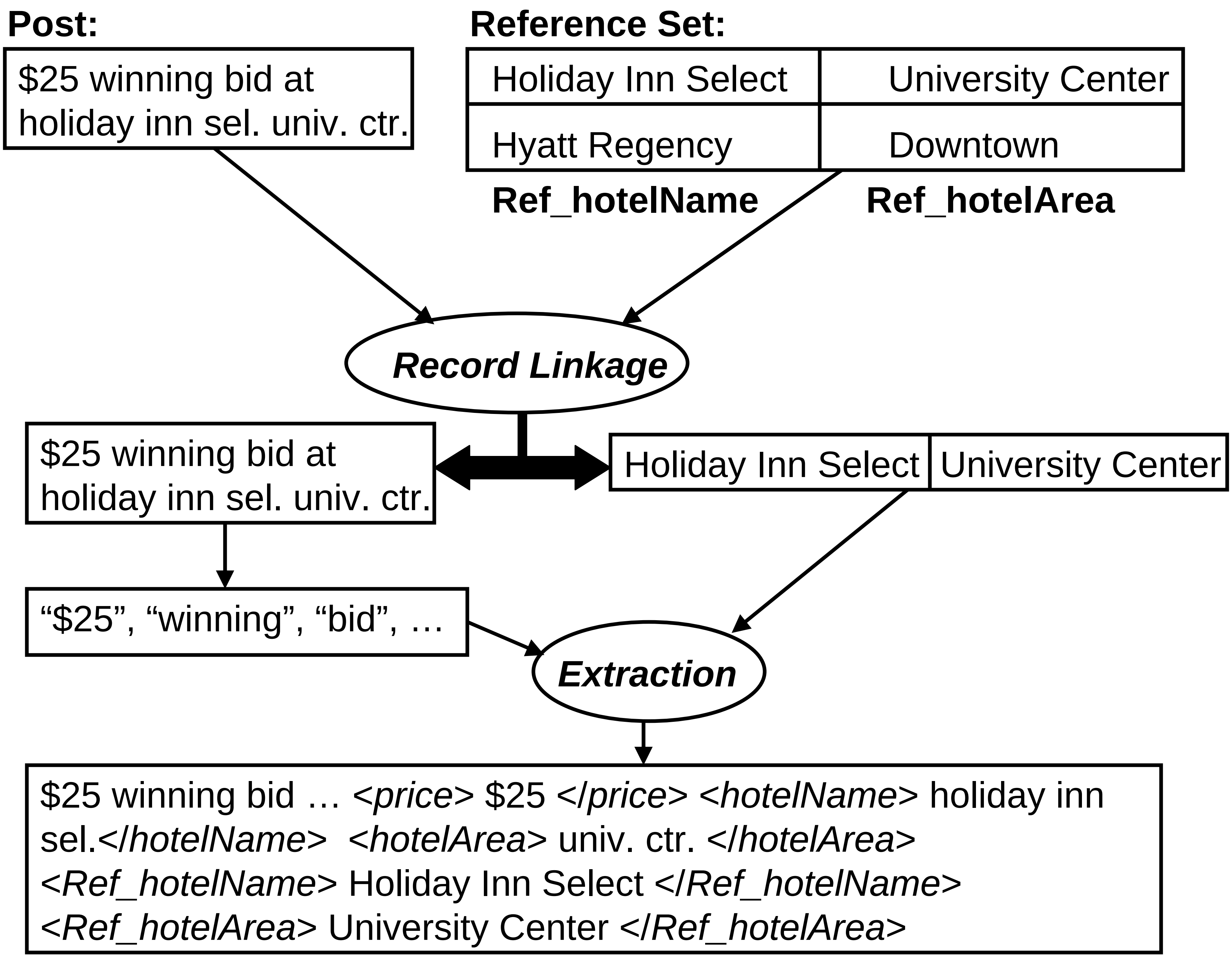
Comics Price Guide Reference Set



Use of Reference Sets

Intuition

- Align post to a member of the reference set
- Exploit the reference set member's attributes for extraction


