

# **Multi-Document Statistical Fact Extraction and Fusion**

by

Gideon S. Mann

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

April, 2006

© Gideon S. Mann 2006

All rights reserved

# Abstract

This dissertation presents original techniques for statistical fact extraction and fusion from multiple documents. **Fact extraction**, or relationship extraction, is a process where natural language text is scanned to find instances of a predetermined class of facts (e.g.  $\text{birthday}(x,y)$ ). A framework for training statistical fact extractors from example is used wherein a set of examples and a target model are used to annotate an automatically collected corpus. This annotation is then used to provide training data for classifiers (Phrase Conditional Likelihood and Naïve Bayes) or sequence models (Conditional Random Fields).

Fact extractors are used in two information retrieval tasks. In **question answering** the set of candidate answers is narrowed using fine-grained proper noun ontological facts ( $\text{is-a}(X,Y)$ ) extracted from a corpus by rote classifiers leading to higher performance. Extracted facts are also used for **name-referent disambiguation**, or cross-document coreference, where one personal name may refer to multiple potential people in the world. The distinguishing biographic facts for each person, such as  $\text{birthday}(x,y)$  and  $\text{occupation}(x,y)$ , are automatically extracted from plain text and these biographic facts are used along with other statistical methods to distinguish between mentions of each of the referents.

This dissertation presents novel techniques for fusion which integrate facts extracted from multiple sources. For the task of **biographic fact extraction**, fusion of factual information extracted from multiple documents improves the precision of the resulting information. Further improvements result from cascaded fact extraction, where certain facts are extracted and fused and then these facts are used to extract additional information. The technique of cascaded fact extraction and fusion is also

applied to time-bounded facts, where a cascade of fact extractors produce a timeline of **corporate management succession**.

Collectively, this research demonstrates the utility of multi-document fact extraction and fusion. It shows that facts can serve as a building-block for deeper text processing such as finding coreferent names in a series of documents, finding the answers to questions, and constructing a timeline for time-variable facts. The key aspects to the process are training with minimal supervision, high-performance statistical fact extraction, fusion across multiple sources of information, and cascaded extraction.

Adviser: David Yarowsky

Second Reader: Jason Eisner

# Acknowledgments

This section would easily dwarf all the others, if I listed, even in brief, all of the help, assistance, support, and encouragement I received from my colleagues, friends and family during this process. Any accounting must therefore be incomplete.

I want to thank first, my parents, Belle Sheppard Mann and Jason Mann, who helped me to think critically, creatively and independently.

I'd like to thank my adviser, David Yarowsky, who gave me space to explore and guidance along the way, and my committee members, Jason Eisner, and Ellen Riloff, who helped refine and better define my work.

During my time here, I'd had the pleasure of being surrounded with lots of wonderful friends and co-conspirators while at the NLP Lab: Richard Wicentowski, Hans Florian, Grace Ngai, Silviu Cucerzan, Jun Wu, Charles Schafer, Michele Banko, Elliott Drábek, Noah Smith, Roy Tromble, David Smith, Markus Dreyer, Brock Pytlik, John Blatz, and Nikesh Garera. and Paul Ruhlen. I'd like to especially thank John Blatz for hand delivering this dissertation to the library. Outside of the NLP lab, I was lucky enough to spend time with Mike Shin, Sean Hundtofte, Geetu Ambwani, and Ameet Jain. I've also benefited from the friendship and exchange of ideas from the NLP community, including Michael

Fleischman, Deepak Ravichandran, and Alexander Koller.

I want to thank my grandmother, Marlene Mann, for all of her love. Though not here in body, she is always with me in spirit. My sister, Elana Mann, and my grandparents, Selma and Leonard Pepkowitz, have been encouraging and supportive during my whole time here. In Baltimore, I had the incredible good fortune to spend time with my extended family Becky Pepkowitz, Gerry Gilstrop, and Chad Tabor, who gave me encouragement and support at crucial moments, and sustained my spirit. Other friends from Baltimore have also been enormously important to my survival and enjoyment of these past few years, among them: John Berndt, Howard Reznick, Mike O'Malley, Lisa Welchman, and Sosha Devi.

And last, but certainly not least, Patricia Driscoll. Without you, I don't know if I could have finished. Certainly, it wouldn't have meant as much.

# Contents

<b>Abstract</b>	<b>ii</b>
<b>List of Figures</b>	<b>ix</b>
<b>List of Tables</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Fact Extraction . . . . .	2
1.2 Question Answering and Cross-Document Coreference . . . . .	4
1.3 Multi-Field, Multi-Document Extraction of Biographical Facts . . . . .	6
1.4 Time-Bounded Facts and Timeline Construction . . . . .	7
1.5 Web Corpora . . . . .	8
<b>2 Training Fact Extractors by Example Facts</b>	<b>10</b>
2.1 The History of Information Extraction . . . . .	11
2.2 System Overview . . . . .	13
2.2.1 Training . . . . .	13
2.2.2 Extraction . . . . .	15
2.3 Positive and Negative Examples . . . . .	16
2.4 Sentence Extraction Methods . . . . .	17
2.4.1 Phrase Conditional Models . . . . .	17
2.4.2 Naïve Bayes Classifiers . . . . .	19
2.4.3 Conditional Random Field Models . . . . .	21
2.4.4 Knowledge Engineering . . . . .	26
2.4.5 Alternate Classifiers . . . . .	27
2.5 Limitations to Example-Based Training . . . . .	27
2.5.1 Alternative Methods for Training via Minimal Supervision . . . . .	28
2.6 Conclusion . . . . .	29
<b>3 Ontology Induction for Question Answering</b>	<b>30</b>
3.1 Introduction . . . . .	30
3.2 Ontologies for Question Answering . . . . .	32

3.3	Building a Proper Noun Ontology . . . . .	35
3.4	Using a Proper Noun Ontology in a Question Answering Task . . . . .	39
3.5	Related work . . . . .	42
3.6	Conclusion . . . . .	45
<b>4</b>	<b>Multi-Document and Multi-field Fact Extraction and Fusion</b>	<b>46</b>
4.1	Biographic Facts . . . . .	47
4.2	Fact Extraction Model Structure and Feature Set . . . . .	50
4.2.1	Feature Set . . . . .	53
4.2.2	Target Set Models . . . . .	53
4.3	Biographic Fact Extraction . . . . .	54
4.4	CRF Viterbi Sequence Extraction and Label Marginal Estimation . . . . .	62
4.5	Multi-Document Biographic Fact Extraction and Fusion . . . . .	63
4.5.1	Comparison of Fusion Methods . . . . .	66
4.5.2	Training-Data Annotation Variants . . . . .	75
4.5.3	Order-Invariant Models . . . . .	77
4.5.4	Familial Relationship Extraction and Fusion . . . . .	80
4.5.5	Test Set Size and Performance . . . . .	82
4.5.6	Training Set Size and Performance . . . . .	83
4.6	Bootstrapping . . . . .	87
4.6.1	Experimental Results . . . . .	90
4.7	Coreference : all gender-matching pronouns . . . . .	92
4.8	Cascaded Fact Extraction and Fusion for Multi-Relationship Extraction . . . . .	96
4.9	Related Work . . . . .	100
4.10	Conclusion . . . . .	103
<b>5</b>	<b>Fact Extraction for Cross-Document Coreference</b>	<b>105</b>
5.1	Name Polyreference . . . . .	106
5.2	User Scenarios . . . . .	107
5.3	Cross-Document Coreference Method . . . . .	108
5.3.1	Applications of Facts for Cross-Document Coreference . . . . .	111
5.4	B-Cubed for Cluster Evaluation . . . . .	113
5.5	Web Pseudoname Experiments . . . . .	114
5.5.1	Unsupervised Clustering . . . . .	116
5.5.2	Seed Clusters from Fact Extraction . . . . .	119
5.5.3	Seed Clusters from Oracle Facts . . . . .	122
5.5.4	Representative Page Initialization . . . . .	124
5.6	Web Polyreference Experiments . . . . .	125
5.6.1	Web Polyreference Data Set . . . . .	129
5.6.2	Unsupervised Clustering . . . . .	132
5.6.3	Seed Clusters from Fact Extraction . . . . .	135
5.7	“John Smith” Experiments . . . . .	137
5.7.1	Fact Extraction for “John Smith” . . . . .	141
5.8	Genre Effects on Cross-Document Coreference Resolution . . . . .	141
5.9	Related Work . . . . .	143

5.10	Conclusion	145
<b>6</b>	<b>Time-Bounded Facts and Timeline Construction</b>	<b>146</b>
6.1	Time-Bounded Facts in Text	147
6.1.1	Extraction Methods for Temporal Information	148
6.1.2	Reasoning about Temporal Events with Uncertain Information	150
6.2	Management Succession	151
6.3	Corporate Entities	154
6.4	Cascaded Extraction and Fusion for Timeline Creation	155
6.5	CEO Extraction	156
6.5.1	Development Set Performance	160
6.5.2	Biographic Fact Extraction	164
6.6	Extracting Temporal Information	166
6.6.1	Co-sentential Year Baseline	166
6.6.2	Succession Facts for Ordering	167
6.6.3	Span Estimation	171
6.6.4	Start/End Year Estimation	172
6.6.5	Using Reasoning to Sharpen Year Estimates	174
6.6.6	Timeline Evaluation and Visualization	175
6.7	Experimental Results	177
6.7.1	CEO Extraction	177
6.7.2	Succession Information	179
6.7.3	Tenure Midpoint Estimates For Relative Order Re-Estimation	180
6.7.4	Transition Estimation	182
6.7.5	Timeline Construction	182
6.8	Related Work	183
6.9	Conclusion	186
<b>7</b>	<b>Conclusion</b>	<b>187</b>
7.1	Contributions	188
7.2	Potential Applications	192
	<b>Bibliography</b>	<b>194</b>
<b>A</b>	<b>Celebrities</b>	<b>210</b>
A.1	IP152	210
A.2	C21	212
<b>B</b>	<b>Individuals in Web Polyreference Experiments</b>	<b>213</b>
<b>C</b>	<b>Corporate Succession Data</b>	<b>214</b>
<b>Vita</b>		<b>220</b>

# List of Figures

1.1	The training process uses example relationships to find sentences which at-test those relationships. Then an extraction system is trained from those sentences. For training, the objects in the example relationship must be easy to identify. . . . .	4
1.2	Timeline of CEO Succession History for IBM, since its founding in 1914. Tics mark the years of transition between adjacent CEOs. . . . .	7
2.1	Example Sentence Labeling. Given the training tuple $t_i = (x_i, y_i) = (\text{“David Hasselhoff”}, \text{“Baltimore, Maryland”})$ , a set of sentences are labeled. In the sentence 1, “David Hasselhoff” is marked as the hook (H) $x_i$ , and “Baltimore, Maryland” is labeled as the target (T) $y_i^1$ . In sentence 2, annotation without negative examples is used, so that even though there is a candidate target, $y_i^2$ (Germany) , it is marked as background. In sentence 3, annotation with negative examples is used, so that a candidate target, $y_i^3$ (California), which is not the correct target is marked as spurious (S). . . . .	16
2.2	These graphs depict the permissible label sequences for the CRFs. The state transition graph on the top has one path which extracts a fact (H(ook) I(nterstitial), T(arget)), and another path which marks the hook and background (O) states. The state transition graph on the bottom has a path which marks a fact, a path which marks spurious facts (H(ook), E(rroneous interstitial), S(purious targets)), and a background model. Spurious targets are candidates of the correct type, but not the correct target. . . . .	23
2.3	These two sentences show training sentence labeling for two sentences. In the top sentence, “Kurosawa” is marked as the hook (H), and “March 23rd” is labeled as the target (T). The bottom sentence gives an example of a sentence with a spurious target, in this case “October 10th” which is not Aaron Neville’s birthday. . . . .	24
3.1	Type preferences can be used in conjunction with WordNet to establish a candidate set of answers. Here a question type preference for “color” would narrow the set of possible answers down to a set of colors, where the correct answer “black” would be easier to pick out. . . . .	34