Post:
$25 winning bid at holiday inn sel. univ. ctr.
Reference Set:
Holiday Inn Select
University Center
Hyatt Regency
Downtown
Ref_hotelName
Ref_hotelArea
Record Linkage
$25 winning bid at holiday inn sel. univ. ctr.
Holiday Inn Select
University Center
“$25”, “winning”, “bid”, …
Extraction
$25 winning bid … <price> $25 </price> <hotelName> holiday inn sel.</hotelName> <hotelArea> univ. ctr. </hotelArea>
<Ref_hotelName> Holiday Inn Select </Ref_hotelName>
<Ref_hotelArea> University Center </Ref_hotelArea>

Outline

1. Introduction
2. Alignment
3. Extraction
4. Results
5. Discussion
6. Related Work
7. Conclusion

Traditional Record Linkage

Match on decomposed attributes.

Field similarities → record level similarity

Post:

hotel name	hotel area
holiday inn sel.	univ. ctr.

Reference Set:

hotel name	hotel area
Holiday Inn	Greentree
Holiday Inn Select	University Center
Hyatt Regency	Downtown

Our Record Linkage Problem

Posts not yet decomposed attributes.

Extra tokens that match nothing in Ref Set.

Post:

$25 winning bid at | holiday inn sel. | univ. ctr.

hotel name *hotel area*

Reference Set:

Holiday Inn	Greentree
Holiday Inn Select	University Center
Hyatt Regency	Downtown
hotel name	*hotel area*

Our Record Linkage Problem

Our technique:

V_{RL} : Vector to represent similarities between data sets

RL_scores **:** Vector of similarities between strings

V_{RL} is composed of multiple ***RL_scores***

$$V_{RL} = \langle RL\_scores(s,t), RL\_scores(a,b), \ldots \rangle$$

But what exactly defines ***RL_scores*** ?

RL_scores

RL_scores(s, t)

< token_scores(s, t), edit_scores(s, t), other_scores(s, t) >

Jensen-Shannon
(Dirichlet & Jelenik-Mercer)

Jaccard

Levenstein

Smith-Waterman

Jaro-Winkler

Soundex
Porter Stemmer

Our Record Linkage Problem

Record Level Similarity (RLS):

RL_scores between ***post*** and all ***reference set attributes concatenated*** together

P = $25 winning bid at holiday inn sel. univ. ctr.

Reference Set:

Hyatt Regency	Downtown

R = Hyatt Regency Downtown

RLS = RL_scores(P, R)

Record Level Similarity Issue…

Post:

1*	Bargain Hotel	Downtown

Cheap!

star *hotel name* *hotel area*

Reference Set:

2*	Bargain Hotel	Downtown
1*	Bargain Hotel	Paradise

star *hotel name* *hotel area*

What if equal ***RLS*** but different attributes? Many more hotels share **Star** than share **Hotel Area** → need to reflect **Hotel Area** similarity more discriminative…

Field Level Similarity

Field Level Similarity → ***RL_scores*** between the ***post*** and ***each attribute*** of the reference set

Reference Set:

Hyatt Regency	Downtown

RL_scores(***P***, ***"Hyatt Regency"***)

RL_scores(***P***, ***"Downtown"***)

Full Similarity – capture both!

V_{RL} = ***Record Level Similarity + Field Level Similarities***

V_{RL} = < ***RL_scores***(***P***, ***"Hyatt Regency Downtown"***),
RL_scores(***P***, ***"Hyatt Regency"***),
RL_scores(***P***, ***"Downtown"***)>

Binary Rescoring

Candidates = < V_{RL1} , V_{RL2} , … , V_{RLn} >

V_{RL}(s) with max value at index i set that value to 1. All others set to 0.

V_{RL1} = < *0.999, 1.2, …, 0.45, 0.22* >

V_{RL2} = < *0.888, 0.0, …, 0.65, 0.22* >

$\Downarrow$

V_{RL1} = < *1, 1, …, 0, 1* >

V_{RL2} = < *0, 0, …, 1, 1* >

Emphasize best match → similarly close values but only one is best match

SVM Classification

$V_{RL1} = < 1, 1, \ldots, 0, 1 >$

$V_{RL2} = < 0, 0, \ldots, 1, 1 >$

SVM

Best matching member of the reference set for the post

SVM Classification

SVM

- Trained to classify matches/ non-matches
- Returns score from decision function
- Best Match: Candidate that is a match & max. score from decision function
  - 1-1 mapping: If more than one cand. with max. score → throw them all away
  - 1-N mapping: If more than one cand. with max. score → keep first/ keep random of set with max.