3.2	A type preference in the question can be also linked to a Named Entity Recognizer. Here a preference for a wingspan is categorized as a magnitude which is then linked to the named-entity type, quantity. . . . .	34
3.3	The subtree depicts a portion of the WordNet ontology which has been augmented with a set of people all of who are different types of singers. . . . .	36
3.4	Another depiction of the descriptions of Bill Gates, placed within the WordNet ontology, with the observed descriptions <span style="border: 1px solid black; padding: 2px;">boxed</span> . . . . .	37
4.1	Example Training Sentence marked with features, for CRF training. Each line corresponds to an observation. Each word in the original sentence has an associated lexical unigram feature (T=X), except for the Hook and the Target, where the lexical feature has been removed to prevent over-generalization. The “word” feature is present for all words, except for first and last names (none of which appear in this snippet), and for the hook and target. . . . .	52
4.2	Example Extractions of PCL from a 150 document test set. The first two examples in each section are correct, and the third is an example of an incorrect extraction. Out-of-vocabulary words are replaced with the token “.”. . . . .	58
4.3	Example Extractions of CRF+E from a 150 document test set. The first two examples in each section are correct, and the third is an example of an incorrect extraction. . . . .	59
4.4	CRF+E Pre-Fusion Precision vs. Test Set Size. As test set size increases, pre-fusion per-extraction precision decreases. . . . .	60
4.5	CRF+E Pre-Fusion Hit-Rate vs. Test Set Size. As test set size increases, pre-fusion hit-rate increases. Results on people still living were omitted from hit-rate results for year-of-death. . . . .	60
4.6	Precision vs. Number of Extractions for PCL, NB+E, CRF, CRF+E. PCL has a few very precise extractions, while CRF+E has a combined high precision along with a relatively large hit-rate. . . . .	61
4.7	Example of differences between Label Marginal and Viterbi Label Extraction Methods. Using label marginals instead of the Viterbi label sequence allows more candidates to be considered in each sentence. Sometimes, the correct target is already marked by the Viterbi label sequence, but often it is missed by the Viterbi label sequence but assigned a reasonably large probability by the underlying model. . . . .	63
4.8	Precision vs. Number of Extractions. Though the Viterbi sequence in some cases has more precise extractions, the Label Marginal method always has significantly higher recall. . . . .	64
4.9	These six variant sentence extraction models are used in the remainder of the chapter. . . . .	65
4.10	Candidates ranked by PCL and Naïve Bayes for Fusion of extracted Birthday, Birthplace and Occupation facts. Candidates in <b>bold</b> are correct. . . . .	69
4.11	Candidates ranked by Viterbi Frequency and Weighted Confidence Estimation for Fusion of extracted Birthday, Birthplace and Occupation facts. Candidates in <b>bold</b> are correct. . . . .	70

4.12 Precision vs. Number of Extractions for Post-Fusion Accuracy. CRF+E curves dominate for Birthday, Birthyear and Year-of-death, but for Birthplace, CRF performs slightly better at most points in the curve. For Occupation, NB and CRF perform about equally. . . . .	72
4.13 In Web data, often there are multiple hooks and targets, or spurious targets in a given automatically segmented “sentence”. Determining what to annotate as the hook and target or spurious target has an impact on performance. . . . .	75
4.14 Hook and Target/Spurious Target markup strategies for when multiple hooks and targets/spurious targets appear in the same sentence. . . . .	76
4.15 CV+E Post-Fusion Accuracy vs. Test Set Size. As test set size increases, post-fusion accuracy increases. Thus, even while precision of extracted facts decreases (Figure 4.4), the increased recall allows for improved performance (Figure 4.5). . . . .	82
4.16 CV+E Post-Fusion Missing Values vs. Test Set Size. Part of the improved performance due to recall comes in the discovery of facts in later documents which are not present in the small test sets. . . . .	83
4.17 Post-Fusion Accuracy vs. Test Set Size across different extractors, CM+E, CV+E, PCL, and NB. The increased performance of post-fusion accuracy due to larger test set sizes is consistent across each of the extractors and the fields. . . . .	84
4.18 CV+E Post-Fusion Accuracy vs. Training Set Size. As with increasing the test set size, increasing the training set size leads to gains in performance. These gains are more erratic, suggesting that later documents have some negative effect in training, and may be introducing more noise into the training process. . . . .	85
4.19 CV+E Pre-Fusion Precision vs. Training Set Size. As the training set size increases, the precision of the extractors generally increases, and then drops at the 150 documents. Presumably, more data yields better extractors, but as the document set gets larger, later instances contaminate the extractors and lead to lower precision. . . . .	86
4.20 CV+E Pre-fusion Hit-Rate vs. Training Set Size. As the training set size increases, the recall decreases and increases erratically. This suggests that the recall thresholds are brittle. . . . .	86
4.21 Precision/Recall Curves for CV+E and CM+E Ranking Methods. CM+E weighted confidence gives better precision at all recall levels than CV+E Viterbi frequency. . . . .	90
4.22 Precision vs. Recall by Fusion score averaged over all 5 fields. For all fields except birthplace, there is a slight gain in performance by bootstrapping. . . . .	91
4.23 Effect of Coreference on occupation extraction with increased test set sizes. Performance improves at all test set size values. . . . .	94

4.24	An Example of Cross-Field Bootstrapping : First the birthday (BD) is extracted and the text marked (middle diagram). From this annotated corpus, birth year (BY) is discovered and the text marked (right diagram). The discovered birth year may appear in contexts where the discovered birthday does not and improve extraction of further data such as birth place. . . . .	98
5.1	Effect on Context Size for Clustering Pseudoname Documents (Dev Set). The 100-word context using either Centroid Clustering or Group Average with the NNP+MI weighting scheme achieves the best performance. Baseline performance is 66%. . . . .	118
5.2	Effect on Context Size for Clustering Pseudoname Documents (Test Set), with the oracle stopping threshold. The 50-word context using Group Average with the NNP+MI weighting scheme achieves the best performance. Most of the other techniques don't perform better than baseline. . . . .	118
5.3	Effect on Context Size for Clustering Pseudoname Documents (Test Set) with the stopping threshold determined by the training data. The 50-word context using Group Average with the NNP+MI weighting scheme achieves the best performance. Most of the other techniques don't perform better than baseline.	119
5.4	Stopping Thresholds for Clustering Methods over the Pseudoname Test Set using NNP+MI weighting. . . . .	120
5.5	Stopping Thresholds for Clustering Methods over the Pseudoname Test Set using TFIDF weighting. . . . .	120
5.6	Effect of extracted facts on Unsupervised Pseudoname Clustering (Dev Set). Methods which use automatically extracted facts show consistently beat the baseline, with the CM+E extraction method achieving the highest performance.	121
5.7	Effect of extracted facts on Unsupervised Pseudoname Clustering (Test Set). In the held-out test set, the CV+E extracted features achieve the highest performance. . . . .	122
5.8	Oracle Experiments where for each referent, the true biographic facts for each referent is used to build a seed cluster, and then pages were clustered to those seeds (Dev Set). Overall, the CV+E extractor yields the best performance on this task. . . . .	123
5.9	Oracle Experiments where for each referent, the correct features were used to build a seed cluster, and then pages were clustered to those seeds (Test Set). The CV+E and CM+E extractors with oracle feature sets yield the best performance on this task. . . . .	123
5.10	Representative Pages Experiment (Dev Set). Representative pages are used to form seeds for clusters to which all remaining documents are then clustered.	124
5.11	Representative Pages Experiment (Test Set). Representative pages are used to form seeds for clusters to which all remaining documents are then clustered.	125
5.12	Male and Female First Names Distribution, the relative number of referents per name, fit to a power law. Only the top 500 names are shown. . . . .	127
5.13	Last Name Distribution, the relative number of referents per name, fit to a power law. Only the top 500 names are shown. . . . .	128
5.14	Number of Referents per Name for W03 with a power law fit. . . . .	130

5.15	Number of Pages per Referent for W03 with a power law fit. . . . .	131
5.16	Number of Referents with a given number of pages for W03. . . . .	131
5.17	Unsupervised Clustering of Manually Found Polyreference Cases using two-fold cross-validation. For each model, the stopping threshold is optimized on one half of the collection, and used on the other half. Cluster Centroid performs significantly better than the other clustering methods. . . . .	133
5.18	Stopping Thresholds for Clustering Methods over the Polyreference Test Set using NNP+MI weighting. . . . .	134
5.19	Stopping Thresholds for Clustering Methods over the Polyreference Test Set using TFIDF weighting. . . . .	134
5.20	Unsupervised Clustering of Manually Found Polyreference Cases with Extracted Features. Unlike in the pseudoname case, extracted features do not appear to help in clustering. . . . .	137
5.21	John Smith corpus coreference resolution. The 50-word context using Centroid Clustering and TF-IDF achieves the best performance. . . . .	139
5.22	Stopping Thresholds for Clustering Methods over the John Smith Corpus using NNP+MI weighting. . . . .	140
5.23	Stopping Thresholds for Clustering Methods over the John Smith Corpus using TFIDF weighting. . . . .	140
6.1	Timeline of CEO Succession History for IBM, since its founding in 1914. Tics mark the years of transition between adjacent CEOs (equivalent to Figure 1.2).	147
6.2	CEO Mentions over time. CEOs from the past decade are mentioned much more frequently than CEOs from earlier times. . . . .	154
6.3	WFST which marks company-positions with in-line XML tags. Example output might be : “<company-position> Boeing/COMPANY Chief/title Executive/title Officer/title < /company-position>”. All strings in < and > are semantic tags with which words are annotated, not actual text strings. . . .	158
6.4	WFST which produces training data for CEO extraction. True managers for the given company have previously been annotated with ceofirst and ceolast tags, and other names of people as with the PERSON tag. In the above machine, IGN stands for ignore. . . . .	159
6.5	FST which annotates for succession information . . . . .	168
6.6	Tenure Span Estimates: Normalized confidence estimated for CEO to be in office for that year, with outliers removed. . . . .	172
6.7	Tenure Span Estimates: Normalized confidence estimated for CEO to be in office for that year, with outliers removed. Marked years indicate estimates of tenure midpoint for each person, and an estimate of the date of transition between the two people. . . . .	177
6.8	Precision at different points in the ranked list. CEOs just ranked first are 100% correct, CEOs ranked first or second are 93% correct. . . . .	179

6.9	Graph of duration of CEO tenures for Gannett. As the CEO manager list has many candidates, the past three candidates have been correctly picked out an identified (John Curley, Douglas McCorkindale, Craig A. Dubow), though only the transition between McCorkindale and Dubow is estimated, and incorrectly at that. Outlying years are removed. . . . .	183
6.10	Graph of duration of CEO tenures for Home-Depot. The relative ordering by tenure midpoint estimation is correct (Arthur Blank is followed by Robert Nardelli), and the transition year is correct. Outlying years are removed. . .	184
6.11	Graph of duration of CEO tenures for IBM. The relative ordering by tenure midpoint estimation is correct (Louis Gerstner followed by Samuel Palmisano), but the transition year is incorrect. Outlying years are removed. . . . .	184
6.12	Graph of duration of CEO tenures for UPS. The relative ordering by tenure midpoint estimation is inconclusive, and the midpoint estimates are similar due to skews in the outlying years, though the graphs suggest the correct ordering. . . . .	185

# List of Tables

1.1	First ten referents for “Jim Clark” for documents retrieved from a Google search. . . . .	5
1.2	Example of Collected Celebrity Biographic Fact Information. These examples were manually collected from the Internet. A complete list of the people used for the experiments can be found in Appendix A. . . . .	6
2.1	Examples of patterns in a Phrase Conditional Model along with example associated confidences for the relationship <code>birthday(x,y)</code> . . . . .	17
2.2	Example negative log-probabilities of words for Naive Bayes model for positive and negative example models for the relationship <code>birthday(x,y)</code> . . . . .	20
3.1	A set of questions from a trivia database and from TREC-8/9 where the question has information about the specific type of answer that is expected (e.g. a type preference for a rock 'n' roll musician). . . . .	32
3.2	For each of the above proper nouns, multiple type descriptions were extracted from news corpora. . . . .	37
3.3	The number of each extracted description for Bill Gates after grouping by WordNet synset. Unlike the prior examples, there are multiple different ways Bill Gates is referred to. . . . .	38
3.4	Example mistakes in automatically Induced Ontology . . . . .	39
3.5	Using the induced proper noun ontology (IPNO) improves the coverage of the system, while precision sees only a slight drop. . . . .	41
3.6	These are examples where the induced proper noun ontology finds an ontological fact and this fact is then used to answer a question. . . . .	41
4.1	All queries which appeared in the four weekly top-50 lists of queries for Lycos in February 2004, with the number of weeks they appeared on the list. Queries for particular people are in <b>bold</b> (24/75). In addition, a few entries refer to fantasy people (e.g. Harry Potter, Barbie, and G.I. Joe), and one entry refers to a particular person but not by their name (Super Bowl Streaker). Organizations account for another 11 top searches. . . . .	48

4.2	These two examples show the 11 facts for which extractors are automatically built and used to extract information from the Web. A “-” marks places where there was no information for that fact. The first 5 facts were automatically collected from a online biography site, while the data for the remaining facts was manually collected. . . . .	51
4.3	Pre-Fusion Precision of extracted facts for the different extraction systems, trained on 15 people each with 150 documents, and tested on 137 people each with 150 documents. PCL and CRF+E have the best precision. . . . .	56
4.4	Pre-Fusion Hit-Rate of extracted facts with the identical training/testing set-up as above. NB+E and CRF have the best hit-rate. . . . .	56
4.5	Pre-Fusion Per-person Deduplicated Precision of extracted facts for the different extraction systems, trained on 15 people each with 150 documents, and tested on 137 people each with 150 documents. Just as with Pre-Fusion precision calculated above, PCL has the highest deduplicated precision. . .	57
4.6	Pre-Fusion Per-person Deduplicated Recall of extracted facts with the identical training/testing set-up as above. Just as with Pre-Fusion Hit-Rate, the CRF has the highest per-person deduplicated recall. . . . .	57
4.7	Average Accuracy of the Highest Confidence (Best) and Most Frequent (Vote) across five extraction fields. The CRF trained with both positive and negative examples, with output by label marginals, and which performs fusion by weighted confidence is the most successful extractor. Training using negative examples gives a 5% increase for the Viterbi frequency fusion methods, and a 10% increase for the weighted confidence fusion. . . . .	67
4.8	Fusion Accuracy and Fusion MRR for biographic fact fields across Multiple Extraction Methods. CM+E, the Conditional Random Field with negative examples and confidence weighted fusion performs the best. . . . .	68
4.9	Performance of various methods for Automatic Annotation as shown in Figure 4.14. The best strategy annotates the closest hook and target pairs, and the longest hook and spurious target pairs. . . . .	77
4.10	Models which expect a relation in a specific order (hook followed by target) are compared with models which accept either order. In general, the former perform better. . . . .	79
4.11	Average Order-Invariant Model Performance. On average, the models which predict a hook followed by a target perform better than those which accept either order. . . . .	79
4.12	Familial Relationship Extraction – Fusion Accuracy. . . . .	80
4.13	Familial Relationship Extraction – Fusion MRR. On these data, the CM+E method works best on average. . . . .	80
4.14	Post-Fusion Accuracy for training on 750 pages, either with 5 people and 150 pages or 15 people and 50 pages, with CV+E system. For the same number of pages, it is better to have them come from different people than to take pages further down the ranked document list. . . . .	87
4.15	Bootstrapping Results : After 5 iterations there is a slight gain in all fields except birthplace, which had the lowest starting performance. . . . .	91

4.16	Effects of Coreference on Biographic Fact Extraction. The “-” column holds the performance of the system without coreference. On average, a simple coreference resolution methods leads to a small gain in post-fusion accuracy for every method. . . . .	93
4.17	System Accuracy for familial relationship extraction with and without coreference. The “-” column holds the performance of the system without coreference. . . . .	95
4.18	System performance by Mean Reciprocal Rank for familial relationship extraction with and without coreference. The “-” column holds the performance of the system without coreference. . . . .	95
4.19	Performance of Cascaded Fact Extraction and Fusion Models. (f) indicates that the best fused result was taken. birth year(f) means birth years were annotated using the system that discovered the most accurate birth years. . . . .	99
6.1	Rank of Correctly Found CEOs. For every company, the rank of the correct CEOs in the retrieved CEO list is given. For each company, the top ranked answer is in fact a valid CEO for the company. . . . .	161
6.2	Top 10 Most Confident CEO Extractions across all companies. When the retrieved CEOs are ranked by weighted confidence, 8/10 are CEOs for the hook company, which suggests that weighted confidence is reliable across corpora. . . . .	161
6.3	Top 10 Found CEOs for Boeing and Heinz, ranked by confidence weight. People in <b>bold</b> were CEOs for that company. Misspellings of CEOs names are also marked in <b>bold</b> . . . . .	162
6.4	Top 10 Found CEOs for Staples and Textron, ranked by confidence weight. People in <b>bold</b> were CEOs for that company. Misspellings of CEOs names are also marked in <b>bold</b> . . . . .	163
6.5	Extracted Birthdays for CEOs . . . . .	165
6.6	Extracted Birthyears for CEOs . . . . .	165
6.7	Extracted CEO Birthplaces . . . . .	165
6.8	Extracted CEO Occupations . . . . .	165
6.9	CRF Confidence estimates for management orderings. The correct ordering (Philip Condit then Harry Stonecipher) get .67 and .89 probability, more than the incorrect orderings which get .02 and .27 probability. Crucial lexical context for the two examples might be “took over”, and “after the resignation of” in the first and “pushed his predecessor” in the second. Because the two orderings have different features marked in the sentences, the probabilities for the different orderings do not sum to 1. . . . .	169
6.10	Succession ordering results for Boeing. Each ordering is shown along with the weighted confidence measure associated with that ordering. . . . .	170
6.11	Estimates of manager order, correct order in <b>bold</b> . All are correct except for Textron, where James Hardymon precedes Lewis Campbell, not the reverse. . . . .	170
6.12	Estimated Tenure Midpoints from weighted sum of densities. The tenure midpoints can be used to provide more reliable relative ordering. . . . .	173

6.13	Start Year Estimation. The correct answers are ranked by the confidence of the system in that answer. For all of the candidates, the correct year is in the top 5. . . . .	173
6.14	End Year Estimation (for CEOs who have actually retired). The correct year is in the top 3 for all of the companies. . . . .	173
6.15	Top 10 system candidates for Anthony O'Reilly's start year. 1973 is the year he became president of the company (not CEO), and 1979 is the year he became CEO. In 2001 he was knighted, in 1998 he retired, and in 1969 he joined the company. 1936 is his birthday. . . . .	174
6.16	A linear interpolation of the end date of one CEO and the start of his successor can lead to improved performance at estimating the transition year. Column 3 is the estimated end year of the preceding CEO, Column 4 is the estimated start year of the succeeding CEO. Column 5 is the estimated year of transition. In each row, all of the values should be the same (succession(x,y) $\rightarrow$ end(x,D), start(y,D), transition(x,y,D)). Years in <b>bold</b> are correct. . . .	175
6.17	Using transition year estimates to re-estimate the start year of the preceding CEO. Columns 3 and 5 give the rank of the correct answer in the ranked start year list before and after using the transition year information. A "-" indicates that the correct year was not found in the ranked list. . . . .	175
6.18	Rank of Correctly Found CEOs for the test set. Typically, the most recent CEOs are correctly found, while some of the older CEOs are missed. . . .	178
6.19	Successor Extraction. For 7/8 companies, an adjacent CEO to the top-1 was picked out correctly. For 6/8 companies, the succession ordering was also correctly chosen (in <b>bold</b> ). For 1 company, an adjacent CEO was picked, but the order wasn't correct (in <i>italics</i> ). . . . .	180
6.20	Additional Successor Extraction. Out of 3 cases, where an adjacent CEO is proposed after the top-1 choice, only one is correct. . . . .	180
6.21	Estimated Tenure Midpoints from weighted sum of densities. The table shows the tenure for the CEOs. All relative tenures found are correct except for Kroger, where Dillon's tenure midpoint estimate comes before Pichler's, and in the case of UPS where the difference between the dates is negligible. . . .	181
6.22	A linear interpolation of the end date of one CEO and the start of his successor. Column 3 is the estimated end year of the preceding CEO, Column 4 is the estimated start year of the succeeding CEO. Column 5 is the estimated year of transition. In each row, all of the values should be the same (succession(x,y) $\rightarrow$ end(x,D), start(y,D), transition(x,y,D)). Years in <b>bold</b> are correct. . . . .	182

# Chapter 1

## Introduction

Each year more than two hundred and forty terabytes of information in printed material is produced [Lyman and Varian, 2003], a rate that far exceeds human processing ability<sup>1</sup>. The acceleration of information creation has out-stripped the capacity of humans to absorb it, and in order to allow access to this increased knowledge the information must be analyzed, categorized, synthesized, and summarized. This thesis presents a set of methods designed to help people gain access to the content of large text collections based on the method of fact extraction.

One relevant aspect of the increased production of textual information is a high degree of redundancy in the information. While these redundancies may not be necessary for human understanding, repeated patterns can be very useful for computational systems. This dissertation presents a method for fact extraction from multiple documents where the redundancy of information across multiple documents is crucial to training the extraction

---

<sup>1</sup>At an average reading rate of one page a minute, reading one book (300 pages) takes 5 hours. Reading all of the books published in 1996 (around 60,000) would take one person 144 years, if they read 40 hours a week.

system and to extracting the information. Trained fact extraction systems are applied to text collections for use in a variety of information retrieval tasks: Question Answering, Cross-Document Coreference, Biographic Fact Extraction, and Timeline Construction.

## 1.1 Fact Extraction

This dissertation concentrates on **fact extraction**, the automatic extraction of knowledge about the world from text. This is distinct from more classically linguistic information about text such as part-of-speech tagging [Brill, 1992] and parsing [Collins, 1999]. Those tasks give information which more closely related to the structure of text rather than information about the world contained in text. The methods presented herein extract facts about objects in the world and primarily focus on relationships between objects such as named entities (e.g. people, organizations, locations, and dates). The concentration on named entities as opposed to described entities (e.g. “the man”) makes it easier to exploit the redundancy in the experimental corpora.

If every fact were expressed in language in only one way, extracting facts would be straightforward. Instead, natural language exhibits both ambiguity, so one sentence can have more than one meaning, and paraphrasing, in which the same content can be stated in multiple different ways. Language can encode facts in a variety of forms with phenomena such as lexical variation and phrase re-ordering. From this perspective, discovering paraphrases is a key problem to extracting semantics from text. For example, the fact `birthplace(“Miles Davis”, “Alton, Illinois”)` might be inferred from any of the following

sentences<sup>2</sup>:

- Miles Davis grew up in Alton, Illinois.
- Miles Davis's hometown is Alton, Illinois. (lexical variation)
- Alton, Illinois was where Miles Davis was born. (reordering)
- Alton, Illinois is Miles Davis's hometown. (reordering)
- Miles Davis was born to a dentist in Alton, Illinois. (phrase insertion).

Manually enumerating the different ways a fact can be expressed is extremely time consuming, so Chapter 2 presents a method for learning how to extract facts given example relationships. This method works well with relationships between named entities, since their form is highly constrained, making them easy to identify in text. Object types whose forms are less constrained would be more difficult to identify automatically. For example, it is difficult to mark up all text might be a definition. By limiting extraction to relationships between named entities, a human can manually specify some example relationships, and the system can find example sentences which contain instances of the specified relationships in a large corpus and build an extraction system (Figure 1.1). Section 2.2 presents the general framework for extraction and learning which is applied in Chapters 4 and 6. A method for finding negative examples (Section 2.3) and a novel use of sequence models for extraction are proposed (Section 2.4.3).

After training, the extraction system scans natural language text and finds instances of specific facts. Each fact is presumed to be contained completely in one sentence

---

<sup>2</sup>Chapter 4 gives found examples of sentences with birthplace information.

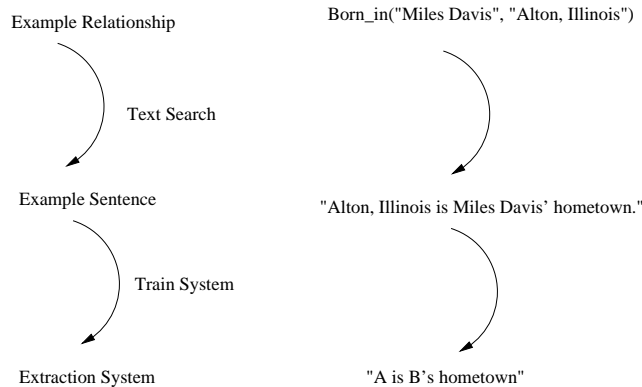


Figure 1.1: The training process uses example relationships to find sentences which attest those relationships. Then an extraction system is trained from those sentences. For training, the objects in the example relationship must be easy to identify.

and able to be extracted using only the words in that sentence. This necessitates that the facts not rely on extra-sentential semantic context to be true. Because these facts refer to objects in the real world, and do not rely on a particular context, they often appear in multiple sentences within a given corpus.

## 1.2 Question Answering and Cross-Document Coreference

One of the main themes of this dissertation is that fact extraction is a good building block for other natural language processing systems. Two particular applications are considered in this dissertation. Chapter 3 examines the task of **question answering**, where the user has a particular question that they want answered, and the system's goal is to return a short answer which exactly answers the question asked. Here, ontological information is extracted ("X is a Y") from a news corpus, and then this information is used to weed out potential answers to "Who is" questions.

Chapter 5 explores an information retrieval problem, where the extracted facts

1. Jim Clark - Race car driver from Scotland
2. Jim Clark - Clockmaker from Colorado
3. Jim Clark - Film Editor
4. Jim Clark - Netscape Founder
5. Jim Clark - Disaster Survivor
6. Jim Clark - Car Salesman in Kansas
7. Jim Clark - Fishing Instructor in Canada
8. Jim Clark - Computer Science student in Hong Kong
9. Jim Clark - Professor at McGill
10. Jim Clark - Gun Dealer in Louisiana

Table 1.1: First ten referents for “Jim Clark” for documents retrieved from a Google search.

from Chapter 4 are applied to the problem of personal name disambiguation or **cross-document coreference**. This information retrieval task deals with the problem of proper-name polyreference and attempts to resolve the ambiguity which occurs when a name (such as Jim Clark) refers to multiple people in the real world (see Table 1.1). To this problem is applied **centroid clustering** with and without extracted facts. First, a celebrity pseudonym polyreference World Wide Web evaluation set is created and described, and the biographic facts extracted by the prior methods are used to improve performance on disambiguating these pseudonyms. Next, a set of real polyreferent names is constructed and the extracted facts are shown not to improve performance on this set for reasons that are discussed. Finally, prior methods for this problem are discussed, and results on a standard news corpus evaluation set [Bagga and Baldwin, 1998] are presented which show the proposed coreference algorithm to give state-of-the-art performance on the standard test set. Extracted facts do not help for this test set either. This suggests that cross-document coreference on pseudonyms and on real polyreferent names may be significantly different tasks.

	Barbara Walters	Hermann Hesse
Occupation(s)	anchor,broadcaster journalist	author,writer novelist
Birthday	September 25	July 2
Birth year	1931	1877
Year of death	-	1962
Birthplace	Boston,Massachusetts	Calw,Germany
Spouse(s)	Robert,Lee,Merv	Marie,Ruth
Father	Lou,Louis	Johannes
Mother	Dena	Marie
Son(s)	-	Bruno,Heiner,Martin
Daughter(s)	Jacqueline	-
College	Sarah Lawrence College	-

Table 1.2: Example of Collected Celebrity Biographic Fact Information. These examples were manually collected from the Internet. A complete list of the people used for the experiments can be found in Appendix A.