Last Alignment Step

Return reference set attributes as annotation for the post

Post:

$25 winning bid at holiday inn sel. univ. ctr.

<Ref_hotelName>Holiday Inn Select</Ref_hotelName>

<Ref_hotelArea>University Center</Ref_hotelArea>

… more to come in Discussion…

Outline

1. Introduction
2. Alignment
3. Extraction
4. Results
5. Discussion
6. Related Work
7. Conclusion

Extraction with Reference Sets

- Exploit matching reference set member
  - Use values as clues for what to extract
  - Use schema for annotation tags

Extraction with Reference Sets

- First, break posts into tokens

  $25 winning bid at holiday inn sel. univ. ctr.

  ⬇

  < ***"$25", "winning", "bid", …*** >

- Next, build vector of similarity scores for token
  - Sims. between token and ref. set attributes
  - Can classify token based on scores

Extraction with Reference Sets

- V_{IE} : Vector of similarities between token and ref. set attributes.
- ***IE_scores*** **:** Vector of similarities between strings
- V_{IE} similar V_{RL}
  - Composed of ***IE_scores*** similar ***RL_scores***

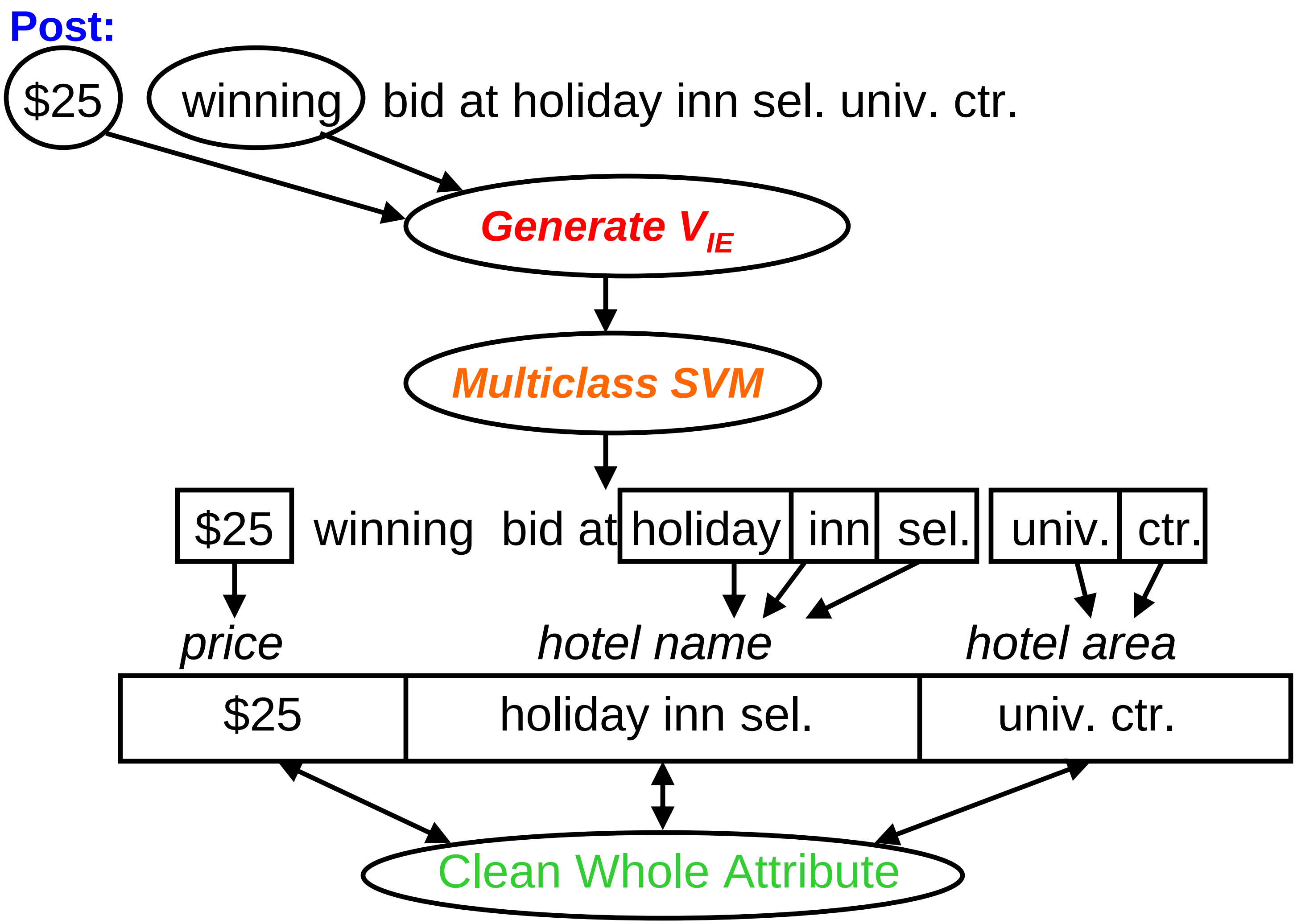
Differences

- Difference between ***IE_scores*** and ***RL_scores***
  - No ***token_scores*** in ***IE_scores***
    - consider 1 token at a time from the post
  - ***IE_scores = <edit_scores, other_scores>***
- Difference between V_{IE} and V_{RL}
  - V_{IE} contains vector ***common_scores***
  - V_{IE} = < ***common_scores(token), IE_scores(token, attr1), IE_scores(token, attr2), …*** >

Common Scores

- Some attributes not in reference set
  - Reliable characteristics
  - Infeasible to represent in reference set
  - E.g. prices, dates
- Can use characteristics to extract/annotate these attributes
  - Regular expressions, for example
- These types of scores are what compose ***common_scores***

Extraction Algorithm



Cleaning an attribute

- Labeling tokens in isolation leads to noise
  - Can use ref. set. attribute vs. whole extracted attribute

- Overview of cleaning algorithm
  1. Uses Jaccard (token) and Jaro-Winkler (edit)
  2. Generate baseline similarities between extracted attribute and the reference set analogue
  3. Then, try removing one token at a time from extracted
     a) If similarities greater than baseline → candidate for removal
     b) After all tokens processed this way, remove candidate with highest scores
     c) Update baseline scores to new high scores
  4. Repeat (3) until no tokens can beat baseline

Baseline scores: *holiday inn sel. in*

Jaro-Winkler (edit): 0.87 Jaccard (token): 0.4

Iteration 1

Scores: *holiday inn sel. ~~in~~*

Jaro-Winkler (edit): 0.92 (> 0.87) Jaccard (token): 0.5 (> 0.4)

New baselines

New Hotel Name: *holiday inn sel.*

Iteration 2

Scores: *holiday inn ~~sel.~~*

Jaro-Winkler (edit): 0.84 (< 0.92) Jaccard (token): 0.25 (< 0.5)

Scores: *holiday ~~inn~~ sel.*

Jaro-Winkler (edit): 0.87 (< 0.92) Jaccard (token): 0.66 (> 0.5)

…

No improvement → terminate

holiday inn sel.

Annotation

Beware 2* at the airport!!!!	2	7/18/00 1:25 am
$25 winning bid at holiday inn sel. univ. ctr.	1	6/26/00 1:48 pm
3* Holiday Inn North-McKnight Rd, $10+20, 1/19	3	1/27/01 6:34 pm

<*price*> $25 </*price*>

<*hotelName*> holiday inn sel. </*hotelName*>

<*Ref_hotelName*> Holiday Inn Select </*Ref_hotelName*>

<*hotelArea*> univ. ctr. </*hotelArea*>

<*Ref_hotelArea*> University Center </*Ref_hotelArea*>

Outline

1. Introduction
2. Alignment
3. Extraction
4. Results
5. Discussion
6. Related Work
7. Conclusion

Experimental Data Sets

Hotels

- ***Posts***
  - 1125 posts from www.biddingfortravel.com
    - Pittsburgh, Sacramento, San Diego
    - Star rating, hotel area, hotel name, price, date booked
- ***Reference Set***
  - 132 records
  - Special posts on BFT site.
    - Per area – list any hotels ever bid on in that area
    - Star rating, hotel area, hotel name