### 1.3 Multi-Field, Multi-Document Extraction of Biographical Facts

Chapter 4 gives a thorough treatment of statistical fact extraction and cross-document fusion for the specific application of automatically generating **biographic summaries** containing 11 distinct attributes: birthday (day and month), birth year, birthplace, occupation, year of death, schooling, spouse, mother, father, and daughters and sons (as shown in Table 1.2). Following the framework outlined in Section 2.2, a method is explored and examined in which sentence extractors are trained from example facts, facts are extracted from multiple documents, and the resulting answers are fused. In particular, methods for automatic annotation are explored, various methods for multi-document information fusion are presented and discussed, learning curves are shown, and a method of bootstrapping the facts fused from the disparate sources is presented.

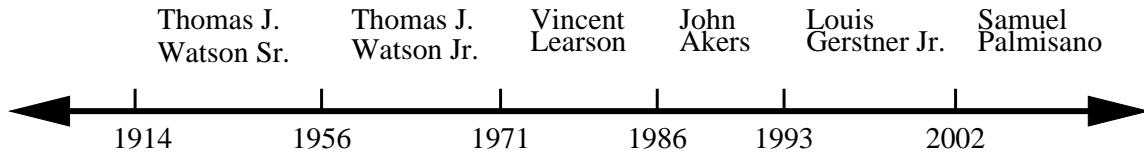


Figure 1.2: Timeline of CEO Succession History for IBM, since its founding in 1914. Tics mark the years of transition between adjacent CEOs.

Along with the experiments outlined above, there are some other experiments of particular interest to this domain. In particular, the use of pronominal coreference is explored to examine its capacity to improve extraction performance most in cases where there is less information available on a particular person or type of relationship. There is also a set of experiments on the use of fact extraction and fusion of one particular type of information to boost performance on extraction and fusion task for an alternative type of information. In particular, a **cascaded fact extraction and fusion** method uses partial biographic information to improve the performance of extraction of the remaining information.

## 1.4 Time-Bounded Facts and Timeline Construction

Prior chapters have treated facts as outside of considerations of time, but many facts are in practice **time-bounded** and only hold for a particular duration. Chapter 6 presents methods for extracting and reasoning about time-bounded facts. In particular, the chapter presents a method for extracting a timeline of management succession. This chapter demonstrates the range of applicability of the general method presented in Section 2.2. In order to build this database, a novel solution of **cascaded information retrieval, fact extraction and fusion stages** is applied, whereby partial information is extracted

at one stage, and later stages use this information to build larger pieces of knowledge. First the system extracts a set of candidate CEOs for a particular company. From this set, a pair of adjacent CEOs is selected and given a relative time ordering, and the time spans for the CEOs and the exact tenure dates for each CEO in the pair are estimated. This collection of a semantic network of connected facts demonstrates the utility of minimally supervised, multi-document, fact extraction and fusion.

## 1.5 Web Corpora

At many points in this dissertation, Web pages are retrieved and processed for training and for testing. In general, the set of Web pages is selected using Google by issuing a query to the server (using the Google API <http://www.google.com/apis/>), and the top ranked Web pages are downloaded and saved locally. The queries are typically named entities, though for CEO succession extraction there is additional lexical context that is used to disambiguate them from other similarly named people. Typically, the search is restricted to pages judged by Google to be in English that are encoded in HTML or plain text format. Web documents in PDF, Microsoft Word, PostScript and other complex encodings are discarded. Occasionally a Web page which is not in English is returned, but there are no additional filters which remove such pages.

A number of preprocessing steps are undertaken before the Web pages are in text form. First, a series of processes removes superfluous characters (i.e. control characters and non-ASCII characters) and HTML tags. The Penn tokenizer [MacIntyre, 1995] is then run over the remaining text. In many cases, a part-of-speech tagger [Florian and Ngai, 2001] is

then applied to the clean text.

The pages on the Web display a tremendous amount of variety. They are written by many different people, for many different reasons, and often take forms unlike the form traditionally present in a “balanced corpus” such as the Brown Corpus [Francis and Kucera, 1964], where news articles, editorials and prose constitute around 40% of the samples. On the Web, pages promoting books and music from Amazon.com and other sites are commonly retrieved from queries for people, and in general, commercial advertisements are represented more here than in traditional corpora. Bulletin-board and message-board postings, personal Web sites, and email lists are sources of informally written language which is not typically presented in traditional corpora. Finally, some pages are not text at all, but instead tables of information such as marathon race results. All of these sources complicate the task of extracting information and introduce a tremendous amount of noise especially after the removal of HTML tags.

## Chapter 2

# Training Fact Extractors by Example Facts

The term “Information Extraction” covers not only fact extraction but also a wide variety of natural language processing tasks. One of the more common subtypes of information extraction is field segmentation and labeling from either semi-structured text, such as seminar announcement field labeling [Freitag and McCallum, 1999], or from unstructured text as in named-entity finding [Bikel et al., 1999]. Fact extraction is a form of information extraction, related to template filling (MUC-3, 1991) where natural language text is scanned for instances of a particular relation or event.

This chapter presents a method for training a fact extraction system by example which will be used throughout this dissertation (Section 2.2). The framework allows for arbitrary classifiers and sequence models to be trained, and this chapter will briefly review three statistical models that are used as a basis for later experiments: two classifiers,

Phrase Conditional Models (PCL), and Naïve Bayes Models (NB), and a sequence model, Conditional Random Fields (CRF).

## 2.1 The History of Information Extraction

Since the early 1990s, there has been a tremendous amount of research into empirical methods for information extraction, spurred by “The Message Understanding Conference” (MUC) evaluations (MUC-1,1989 - MUC-7,1996). The trend through the decade has been away from fewer hand-written rules towards more machine learning.

The first systems used extensive amounts of hand-coded information. In the MUC-3 evaluation, the most successful system was the UMass CIRCUS system [Lehnert et al. 1991], which performed shallow parsing and used a large set of hand-built lexico-syntactic patterns and extensive dictionaries to extract entire frames of information. Surprisingly, this system performed as well as more sophisticated processing systems like SRI’s TACITUS system[Hobbs, 1986] which performed a full parsing and used abductive inference rules to deduce whether a given sentence supported a requested piece of information. The following year, in MUC-4, SRI responded with FASTUS, which implemented a cascaded finite-state architecture [Appelt et al., 1993], abandoned full parsing, relied on lexical rules for extraction, and performed the same type of extraction as the CIRCUS system using a robust inventory of lexical patterns. The simpler FASTUS performed significantly better than the more complicated TACITUS, in what was a surprising result at the time.

The major drawback of these early systems was the amount of human time and effort that were required to build them. Researchers spent weeks fine-tuning lexical rules and

dictionaries. In response, AutoSlog [Riloff, 1993] helped automate the dictionary creation phase needed for extraction from specialized domains. A large number of systems were later developed to learn extraction rules from annotated data, where the annotation consists of a set of sentences, each of which is annotated as to whether or not it contains a desired fact or not. Initially, researchers investigated specialized models such as WHISK [Soderland, 1999], but gradually more general statistical models such as Hidden Markov Models (HMMs) [Leek, 1997], Conditional Random Fields (CRFs) [Lafferty et al., 2001] and Support Vector Machines (SVMs) [Culotta and Sorensen, 2004] have become more popular.

These statistical methods still rely on time-consuming manual annotation and thus it takes a large investment of time in order to build a system for new domains. To compensate, there has been research in training specialized extraction models from example facts instead of from annotated data [Brin, 1998]. This chapter describes methods for training arbitrary statistical extraction models by example (Section 2.2).

Later in the dissertation, Chapters 4 and 6 present results on the performance of these automatically trained general extractors on different fact extraction tasks. Chapter 4 shows that using this generalized training method, a sequence labeling model can achieve higher performance than previously proposed methods. Chapter 6 presents results on training extractors by example for the domain of time-bounded facts, and demonstrates that the precision of extracted facts is high quality enough to be used in a pipeline of information.

## 2.2 System Overview

This section presents the framework for training arbitrary fact extractors by example. The next section discusses two schemes for automatically annotating sentence data, either with positive or negative examples, and the following section describes the sentence extractors that are used for the rest of the dissertation. The overall extraction process has two steps, training and extraction.

### 2.2.1 Training

The training model follows [Brin, 1998], [Agichtein and Gravano, 2000], and [Ravichandran and Hovy, 2002] which all start with a set of positive training facts, search the Web for instances where that relationship occurs, and learn a set of patterns from these example sentences. A formal description of the training process, which is slightly modified from the prior work, can be stated as follows.

1. A relationship  $R$  is defined as a set of tuples  $t = (x, y)$ .
2. In training, the system is given a set of  $I$  example tuples  $t_i = (x_i, y_i) \in R$ , where each  $x_i$  is a **hook** for a particular **target**  $y_i$ . For the extractors used in the dissertation, typically there are 10-20 examples per relationship.
3. For each training tuple  $t_i$ , a function  $\beta_R(x_i)$  creates a **hook query** for the **hook**, and it is sent to the Web search engine (in this dissertation, Google) to retrieve a set of relevant documents from the Web creating a separate **hook corpus**  $D_i$  for each training tuple. In biographic fact extraction, for a tuple  $(x, y)$  where  $x$  is a person,

and  $y$  the target (e.g. birthyear),  $\beta_R(x)$  is the last name of  $x$ . For the management succession task, many different types of hooks are used, including company names and complex title clauses (e.g. “CEO of Boeing OR Boeing CEO”). After being downloaded, the pages in the hook corpus are tokenized, part-of-speech tagged and marked with named-entities.

4. From each hook corpus  $D_i$ , the system finds the set of all  $K$  sentences, where each sentence  $s_i^k$  contains the hook  $x_i$  and at least one candidate target  $y_i$ . For example, for birthyear extraction, each sentence would require that the person’s name and a possible birthyear is present in each sentence. In order to find candidate targets, there must be a target set model  $A_R(y)$  which determines whether or not a word sequence is a candidate target for a relationship  $R$  (e.g. for birthyear the model might only allow words that are four digit numbers). From each training sentence  $s_i^k$ , a tuple  $t_i^k = (x_i, y_i^k)$  is found, where  $A_R(y_i^k) = \text{true}$ .  $s_i^k$  is considered to be a positive example of the relation if  $y_i^k = y_i$ , and a negative example if  $y_i^k \neq y_i$ . If a true target is found (i.e.  $y_i^k = y_i$ ), it is always marked. If a spurious target is found (i.e.  $y_i^k \neq y_i$ ), there are two methods for annotating that sentence (see Section 2.3).
5. The sentence extractors are then trained from these example sentences. A variety of models can be explored in this step. Considered in this dissertation are Phrase Conditional Models (Section 2.4.1), Naïve Bayes Models (Section 2.4.2), and CRF models (Section 2.4.3).

Earlier systems have explored the initial steps of the training process. The closest relative to annotating negative examples is [Agichtein, 2005] which presents a model for

ranking extraction patterns that takes into account bad extractions but doesn't generalize to annotating negative examples. The particular sentence extractors are all standard models, though the use of CRFs in this context is new and therefore given more detail.

### 2.2.2 Extraction

In extraction, a very similar process to training is undertaken.

1. A set of  $J$  query hooks  $x_j$  are given as input to the system (where  $I + 1 \leq j \leq I + J$ ).

Unlike other extraction systems, the goal isn't to find all tuples  $t \in R$  that are present in the corpus, but rather to find the elements  $y_j$  where  $t_j = (x_j, y_j) \in R$  for each query hook  $x_j$ .

2. For each hook  $x_j$ , a hook query is created using the function  $\beta_R(x_j)$ , and the search engine is queried in order to create a hook corpus  $D_j$ , and pages are downloaded.
3. From each hook corpus  $D_j$ , the set of  $k$  sentences  $s_j^k$  are selected, where each sentence must contain the hook  $x_j$  and a candidate target  $y_j^k$  where  $A_R(y_j^k) = \text{true}$ . For example, for birthyear extraction, only sentences which contain the individual's name and a four-digit-number would be retained.
4. Finally, the sentence extractors are applied to each sentence  $s_j^k$  and for each of the possible  $l$  candidates,  $\hat{y}_j^{l,k}$ , the system returns the confidence  $C_r(x_j, \hat{y}_j^{l,k}, s_j^k)$  that  $s_j^k$  indicates that  $t_j^l = (x_j, \hat{y}_j^l) \in R$ . It is assumed that only one target in a sentence can be correct.

1.	David/H Hasselhoff/H grew/o up/o in/o Baltimore/T ./o
2.	David/H Hasselhoff/H became/o famous/o in/o Germany/o ./o
3.	David/H Hasselhoff/H filmed/o Baywatch/o in/o California/S ./o

Figure 2.1: Example Sentence Labeling. Given the training tuple  $t_i = (x_i, y_i) = (\text{“David Hasselhoff”}, \text{“Baltimore, Maryland”})$ , a set of sentences are labeled. In the sentence 1, “David Hasselhoff” is marked as the hook (H)  $x_i$ , and “Baltimore, Maryland” is labeled as the target (T)  $y_i^1$ . In sentence 2, annotation without negative examples is used, so that even though there is a candidate target,  $y_i^2$  (Germany) , it is marked as background. In sentence 3, annotation with negative examples is used, so that a candidate target,  $y_i^3$  (California), which is not the correct target is marked as spurious (S).