Experimental Data Sets

Comics

- ***Posts***
  - 776 posts from EBay
    - "Incredible Hulk" and "Fantastic Four" in comics
    - Title, issue number, price, condition, publisher, publication year, description (1st appearance the Rhino)
- ***Reference Sets***
  - 918 comics, 49 condition ratings
  - Both come from ComicsPriceGuide.com
    - For FF and IH
    - Title, issue number, description, publisher

Experimental Data Sets

Cars

- ***Posts***
  - 855 posts from Craig's list (cars section)
    - 1st 10 pages from LA, NYC and SF sites
    - Remove those that have car not in ref set. (But not if no car or mult. cars w/ at least 1 in ref set)
    - Make, model, trim, year, price
- ***Reference Set***
  - 3171 records
  - Edmunds website - courtesy of Fetch Technologies Inc.
    - Japanese cars and SUVs from 1990-2003
    - Make, model, trim, year

Comparisons

Record Linkage

- WHIRL [5]

Information Extraction

- Simple Tagger (CRF) [6]
- Amilcare [7]

Record linkage results

	Prec.	Recall	F-measure
Hotel			
Phoebus	93.60	91.79	**92.68**
WHIRL	83.52	83.61	83.13
Comic			
Phoebus	93.24	84.48	**88.64**
WHIRL	73.89	81.63	77.57
Cars			
Phoebus	93.15	99.57	**96.53**
WHIRL	75.18	40.46	51.86

10 trials – 30% train, 70% test

Extraction results (token): Hotel domain

Hotel					
		Prec.	Recall	F-Measure	Freq
Area	Phoebus	89.25	87.5	**88.28**	809.7
	Simple Tagger	92.28	81.24	86.39	
	Amilcare	74.20	78.16	76.04	
Date	Phoebus	87.45	90.62	**88.99**	751.9
	Simple Tagger	70.23	81.58	75.47	
	Amilcare	93.27	81.74	86.94	
Name	Phoebus	94.23	91.85	93.02	1873.9
	Simple Tagger	93.28	93.82	**93.54**	
	Amilcare	83.61	90.49	86.90	
Price	Phoebus	98.68	92.58	**95.53**	850.1
	Simple Tagger	75.93	85.93	80.61	
	Amilcare	89.66	82.68	85.86	
Star	Phoebus	97.94	96.61	**97.84**	766.4
	Simple Tagger	97.16	97.52	97.34	
	Amilcare	96.50	92.26	94.27	

Not Significant

Extraction results (token): Comic domain

		Prec.	Recall	F-Measure	Freq
Condition	Phoebus	91.80	84.56	**88.01**	410.3
	Simple Tagger	78.11	77.76	77.80	
	Amilcare	79.18	67.74	72.80	
Descript.	Phoebus	69.21	51.50	59.00	504.0
	Simple Tagger	62.25	79.85	**69.86**	
	Amilcare	55.14	58.46	56.39	
Issue	Phoebus	93.73	86.18	**89.79**	669.9
	Simple Tagger	86.97	85.99	86.43	
	Amilcare	88.58	77.68	82.67	
Price	Phoebus	80.00	60.27	**68.46**	10.7
	Simple Tagger	84.44	44.24	55.77	
	Amilcare	60.0	34.75	43.54	
Publisher	Phoebus	83.81	95.08	**89.07**	61.1
	Simple Tagger	88.54	78.31	82.83	
	Amilcare	90.82	70.48	79.73	
Title	Phoebus	97.06	89.90	93.34	1191.1
	Simple Tagger	97.54	96.63	**97.07**	
	Amilcare	96.32	93.77	94.98	
Year	Phoebus	98.81	77.60	**84.92**	120.9
	Simple Tagger	87.07	51.05	64.24	
	Amilcare	86.82	72.47	78.79	