## 2.3 Positive and Negative Examples

In training, once an initial set of  $K$  example sentences  $s_i^k$  are found for a hook  $x_i$ , there are two main alternatives for annotating the sentences to train the sentence extractors. The first alternative is to mark all hooks and targets and mark all other words as background (Figure 2.1, sentences 1 and 2). The second alternative is to attempt to find tuples  $(x_i, y_i^k) \notin R$  and mark those words  $y_i^k$  as spurious (Figure 2.1, sentence 3). In the markup phase, the system annotates the text with negative examples by searching each sentence  $s_i^k$  in the hook corpus  $D_i$  for the correct  $y_i$ , and if  $y_i$  isn't found in a particular sentence  $s_i^k$  it looks for other candidate targets  $y_i^k$  where  $A_R(y_i^k)$  is true and marks them as spurious. In some cases, there may be more than one spurious target per sentence, in which case only one is marked. The negative examples where a spurious target is marked are referred to as confusers<sup>1</sup>.

---

<sup>1</sup>Some of the details in the annotation method (in particular what happens when there are duplicates in one sentence) is explored in Section 4.5.2

X was born on Y/1  
X Date of Birth: Y/1  
X was born on Y/.93  
X, born Y/.5

Table 2.1: Examples of patterns in a Phrase Conditional Model along with example associated confidences for the relationship  $\text{birthday}(x,y)$ .

## 2.4 Sentence Extraction Methods

Sentence extractors (i.e. methods for extracting tuples from an individual sentence) are the core of the overall extraction system. In training, the sentence extractors are trained to extract a particular relationship, and in extraction they are used to find the targets for a given hook in a particular sentence. This section describes the sentence extraction methods that are used in this dissertation: a Phrase Conditional Model (Section 2.4.1), a Naïve Bayes Model (Section 2.4.2), and a Conditional Random Field Model (Section 2.4.3).

### 2.4.1 Phrase Conditional Models

Phrase Conditional Models (PCL) can be built from positive example annotation alone. In PCL models, there is a finite set of extraction patterns  $E_R^c(s)$ , which return the tuple  $(x,y) \in R$  if that the extraction pattern  $c$  was detected with arguments  $x$  and  $y$  in sentence  $s$  (see Table 2.1). The confidence of a given extraction pattern  $e^c$  of extracting a tuple  $(x,y) \in R$  from a sentence  $s$  is estimated as:

$$C_R^c(x,y,s) = P((x,y) \in R | e_s^c(x,y))$$

One naïve way to train this estimate might be:

$$P((x, y) \in R | (x, y) = E_R^c(s)) = \frac{\sum_{i=1}^I \sum_{s \in D_i} c((x_i, y_i) = E_R^c(s))}{\sum_{t_w=(x_w, y_w) \in (\Sigma^*, \Sigma^*)} \sum_{s \in D} c((x_w, y_w) = E_R^c(s))}$$

Where  $I$  is the number of training examples,  $\Sigma^*$  is the set of all strings, and  $D_i$  is the hook corpus for a given hook  $x_i$ .  $c((x, y) = E_R^c(s))$  is the number of times the tuple  $(x, y)$  was extracted by the extraction pattern  $c$ . To restate the above computation, it is the number of times that the extraction pattern is seen with arguments which are known to be in the desired relationship over the number of times the extraction pattern is seen with any arguments.

However, the above method cannot work as it treats all the tuples that aren't in the training set as negative examples. Since, for all of the cases considered later in the chapter, the goal is to find new, partially specified tuples<sup>2</sup>, the phrase conditional model is instead trained as:

$$P((x, y) \in R | (x, y) = E_R^c(s)) = \frac{\sum_{i=1}^I \sum_{s \in D_i} c((x_i, y_i) = E_R^c(s))}{\sum_{t_w=(x_w, y_w) \in (I(x), \Sigma^*)} \sum_{s \in D_i} c((x_w, y_w) = E_R^c(s))}$$

Here, the denominator counts the number of times when the extraction pattern found the hook without the corresponding target, and  $I(x)$  is all of the possible hooks in the training set. This probability estimates the conditional probability that  $y$  is the target given the extraction pattern and the hook  $x$ . For this model, there is no need for the target set model ( $A_R(y)$ ), since there are a finite number of extraction patterns. Like the other models

---

<sup>2</sup>In these partially specified tuples, the hook is known but the target is unknown.

presented in this section, the model can be best estimated when all possible targets for a given hook are known, but in later chapters is shown to be effective even when this is not the case.

Typically, only one extraction pattern will be found in a given sentence, but if more than extraction pattern gives a non-zero probability to a target for a given sentence, the max is taken:

$$C_R^M(x, y, s) = \max_{(x,y)=E_R^c(s)} P((x, y) \in R | (x, y) = E_R^c(s))$$

### 2.4.2 Naïve Bayes Classifiers

One drawback of the PCL model is that it requires exact string matches and doesn't do any generalization across found patterns. As an alternative, the Naïve Bayes (NB) unigram classifier assigns a probability to a found pattern based on the constituent words and estimates the probability of a word being part of a valid pattern across all training contexts in which the word is seen. The resulting classifier is less precise than the Phrase Conditional Model, but as Chapter 4 will show, its improved recall will give it better overall performance in the context of fusion.

In the NB models considered in this dissertation only interstitial words are considered, but the presentation here extends to prefix and suffix strings. Given a sentence  $s$

Word	$P(n_i (x, y) \in R)$	$P(n_i (x, y) \notin R)$
born	4.50	13.9
Birth	4.79	9.69
years	9.30	8.92
is	6.59	5.62
died	16.21	8.93
biography	5.74	16.91

Table 2.2: Example negative log-probabilities of words for Naive Bayes model for positive and negative example models for the relationship  $\text{birthday}(\mathbf{x}, \mathbf{y})$ .

made up of words  $w_{0..k} x w_{l..m} y w_{p..q}$ :

$$\begin{aligned}
C_{NB}(x, y, s) &= P((x, y) \in R | s) \\
&= P((x, y) \in R | x, w_{l..m}, y) \\
&= P(w_{l..m} | (x, y) \in R) P((x, y) \in R) P(w_{l..m})^{-1} \\
&= \prod_{i=l}^m P(w_i | (x, y) \in R) P((x, y) \in R) P(W_{l..m})^{-1}
\end{aligned}$$

The second step follows from Bayes' Theorem. The final step uses the Naïve Bayes assumption that each of the words are conditionally independent of each other given the underlying relationship. An appropriate prior is selected for  $P((x, y) \in R)$  using development data, and add-lambda back off is performed to account for unknown words.  $P(w | (x, y) \in R)$  is then estimated using the number of times the word  $w$  appears in a string  $x..w..y$

With only positive examples, the only way to calculate  $P(w_{l..m})$  is using a language model, such as a unigram word model. With negative example annotation, it is also possible to calculate  $P(w_{l..m})$  using the number of times the word appears in a string  $x..w..p$ , where  $p$  is a spurious target. Table 2.2 shows an example of some unigram probabilities in a Naïve Bayes model.

This model, Naïve Bayes trained with negative examples (**NB+E**) for  $P(n)$ , was found to be better performing than then the model trained without negative examples, and only results with this second model will be shown. In practice, this model assigns a score to each sentence and candidate target and is able to differentiate between real and spurious targets.

### 2.4.3 Conditional Random Field Models

Linear-Chain Conditional Random Field (CRF) Models [Lafferty et al., 2001] have been used for a wide variety of sequence labeling tasks including information extraction. These models give the probability of a label sequence  $L$  given an observation sequence  $W$ , where each observation is assigned exactly one label. For fact extraction, the CRF label sequence can encode the presence or absence of a given fact, where the hook is marked as such, and candidate targets are either marked as targets, left alone, or marked as spurious targets (if the model is trained with negative examples).

[Lafferty et al., 2001] provides a detailed explanation of CRF training and estimation, and the Mallet toolkit [McCallum, 2002] was used for the experiments performed in this dissertation. What is most relevant for this dissertation is that a CRF is trained by a given sequence of labels and observations, where each observation is a set of binary features. In this dissertation, a Markov assumption of length 1 is made for the linear chain CRFs, so that binary features can only encode information about the current label, the prior label or the observation sequence. One major advantage of CRFs over the Naïve Bayes model is that the features do not have to be independent, and so, for example, it is possible to have both unigram and bigram features.

## CRF Structures for Fact Extraction

Since the system is automatically annotating the training data for the CRF (as in Section 2.3), and thus determining the label sequence, there is a great deal of flexibility and the choice of how to use negative examples in training is especially important. Instead of allowing any possible sequence of labels (i.e. a fully connected model), the dissertation considers two broad variants of label sequences. The first model labels only the hook and a target if it is present. The second model labels the hook and either the target or a spurious target. As depicted in Figure 2.2, there are two topologies for these models. In the first (referred to as **CRF**), the model can either tag an entire sentence as irrelevant (with O label only), or can pick out a piece of information with H (hook), I (interstitial), T (target) labels. In the second (referred to as **CRF+E**), the model can additionally tag a sentence as containing a spurious target with H (hook), E (erroneous interstitial), and S (spurious target) labels.

Figure 2.3 gives an example of the training data produced by training with both positive and negative examples for the birthday relation. Training without negative examples would lead to an identical first training sentence, while each word in the second training sentence would be labeled with the O background label. In these examples, the target relationship is birthday, which is indicated by a label of T, when dates are marked the DATE feature (as a way of labeling potential candidates). Confusers are marked as S. The relationship which appears in the first sentence is `birthday('Akira Kurosawa', 'March 23')`.

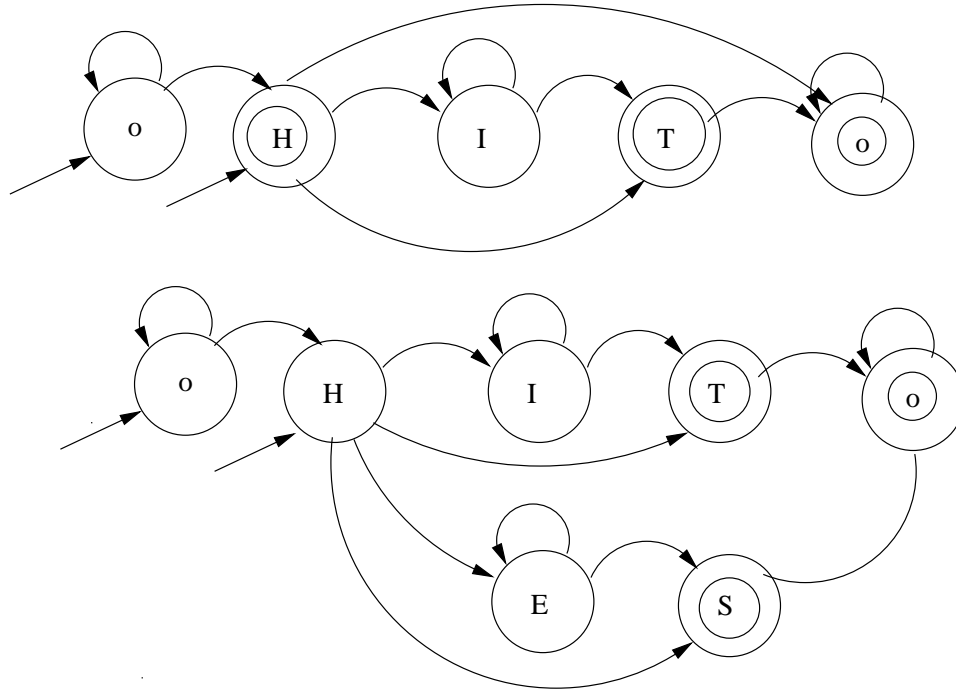


Figure 2.2: These graphs depict the permissible label sequences for the CRFs. The state transition graph on the top has one path which extracts a fact (H(ook) I(nterstitial), T(arget)), and another path which marks the hook and background (o) states. The state transition graph on the bottom has a path which marks a fact, a path which marks spurious facts (H(ook), E(rroneous interstitial), S(purious targets)), and a background model. Spurious targets are candidates of the correct type, but not the correct target.

### Extracting Confidence Estimates from a Trained CRF

After the CRF has been trained with paired label and observation sequences, given a new observation sequence  $W$  the CRF will yield a distribution  $P(L|W)$ . There are two ways in which the output distribution  $P(L|W)$  is used in this dissertation. First, the system can find the most likely label sequence for a given observation, which is called the Viterbi label sequence.

$$L^* = \underset{L}{\operatorname{argmax}} P(L|W)$$

World/o Cinema/o :/o Directors/o -/o Akira/o Kuro- sawa/H AKIRA/I KUROSAWA/I Born/I :/I Omori/I ,/I Tokyo/I ,/I Japan/I ,/I March/DATE-T 23rd/DATE-T 1910/o ./o
--

THE/o NEVILLE/o BROTHERS/o ON/o LATE/o NIGHT/o WITH/o CONAN/o O'BRIEN/o THIS/o FRIDAY/o ,/o NOVEMBER/o !/o !/o The/o Neville/H Brothers/E will/E be/E performing/E on/E Late/E Night/E with/E Conan/E O'Brien/E this/E coming/E Friday/E ,/E October/DATE-S 10th/DATE-S on/o NBC/o ./o
---

Figure 2.3: These two sentences show training sentence labeling for two sentences. In the top sentence, “Kurosawa” is marked as the hook (H), and “March 23rd” is labeled as the target (T). The bottom sentence gives an example of a sentence with a spurious target, in this case “October 10th” which is not Aaron Neville’s birthday.