Extraction results (token): Cars domain

Cars					
		Prec.	Recall	F-Measure	Freq
Make	Phoebus	99.96	97.53	**98.73**	459.4
	Simple Tagger	95.66	86.01	90.56	
	Amilcare	92.34	96.82	94.51	
Model	Phoebus	98.35	94.70	**96.49**	514.2
	Simple Tagger	94.25	79.57	86.28	
	Amilcare	83.71	76.18	79.73	
Trim	Phoebus	91.85	73.36	**81.54**	482.6
	Simple Tagger	84.31	66.68	74.25	
	Amilcare	66.98	58.47	62.33	
Year	Phoebus	97.68	92.10	**94.79**	474.1
	Simple Tagger	79.91	91.47	85.27	
	Amilcare	92.73	85.96	89.18	
Price	Phoebus	97.24	97.12	**97.18**	489.4
	Simple Tagger	98.19	83.91	90.49	
	Amilcare	90.90	91.11	90.93	

Extraction results: Summary

	Hotel						
	Token level				Field level		
	Prec.	Recall	F-Mes.		Prec.	Recall	F-Mes.
Phoebus	93.60	91.79	**92.68**		87.44	85.59	**86.51**
Simple Tagger	86.49	89.13	87.79		79.19	77.23	78.20
Amilcare	86.12	86.14	86.11		85.04	78.94	81.88
	Comic						
	Token level				Field level		
	Prec.	Recall	F-Mes.		Prec.	Recall	F-Mes.
Phoebus	93.24	84.48	**88.64**		81.73	80.84	**81.28**
Simple Tagger	84.41	86.04	85.43		78.05	74.02	75.98
Amilcare	87.66	81.22	84.29		90.40	72.56	80.50
	Cars						
	Token level				Field level		
	Prec.	Recall	F-Mes.		Prec.	Recall	F-Mes.
Phoebus	97.20	92.22	**94.65**		92.67	90.63	**91.64**
Simple Tagger	89.80	81.49	85.44		86.49	80.79	83.54
Amilcare	85.73	81.53	83.58		87.02	79.28	82.92

Results

3 attributes where Phoebus not max F-measure

- Hotel name – tiny difference
- Comic Title – low recall → lower F-measure
  - recall: missed tokens of titles not in ref. set
  - "The Incredible Hulk and Wolverine" → "The Incredible Hulk"
- Comic description
  - ST learned internal structure of descs (label too many)
    - High recall, low precision
  - Phoebus labels in isolation
    - Only meaningful tokens (like prop. Names) labeled
    - higher precision, lower recall → 2nd best F-measure

Outline

1. Introduction
2. Alignment
3. Extraction
4. Results
5. Discussion
6. Related Work
7. Conclusion

Extraction results (token) summary

Cost of labeling data is expensive…

	Prec.	Recall	F-measure
Hotel (30%)	93.60	91.79	92.68
Hotel (10%)	93.66	90.93	92.27
Comic (30%)	93.24	84.48	88.64
Comic (10%)	91.41	83.63	87.34
Cars (30%)	97.20	92.22	94.65
Cars (10%)	96.51	91.82	94.11

Reference Set Attributes as Annotation

- Standard query values
- Include info not in post
  - If post leaves out “Star Rating” can still be returned in query on “Star Rating” using ref. set annotation
- Perform better at annotation than extraction
  - Consider Rec. link results as field level extraction
  - E.g. no system did well extracting comic desc.
    - +20% precision, +10% recall using rec. link

Reference Set Attributes as Annotation

Then why do extraction at all?

- Want to see actual values
- Extraction can annotate when record linkage is wrong
  - Better in some cases at annotation than rec. link
  - If wrong rec. link, usually close enough record to get some extraction parts right
- Learn what something is not
  - Helps to classify things not in reference set
  - Learn which tokens to ignore better

Outline

1. Introduction
2. Alignment
3. Extraction
4. Results
5. Discussion
6. Related Work
7. Conclusion

Related Work

- Generate mark-up for Semantic Web
  - Rely on lexical info [8,9,10,11] or structure [12]
- Record Linkage
  - Require decomposed attributes
  - WHIRL is exception, used in experiments
- Data Cleaning
  - Tuple-to-tuple transformations [13,14]
- Info. Extraction (for Annotation)
  - Conditional Random Fields (Simple Tagger)
  - Datamold / CRAM [15,16]
    - Require all tokens to receive label / no junk
  - NER with Dictionary [17]
    - Whole segments receive same label – attributes can’t be interrupted