Given appropriate model training, the label sequence will then encode whether or not a fact was extracted from a sentence and if a fact is found what it is. With the type of training sentences as presented above, the model will mark as H(ook) the given hook and will mark a candidate target as T(arget) to indicated that the specified relationship holds between that hook and target. If the Viterbi label sequence marks an extracted fact, the probability of that sequence  $P(L^*|W)$  can be used as a measure to judge the probability that the extracted answer is expressed by the sentence.

$$C_{crf}^*(x, y, s) = P(L^*|W)$$

This measure doesn’t seem appropriate for the task, since it measures the entire probability of the label sequence, not the probability that the a given relationship is present in the sentence. As a consequence, any one labeling of a longer sentence will typically be less likely than any one labeling of a shorter sentence because the probability has to be split between

many more candidate labelings. Another shortcoming of the Viterbi parse probability as a measure of confidence of extraction is that this measure gives no credit to labelings other than the most probable labeling.

The second way the CRF output distribution is used in this dissertation is to calculate the marginal distribution of a certain subset of labels (also called the field confidence) [Culotta and McCallum, 2004]. This distribution is calculated by fixing the hook and target labels, and then summing over all possible labels for the other observations. To calculate these label confidence estimates, first the text is segmented by possible answer candidates (i.e. all dates are marked). Next, for each possible answer candidate, the system calculates the probability of that candidate receiving the target label and that label occurring nowhere else in the sentence. If  $y = W_m$  is the observation element which corresponds to a given candidate, and  $l_r$  is the relationship label:

$$C_{crf}^{CE}(x, y, s) = P((x, y) \in R|W) = P(L_m = l_r|W) = \sum_{L': L'_m = l_r} P(L'|W)$$

The label marginal probability has a number of major advantages over the Viterbi labeling for use as a confidence estimate. First, it allocates some probability to each potential candidate in the sentence, as opposed to the Viterbi label sequence which chooses only the most likely target or chooses none at all. Second, it removes the effects of sentence length, since the probability is computed only with respect to the labeling of a particular state, not over the entire label sequence. Section 4.5 shows how these properties allow it to be used for fusion.

#### 2.4.4 Knowledge Engineering

In most extraction systems, there is a significant amount of human knowledge and effort that is placed into the system to extract each particular type of information. Though this system has less knowledge needed than most, there are still a few places where human knowledge must come into play during system development.

1. **Example Selection:** Someone must decide which examples to present to the system to learn from. Ideally, the elements in examples should be frequent and each of the elements in the tuple should be unambiguous. This step is a huge improvement over other training methods which require annotation of a large corpus, yet it still includes a fair amount of art in picking examples.
2. **Information Annotation and Target Set Modeling:** Deciding exactly how to annotate the hooks and target sets must be done on a per-relation basis. For the extractors described in this dissertation, the hooks took very little time, and target sets required at most a couple of hours to model appropriately. Automatic induction of potential target sets could be attempted in future work, such that given a training set of examples, it might be possible to automatically characterize the types of targets searched for, and thus eliminate the need for a human-created target set.
3. **Feature Selection:** As with most IE systems, a human must decide which features to provide for the classifier to learn to learn the weights of. (Chapters 4 and 6 go into more detail on this point).

### 2.4.5 Alternate Classifiers

The above three methods, Phrase Conditional Models, Naïve Bayes Models, and Conditional Random Field Models are used in this dissertation to train fact extractors. However, within the framework presented which demonstrates how to annotate a corpus with negative examples, any arbitrary classifiers or sequence models could be used. Some of the other classifiers and sequence models that have been applied to information extraction are Hidden Markov Models [Leek, 1997], Probabilistic Context Free Grammars [Miller et al., 1998], Support Vector Machines [Zelenko et al., 2003], and Maximum Entropy Models [Chieu and Ng, 2002]. Alternatively, specialized information extraction models like CRYSTAL [Soderland et al., 1995] and WHISK [Soderland, 1999], RAPIER [Califf and Mooney, 1998], SRV [Freitag, 1998] and Boosted wrapper induction [Freitag and Kushmerick, 2000], and (LP)<sup>2</sup> [Ciravenga, 2001] could be used in this setting to generate sentence extractors.

## 2.5 Limitations to Example-Based Training

This dissertation demonstrates that example based training can be applied to different kinds of facts, like biographic facts and familial relationships (Chapter 4), as well as management roles and relative ordering (Chapter 6). One of the most important characteristics for training from these facts is that they be frequently attested in training. If an example appears only once in a corpus, choosing that example amounts to little more than annotating one sentence. In addition, these types of fact do not cover all possible information, such as facts which are stated across sentences.

There are a few deficiencies with this training procedure which are of import. There may be false negatives, that is sentences which contain correct information but which are not labeled as such because the information is not known. There may also be false positives, cases where the sentence appears to contain the desired information but does not. In these cases, the system will erroneously tag those sentences as positive examples which may cause a degradation of system performance. Section 4.5.1 gives a thorough discussion of the effects due to these deficiencies.

Perhaps the most crucial deficiency is that the relationship learned from examples is the most salient relationship between the entities, even though many other relationships may exist. For example, in the case of management succession, though the most salient relationship between a company and a person may be the role the person assumes in the company (X is a CEO of Y), there may be other relationships between them (e.g. X sued Y).

### **2.5.1 Alternative Methods for Training via Minimal Supervision**

While example-based learning of information extraction systems is attractive, there are other minimal supervision training methods which have been proposed. Meta-bootstrapping [Riloff and Jones, 1999a] has been proposed as a method for finding extraction patterns using text and a small set of seed words. The system iteratively builds dictionaries and extraction patterns which are useful for building those dictionaries. Similarly, ExDisco [Yangarber et al., 2000] explores a related method which bootstraps patterns from relevant documents and a set of seed patterns.

In an interesting combination of the above approaches, [Etzioni et al., 2004, Et-

zioni et al., 2005] propose an alternative training method, where for each predicate (e.g. `MayorOf(X,Y)`), the system is given one or more predicate label (e.g. “mayor of”) and class labels for X and Y (i.e. “mayor” and “city”). Essentially, the system is given an example extraction pattern and colloquial names for the arguments for the extraction pattern (as opposed to example seed words). From this information, the system bootstraps an extractor.

## 2.6 Conclusion

This chapter has presented a system for training a fact extractor by example, and has explained the core sentence extractors (Phrase Conditional Likelihood, Naïve Bayes, and CRFs) that will be used in Chapters 3 through 6 to extract biographic facts and corporate management succession information. In particular, this chapter introduced a method for using negative examples for training arbitrary classifiers. This general method for training by example will be the basis for training, and Chapter 4 will demonstrate the effectiveness of this training method, as well as point out some crucial details associated with the method. Chapter 6 will present experiments which apply these methods to CEO management succession and demonstrate the use of a cascade of fact extractors.

## Chapter 3

# Ontology Induction for Question Answering

Chapter 2 introduced a set of methods for fact extraction. This chapter explores the use of multi-document fact extraction to build a knowledge base that can be used by a question answering system. In particular, this chapter considers the notion of a fine-grained proper noun ontology and argues for the utility of such an ontology in retrieval tasks. To support this claim, a fine-grained proper noun ontology is built from unrestricted news text and this ontology is then used to improve performance on a question answering task.

### 3.1 Introduction

The WordNet lexical ontology [Miller, 1990] contains more than 100,000 unique noun forms. Most of these noun forms are common nouns (nouns describing non-specific members of a general class, e.g. “detective”). Only a small percentage of the nouns in Word-

Net are proper nouns (nouns describing specific entities, e.g. “[the detective] Columbo”)<sup>1</sup>.

The WordNet ontology has been widely useful and an up-to-date bibliography of work using WordNet can be found in [Csomai, 2005]. Some of the most important applications of WordNet have been in word sense disambiguation [Voorhees, 1993], information retrieval [Sussna, 1993], text classification [Scott and Matwin, 1998], prepositional phrase attachment [Brill and Resnik, 1994] and question answering [Pasca and Harabagiu, 2001]. These successes have shown that common-noun ontologies have wide applicability and utility.

There exists no ontology with similar coverage and detail for proper nouns. Prior work in proper noun identification has focused on named entity recognition [Chincor et al.1999]. In this task, each proper noun is categorized into one of a small number of types, for example PERSON, LOCATION, or ORGANIZATION.

These coarse categorizations are useful, but more finely grained classification might have additional advantages. While Bill Clinton is appropriately identified as a PERSON, this neglects his identity as a president, a southerner, and a amateur saxophone player. If an information request identifies the object of the search not merely as a PERSON, but as a **typed proper noun** (e.g. “a southern president”), this preference should be used to improve the search.

Unfortunately, building a proper noun ontology by hand is more difficult than building a common noun ontology, since the set of proper nouns is rapidly expanding, and already orders of magnitude larger than the common noun inventory. With a world

---

<sup>1</sup>A random 100 synset sample was composed of 9% proper nouns.

population of just under 6.5 billion people, around three hundred thousand new proper nouns (or at least new referents) are introduced daily.<sup>2</sup> Thus, a broad-coverage proper noun ontology must be constantly updated, and to propose the use of a proper noun ontology, a method, however limited, must be presented to build a proper noun ontology.

This chapter explores the idea of a fine-grained proper noun ontology and its use in question answering. A proper noun ontology is built from unrestricted text using simple textual co-occurrence patterns (Section 3.3). This automatically constructed ontology is then used on a question answering task to give preliminary results on the utility of this information (Section 3.4).

## 3.2 Ontologies for Question Answering

What <b>king of Babylonia</b> reorganized the empire under the Code that bears his name?
What <b>rock 'n roll musician</b> was born Richard Penniman on Christmas Day?
What is the oldest <b>car company</b> which still exists today?
What was the name of the first <b>Russian astronaut</b> to do a spacewalk?
What was the name of the <b>US helicopter pilot</b> shot down over North Korea?
Which <b>astronaut</b> did Tom Hanks play in 'Apollo 13'?
Which former <b>Klu Klux Klan member</b> won an elected office in the U.S.?
Who's the <b>lead singer</b> of the Led Zeppelin band?
Who is the <b>Greek goddess</b> of retribution or vengeance?
Who is the <b>prophet</b> of the religion of Islam?
Who is the <b>author</b> of the book, "The Iron Lady: A Biography of Margaret Thatcher"?
Who was the <b>lead actress</b> in the movie "Sleepless in Seattle"?

Table 3.1: A set of questions from a trivia database and from TREC-8/9 where the question has information about the specific type of answer that is expected (e.g. a type preference for a rock 'n' roll musician).

Modern question answering systems rely heavily on the fact that questions contain

---

<sup>2</sup>From <http://www.wholesomewords.org/missions/greatc.html#birdatrate>.

strong preferences for the types of answers they expect. [Kupiec, 1993] observes that the “Wh” word itself provides preferences (e.g. “Who” questions prefer PERSON answers). The paper further observes that questions also include type preferences in other parts of the question. Sometimes these preferences occur within the “Wh” phrase (“what **color**”), and sometimes they are embedded elsewhere within the question (“what is the **color** ...”). In both, the question indicates a preference for colors as answers.

Current question answering systems use ontologies when these type preferences are detected. One simple method is as follows: when a type preference is recognized, the preference is located within the WordNet ontology, and children of that synset are treated as potential answers. Given the question “In pool, what **color** is the eight ball?”, and the ontology excerpt shown in Figure 3.1, the system can narrow down the range of choices. This approach has high precision: if the type preference can be located, and a candidate answer is found in a child node (in a suitable corpus context), then the candidate is likely to be the answer.

Instead of directly linking from Wh-phrases to ontology subtrees, [Harabagiu et al., 2000] links WordNet subtrees to named-entity types. Their system works as follows: given the question “What is the **wingspan** of a condor?”, it locates “wingspan” in the WordNet ontology. It then detects that “wingspan” falls into the MAGNITUDE subtree which is linked to the QUANTITY Named-entity type (Figure 3.2). This indicates that questions which have question words in the MAGNITUDE subtree prefer numbers as answers.

While the WordNet ontology is primarily composed of common nouns, it contains some proper nouns, typically those least likely to be ephemeral (e.g. countries, cities, and

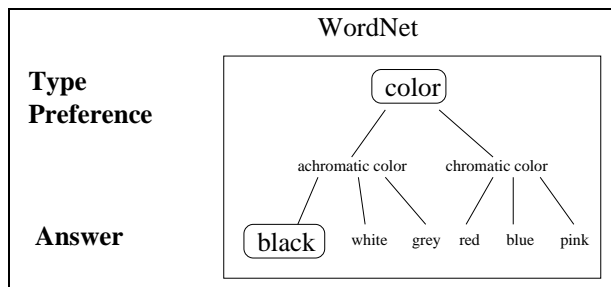


Figure 3.1: Type preferences can be used in conjunction with WordNet to establish a candidate set of answers. Here a question type preference for “color” would narrow the set of possible answers down to a set of colors, where the correct answer “black” would be easier to pick out.

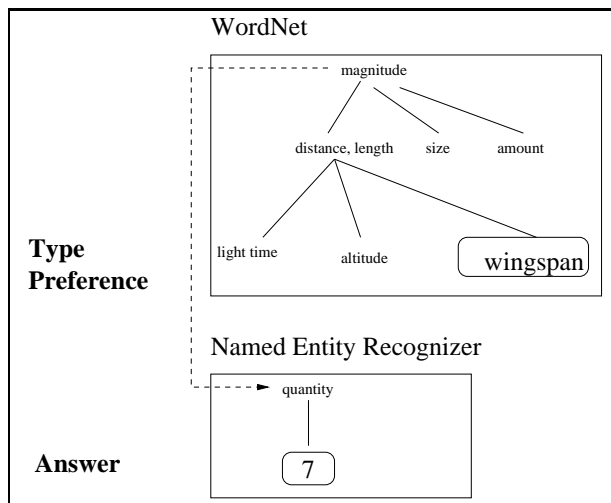


Figure 3.2: A type preference in the question can be also linked to a Named Entity Recognizer. Here a preference for a wingspan is categorized as a magnitude which is then linked to the named-entity type, quantity.

famous figures in history). These proper nouns in the ontology can be used just as other common nouns are. Given the question “Which composer wrote ‘The Marriage of Figaro’?”, the WordNet ontology will provide the fact that “Wolfgang Amadeus Mozart” is a composer, and this can help guide search as proposed in Figure 3.1.

Table 3.1 lists sample questions where a proper noun ontology would be useful. Some of the proper noun types are relatively static (Greek gods, kings of Babylonia). Other categories are more ephemeral (lead singers, British actresses). WordNet enumerates 70 Greek gods and 80 kings, but no lead singers and no British actresses.

[Ravichandran and Hovy, 2002] present an alternative ontology for type preference and describe a method for using this alternative ontology to extract particular answers using surface text patterns. Their proposed ontology is orders of magnitude smaller than WordNet and ontologies considered here, having fewer than 200 nodes.

### **3.3 Building a Proper Noun Ontology**

In order to better answer the questions in Table 3.1, a set of proper noun ontological facts were extracted from 1 gigabyte of newswire text collected from AP newswire for the years 1989, 1990 and 1995. To do so, the text was tokenized [MacIntyre, 1995] and part-of-speech tagged [Brill, 1992]. Next, instances of a common noun followed immediately by a proper noun were searched for (using the pattern “NN+ NNP+”). This is a untrained, unweighted version of a phrase-conditional model – much simpler than what was proposed in the prior chapter. While the system had decent performance with this pattern, the use of more sophisticated patterns might deliver improvements (see Section 3.5). This pattern

detects phrases of the form ‘[the] automaker Mercedes Benz’, and is ideally suited for proper nouns. In the newswire corpus this was a productive and high precision pattern, generating nearly 200,000 unique descriptions, with 113,000 different proper nouns and 20,000 different descriptions from around 500,000 pattern matches. In comparison, the “such as” pattern (Section 3.5) is matched less than 50,000 times in the same size corpora. Table 3.2 shows the descriptions generated for a few proper nouns using this simple pattern. From a sample of 100 pairs extracted from the AP-news text, 79 of the items classified as named entities were in fact named entities, and out of those, 60 (75%) had legitimate descriptions.

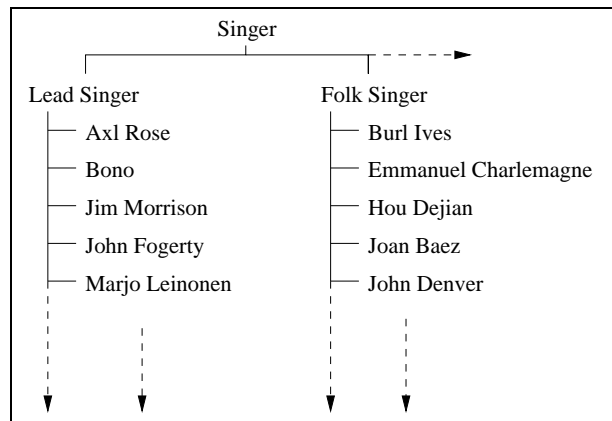


Figure 3.3: The subtree depicts a portion of the WordNet ontology which has been augmented with a set of people all of who are different types of singers.

To build the complete ontology, first each description and proper noun forms its own synset. Then, links are added from a description to each proper noun it appears with. Further links are put between descriptions “X Y” and “Y” (noun compounds and their heads), for example in Figure 3.3 “lead singer” is linked to the “singer” node. This process is somewhat noisy due to polysemous words and complex noun-noun constructions (“slalom king”)

This proper noun ontology fills many of the holes in WordNet’s world knowledge.

Proper Noun	Count	Description
Axel Rose	3	singer
	2	lead singer
	2	vocalist
Emma Thompson	3	actress
Mercedes-Benz	4	Luxury car maker
	4	car maker
	3	automaker
	2	family
	2	luxury
	1	gold
	1	service
	1	subsidiary

Table 3.2: For each of the above proper nouns, multiple type descriptions were extracted from news corpora.

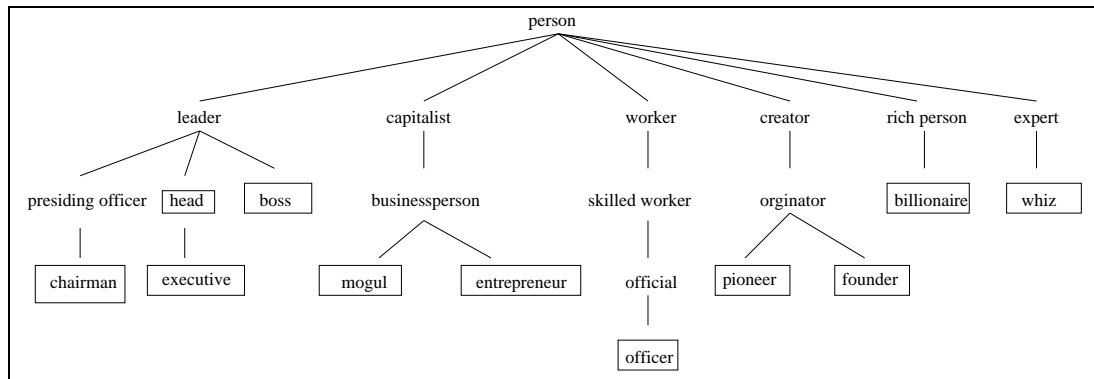


Figure 3.4: Another depiction of the descriptions of Bill Gates, placed within the WordNet ontology, with the observed descriptions boxed.

While WordNet has no lead singer synset or concept, the induced proper noun ontology contains 13 distinct lead singers (Figure 3.3). WordNet has 2 folk singers; the proper noun ontology has 20. In total, WordNet lists 53 proper nouns as singers, while the induced proper noun ontology has more than 900. While the induced ontology is not complete, it is more complete than what was previously available.

As can be seen from the list of descriptions generated by this pattern, people are described in a variety of different ways, and for popular people this pattern detects many

Proper Noun	Count	Description
Bill Gates	15	chairman
	9	mogul, tycoon, magnate
	2	officer
	2	whiz, genius
	1	pioneer
	1	head
	1	founder
	1	executive
	1	entrepreneur
	1	boss
	1	billionaire

Table 3.3: The number of each extracted description for Bill Gates after grouping by WordNet synset. Unlike the prior examples, there are multiple different ways Bill Gates is referred to.

different descriptions. Table 3.3 shows the descriptions generated for a common proper noun (“Bill Gates”). When the descriptions are grouped by WordNet synsets and the senses are manually resolved, the variety of descriptions decreases dramatically (Figure 3.4). “Bill Gates” can be described by a few distinct roles, and a distribution over these descriptions provide an informative understanding: leader (.48), businessperson (.27), worker (.05), originator (.05), expert (.05), and rich person (.02). Steve Jobs, who has a career path similar to Bill Gates, has a similar but distinct signature: originator (.6), expert (.4).

One immediate observation is that some of the descriptions may be more relevant than others. Is Gates’ role as a “whiz” as important as his role as a “billionaire”? The current system makes no decision and treats all descriptions as equally relevant and stores all of them. There is no need to weed out redundant descriptions, though incorrect descriptions may lead to errors in the final system.

The previous examples have focused on proper nouns which are people’s names. However, this method works for many organizations as well – one organization extraction is

Name	Mistaken Description
Greenspan	give
Macedonia	time
guerilla	blow
tension	rises

Table 3.4: Example mistakes in automatically Induced Ontology

shown in Table 3.2. However, while description extraction for people is high quality (84% correct descriptions in a 100 example sample), for non-people proper names, the quality of extraction is poorer (47% correct descriptions). This is a trend which requires further study. Table 3.4 shows some of the mistakes that are typical from the induced ontology. Typically, the mistakes appear to be a result of errors in the part-of-speech tagging.

### 3.4 Using a Proper Noun Ontology in a Question Answering Task

The above ontology was generated and to test its efficacy, it was used in a sentence comprehension task: given a question and a sentence which answers the question, extract the minimal short answer to the question from the sentence. This toy task is motivated as an easier version of short-answer question answering by the observation that extracting short answers is more difficult than extracting full sentence or passage length ones as evidenced by results in TREC QA tracks [TREC-9 Proceedings, 2000]. This system may act as a second processing step once a passage-length answer has been retrieved. Furthermore, retrieving answers from smaller document spaces may be more difficult than retrieving answers from larger ones, if smaller spaces have less redundant coverage of potential answers. In this task, there is virtually no redundancy. The trivia game corpus provided for this task had a

set of questions, short answers, and sentences in which that answer occurred. [Mann, 2002].

Baseline experiments used the WordNet ontology alone. From a semantic type preference stated in the question, a word was selected from the sentence as an answer if it was a child of the type preference. ‘Black’ would be picked as an answer for a ‘color’ type preference (Figure 3.1).

To utilize the induced proper noun ontology, from the raw data the head of each proper noun phrase and description were selected. Thus, for an extraction of the form “computer mogul Bill Gates”, the found pattern was “Gates ISA mogul”, and a node in the ontology was added. In cases of ambiguity, the proper noun is added into the ontology in multiple places<sup>3</sup>.

This induced proper noun ontology was put into the pipeline as follows: if WordNet failed to find a match, the induced proper noun ontology was used. If that ontology failed to find a match, the question was ignored. This is a very shallow level of processing. In a full system, a named entity recognizer or alternative method might be added to resolve the other questions (e.g. [Mann, 2002]).

One thousand trivia game questions were selected at random as a test set. Table 3.5 shows the results of the experiment with Wordnet alone, the Induced Proper Noun Ontology (IPNO), and a system with the two merged ontologies. The boost from merging the ontologies is clear: a gain of 14% (145 vs. 127) in recall, or questions correctly answered, with a .04% (75.1 to 74.7) decrease in precision. Gains made by inducing an ontology from an unrestricted text corpus (news text) and applying it to a unmatched test set (trivia

---

<sup>3</sup>Thus, Bill Gates would also be a bump on a ski slope, and a member of the Muslim dynasty that ruled India, the other senses of “mogul”.

Ontology	Correct	Total Answered	Precision
WordNet	127	169	75.1
IPNO	46	67	68.6
WN + IPNO	145	194	74.7

Table 3.5: Using the induced proper noun ontology (IPNO) improves the coverage of the system, while precision sees only a slight drop.

Correct Answer	Question
(Debbie) Reynolds	What <b>actress</b> once held the title of ‘Miss Burbank’?
(Jim) Lovell	Which <b>astronaut</b> did Tom Hanks play in ‘Apollo 13’?
Xerxes	Which Persian <b>king</b> moved an invasion force across the Hellespont on a bridge of ships?
(Donna) Summer	What was the name of the female Disco <b>singer</b> who scored with the tune ‘Dim All the Lights’ in 1979?
MGM	The 1974 film ‘That’s Entertainment!’ was made from film clips from what Hollywood <b>studio</b> ?

Table 3.6: These are examples where the induced proper noun ontology finds an ontological fact and this fact is then used to answer a question.

games), suggests that a broad-coverage general proper noun ontology may be useful.

It is further surprising that this improvement comes at such a small cost. The proper noun ontology wasn’t trimmed or filtered – all extracted facts were used. The only disadvantage of this method is simply that its coverage is small. Coverage may be increased by using ever larger corpora. Alternatively, different patterns (for example, appositives) may increase the number of words which have descriptions. A rough error analysis suggests that most of the errors come from mis-tagging, while few come from correct relationships in the ontology. This suggests that attempts at noise reduction might lead to larger gains in performance.

Another potential method for improving coverage may leverage related data. Our test corpus contained a question whose answer was “Mercedes-Benz”, and whose type

preference was “car company”. While our proper noun ontology contained a related link (Mercedes-Benz automaker), it did not contain the exact link (Mercedes-Benz car company). However, elsewhere there existed the links (Opel automaker) and (Opel car company). Potentially these descriptions could be combined to infer (Mercedes-Benz car company).