Outline

1. Introduction
2. Alignment
3. Extraction
4. Results
5. Discussion
6. Related Work
7. Conclusion

Conclusion

- Annotate unstructured and ungrammatical sources
  - Don’t involve users
  - Structured queries over data sources
- Future:
  - Automate entire process
    - Unsupervised RL and IE
    - Mediator gets Reference Sets

References

1. Ion Muslea, Steven Minton, and Craig A. Knoblock. Hierarchical wrapper induction for semistructured information sources. *Autonomous Agents and Multi-Agent Systems*, 4(1/2):93–114, 2001.
2. Valter Crescenzi, Giansalvatore Mecca, and Paolo Merialdo. Roadrunner: Towards automatic data extraction from large web sites. In *Proceedings of 27th International Conference on Very Large Data Bases*, pages 109–118, 2001.
3. Mary Elaine Califf and Raymond J. Mooney. Relational learning of pattern-match rules for information extraction. In *Proceedings of the 16th National Conference on Artificial Intelligence and 11th Conference on Innovative Applications of ArtificialIntelligence*, pages 328–334, Orlando, Florida, August 1999.
4. Stephen Soderland. Learning information extraction rules for semi-structured and free text. *Machine Learning*, 34(1-3):233–272, 1999.

References (2)

5. William W. Cohen. Data integration using similarity joins and a word-based information representation language. *ACM Transactions on Information Systems*, 18(3):288–321, 2000.
6. Andrew McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.
7. Fabio Ciravegna. Adaptive information extraction from text by rule induction and generalisation. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, 2001.
8. Maria Vargas-Vera, Enrico Motta, John Domingue, Mattia Lanzoni, Arthur Stutt, and Fabio Ciravegna. Mnm: Ontology driven semi-automatic and automatic support for semantic markup. In *Proceedings of the 13th International Conference on Knowledge Engineering and Management*, 2002.

References (3)

9. Siegfried Handschuh, Steffen Staab, and Fabio Ciravegna. S-cream - semi-automatic creation of metadata. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management*. Springer Verlag, 2002.
10. Philipp Cimiano, Siegfried Handschuh, and Steffen Staab. Towards the self-annotating web. In *Proceedings of the 13th international conference on World Wide Web*, pages 462–471. ACM Press, 2004.
11. Alexiei Dingli, Fabio Ciravegna, and Yorick Wilks. Automatic semantic annotation using unsupervised information extraction and integration. In *Proceedings of the Workshop on Knowledge Markup and Semantic Annotation*, 2003.
12. Kristina Lerman, Cenk Gazen, Steven Minton, and Craig A. Knoblock. Populating the semantic web. In *Proceedings of the Workshop on Advances in Text Extraction and Mining*, 2004.

References (4)

13. Mong-Li Lee, Tok Wang Ling, Hongjun Lu, and Yee Teng Ko. Cleansing data for mining and warehousing. In *Proceedings of the 10th International Conference on Database and Expert Systems Applications*, pages 751–760. Springer-Verlag, 1999.
14. Surajit Chaudhuri, Kris Ganjam, Venkatesh Ganti, and Rajeev Motwani. Robust and efficient fuzzy match for online data cleaning. In *Proceedings of ACM SIGMOD*, pages 313–324. ACM Press, 2003.
15. Vinayak Borkar, Kaustubh Deshmukh, and Sunita Sarawagi. Automatic segmentation of text into structured records. In Proceedings of ACM SIGMOD, 2001.
16. Eugene Agichtein and Venkatesh Ganti. Mining reference tables for automatic text segmentation. In the *Proceedings of the 10th ACM Int'l Conf. on Knowledge Discovery and Data Mining, Seattle*, Washington, August 2004. ACM Press.

References (5)

17. William Cohen and Sunita Sarawagi. Exploiting dictionaries in named entity extraction: combining semi-markov extraction processes and data integration methods. In *Proceedings of the 10th ACM Int'l Conf. Knowledge Discovery and Data Mining*, Seattle, Washington, August 2004. ACM Press.