Formally :

$$(B\ Y)\ \textit{and}\ (A\ Y)\ \textit{and}\ (A\ Z)\ \Rightarrow\ (B\ Z)$$

$$\begin{aligned} &(\text{Mercedes-Benz automaker})\ \textit{and}\ (\text{Opel automaker})\ \textit{and}\ (\text{Opel car company}) \\ &\Rightarrow (\text{Mercedes-Benz car company}) \end{aligned}$$

Expanding descriptions using a technique like this may improve coverage. Still, a proposed system must ensure that the proper inferences are made since this rule is not always appropriate, possibly using a probabilistic model of inference. They are both founders of computer companies, but Bill Gates is a ten-billionaire while Steve Jobs isn't. Additionally, ambiguous names (e.g. “Paul Simon”) and descriptions (e.g. “mogul”) might cause additional problems here.

### 3.5 Related work

There has been considerable work in the past decade on building ontologies from unrestricted text. Typically, these methods operate over both common nouns and proper nouns, without distinguishing between the two. [Hearst, 1992] uses textual patterns (e.g. “such as”) to identify common class members. [Caraballo, 1999] and [Caraballo, 1999] augment these lexical patterns with more general lexical co-occurrence statistics (such as relative entropy). [Roark and Charniak, 1998] builds ontologies from noun-phrase co-occurrence statistics. [Berland and Charniak, 1999] use Hearst style techniques to learn

meronym relationships (part-whole) from corpora. [Phillips and Riloff, 2002] applies co-training to build semantic lexicons of ontological information.

There has also been work in building ontologies from semi-structured text, notably in the AQUILEX project (e.g. [Copestake, 1990]) which builds ontologies from machine readable dictionaries. MindNet [Richardson et al., 1998] is a related effort to build up semantic information from encyclopedia and other semi-structured sources.

The most closely related work is [Girju, 2001], which described a method for inducing a domain-specific ontology using some of the techniques described in the previous paragraph. This induced ontology is then potential useful for a matched question domain. The work presented in this chapter is closely related to what is presented in that paper as it also targets proper nouns, in particular people. Furthermore, this work presents initial results which attempt to measure coverage improvement as a result of the induced ontology.

[Srihari and Li, 1999] proposes the use of information extraction to aid question answering in two ways, first by tagging the corpus with named-entity information and second by directly transforming questions into information extraction queries. In their work, questions are fit into information extraction templates. The work presented in this chapter instead views information extraction as the process of creating a knowledge base which then can be leveraged to answer related questions.

Another related line of work is word clustering. In these experiments, the attempt is made to cluster similar nouns, without regard to forming a hierarchy. [Pereira et al., 1993] presents initial work by clustering nouns using noun-verb co-occurrence information. [Lin and Patel, 2001] extends these methods by using many different types of relations and

exploiting corpora of tremendous size.

One important distinction between the ontological induction methods and the clustering methods is that clusters are unlabeled. The ontology induction methods can identify that a “Jeep Cherokee” is a type of car. In contrast, the clustering methods group together related nouns, but their relationships may be subtle (e.g. “Sierra Club”, “Environmental Defense Fund”, “Natural Resources Defense Council”, “Public Citizen”, “National Wildlife Federation”). Generating labels for proper noun clusters may be another way to build a proper noun ontology.

The method used here to build the fine-grained proper name ontology also resembles some of the work done in coarse-grained named entity recognition. In particular, [Collins and Singer, 1999] presents a sophisticated method for using bootstrapping techniques to learn the coarse-classification for a given proper noun. [Riloff and Jones, 1999b] also shows a method to use bootstrapping to create semantic lexicons. These methods may be applicable for use in fine-grained proper noun ontology construction as well.

[Schiffman et al., 2001] describes work on producing biographical summaries. This work attempts to synthesize one description of a person from multiple mentions. This summary is an end in itself, as opposed to general knowledge collected. These descriptions also attempt to be parsimonious in contrast to the rather extended associations extracted by the method presented above.

Since the initial publication of this work, the general approach has been extended by a number of different people. [Fleischman and Hovy, 2003] examines the same task, answering “Who is ...” questions. In his approach, he collects a 15GB news corpus, and

learns a variety of classifiers for classifying extracted ontological facts as to whether they are correct or not. The best classifier they found is able to find correct facts with a precision of .95 and recall of .92 (as compared to their initially proposed candidates). From this corpus of 15GB of text and the improved extractors, they extract 930,000 unique ontological facts. They compared performance relative to a state-of-the-art question answering system and showed that the ontological facts substantially improved performance (by as much as 25%). Separately, [Hildebrandt et al., 2004] looks at definition questions and added an additional set of rules to pick out “definitional” information, and thus applied the techniques to definition questions in general, extending beyond questions about people.

### **3.6 Conclusion**

This chapter has demonstrated the potential of ontology construction for improving question answering performance. In particular, it presented a method for inducing a proper noun ontology from free text and for using this ontology in a question answering task. On the selected task, using the induced ontology substantially increased the recall of the system (by 14%) at a slight loss of precision (0.4%).

## Chapter 4

# Multi-Document and Multi-field Fact Extraction and Fusion

This chapter uses the architecture proposed in Chapter 2.2 to train fact extraction systems for biographic facts from examples. Phrase conditional models, Naïve Bayes, and Conditional Random Field models are all explored (Section 4.3) . Two variant Conditional Random Field model structures are proposed, one which uses negative examples and one which does not.

The main contributions of the chapter are in the use of cross-document fusion. Three alternative methods for fusion are explored: highest confidence, the most frequently extracted, and weighted confidence (Section 4.5). The effects of changing the test set size and the training set size is additionally explored (Sections 4.5.5 and 4.5.6). Experiments show that bootstrapping of extractors given a small seed set of examples works for this extraction/fusion system in a way similar to other bootstrapping methods (Section 4.6),

and that coreference can improve overall model performance (Section 4.7). Finally the chapter presents results using a cascaded fact extraction model (Section 4.8). This cascade model presents a model for extracting entire semantic networks, and demonstrates that redundancy in a given corpus allows the high-accuracy extraction of complex information. Section 4.9 presents related work.

## 4.1 Biographic Facts

This chapter focuses on people and the most common facts associated with them, biographic facts. People are a tremendously important topic of information needs. Lycos ([www.lycos.com](http://www.lycos.com)) makes available its 50 most popular Web searches each month (see Table 4.1), and for each week in February, 2004 on average 36% of the queries were for specific individuals by name. Given this large focus on people, it might make sense to give special care to handling searching for people. In pulling apart the various aspects of serving information needs about people, the biographic facts which surround mentions of individuals seems like an appropriate place to start<sup>1</sup> This frequency of mention also makes them an appealing object of study for they are both relevant and appear in large quantities.

These biographic facts have another property which makes them especially appealing for study, which is that they circumscribe a small domain of events, and the relationships between the various facts associated with a person are well known. For example, someone's birth must precede their death, and typically one's death does not fall more than 100 years after their birth. People (typically) have one father and one mother, though due to step-

---

<sup>1</sup>In 2005, Google started returning biographical information in response to Web queries with biographic keywords in them.

Query	Weeks on Top 50	Query	Weeks on Top 50
American Idol	4	The Bible	3
Atkins Diet	4	Valentine's Day	3
<b>Beyonce</b>	4	<b>50 Cent</b>	2
Black History Month	4	Barbie	2
<b>Britney Spears</b>	4	Baseball	2
<b>Brooke Burke</b>	4	Dragon Ball	2
<b>Carmen Electra</b>	4	Drudge Report	2
<b>Catherine Bosley</b>	4	IRS	2
<b>Christina Aguilera</b>	4	<b>John Kerry</b>	2
<b>Clay Aiken</b>	4	Mars Rover	2
Final Fantasy	4	<b>Pam Anderson</b>	2
Harry Potter	4	<b>Pamela Anderson</b>	2
<b>Hilary Duff</b>	4	Spam Rage	2
Inuyasha	4	The IRS	2
<b>Janet Jackson</b>	4	The Passion of the Christ	2
<b>Jennifer Lopez</b>	4	Weight Watchers	2
<b>Jessica Simpson</b>	4	<b>William Hung</b>	2
KaZaA	4	Yu-Gi-Oh!	2
Las Vegas	4	Care Bears	1
Lord of the Rings	4	Dragonball	1
NASA	4	G.I. Joe	1
NASCAR	4	Grammy Awards	1
NBA	4	Groundhog Day	1
NFL	4	<b>Howard Stern</b>	1
Neopets	4	<b>Justin Timberlake</b>	1
<b>Orlando Bloom</b>	4	Lingerie Bowl	1
<b>Paris Hilton</b>	4	Mardi Gras	1
Pokemon	4	NHL	1
Prom Dresses	4	Passion of Christ	1
South Beach Diet	4	RI Nightclub Fire	1
WWE	4	<b>Simon Cowell</b>	1
<b>Anna Kournikova</b>	3	Super Bowl	1
<b>Anna Nicole Smith</b>	3	Super Bowl Commercials	1
FAFSA	3	Super Bowl Halftime Show	1
Love Poems	3	Super Bowl Streaker	1
Marijuana	3	Survivor	1
Taxes	3	<b>Tupac Shakur</b>	1

Table 4.1: All queries which appeared in the four weekly top-50 lists of queries for Lycos in February 2004, with the number of weeks they appeared on the list. Queries for particular people are in **bold** (24/75). In addition, a few entries refer to fantasy people (e.g. Harry Potter, Barbie, and G.I. Joe), and one entry refers to a particular person but not by their name (Super Bowl Streaker). Organizations account for another 11 top searches.

parents and adoption, this may be not true<sup>2</sup>. In the United States, schooling typically follows a relatively orderly progression of elementary school, middle school, high school, and college, and occasionally higher education. The time associated with each phase of school is relatively fixed with respect to a person's life, and the ordering is almost inviolate.

Because of their ubiquity and because of the richness of the domain, these set of facts are of particular scientific interest and provide a good testing ground for extraction techniques and fusion methods. While the domain is primarily restricted to biographic facts, the corpora used are from the Web, which contains a very wide sampling of genres and styles. Chapter 6 demonstrates that corporate succession facts are amenable to the techniques described here.

Out of all of the potential biographic facts, this chapter primarily is concerned with facts which are static and non-contextual. The stability of this set of facts allows for them to be easily matched across various documents. The types of facts examined are nominal attributes, either dates of certain distinguishing events (birth, death), familial relationships (mother, father, children, spouse), or societal information (occupation and schooling). The attributes chosen are those which are listed in biographic databases such as infoplease.com, the IMDB database, and encyclopedias. Events are excluded from this list, both events central to identification (e.g. "X designed the American flag") and those which are incidental (e.g. "X bought milk at the grocery store"). The above bits of information are the ones most commonly discussed and most often used to describe a person. Thus, for this set of facts there are many available documents that contain these facts.

---

<sup>2</sup>The models presented here are limited by a 1-mother/1-father assumption

In the system presented in this dissertation, extraction performance depends on the ability of a system to pick out for each person likely candidates for each category of biographic fact. The above list of facts includes some easy categories which have candidates which are very easily marked or where the size of the candidate set is small (e.g. birth day – which has only 365 possibilities) to harder categories which are more difficult to mark and may have upward of ten thousand possible fillers (e.g. last names). Many other types of attributes such as phone number, email and height would fall in the easy end of the spectrum. Additional attributes of interest are those that have people as a filler (e.g. parents and children).

An example of the type of biographic facts considered in this dissertation can be found in Table 4.2. In this table, unknown or missing values are marked with “-”. It is assumed that the training and test examples have no errors, while they may contain contradictions. The two birthdays for Miles Davis reflect attestations from different source texts, and the name variants for Barbara Walter’s father (Lou and Louis) and for Miles Davis’ children (Miles IV and Miles) reflect the variety of usage on the Internet.

## **4.2 Fact Extraction Model Structure and Feature Set**

This chapter uses the system described in Section 2.2 to build fact extractors by starting with a set of example relationships, mining example sentences from the Web, and training sentence extractors from this set of example sentences.

Section 2.4 presented the sentence extraction models that will be used in this chapter : Phrase Conditional Models (PCL), Naïve Bayes Models trained with positive and

	Barbara Walters	Miles Davis
Occupation(s)	anchor,broadcaster journalist	musician,player, saxophonist
Birthday	September 25	May 25,May 26
Birth year	1931	1926
Year of death	-	1991
Birthplace	Boston,Massachusetts	Alton,Illinois
Spouse(s)	Robert,Lee,Merv	Frances,Cicely,Betty
Father	Lou,Louis	Miles
Mother	Dena	Cleota
Son(s)	-	Gregory,Miles IV,Miles,Erin
Daughter(s)	Jacqueline	Cheryl
College	Sarah Lawrence College	Juilliard School of Music

Table 4.2: These two examples show the 11 facts for which extractors are automatically built and used to extract information from the Web. A “-” marks places where there was no information for that fact. The first 5 facts were automatically collected from a online biography site, while the data for the remaining facts was manually collected.

negative examples(NB+E), and Conditional Random Field Models trained either only with positive examples (CRF) or with positive and negative examples (CRF+E). As the sentence extractors used in this chapter are general purpose estimators, they require some care in designing and choosing appropriate model structure and features.

In particular, lexical unigram and named-entity features were used. Lexical features which corresponded to particular training tuples were removed. In training and testing, the hook’s last name (e.g. “Davis”) was not used as a lexical feature. Likewise, in training, the lexical unigram features corresponding to correct and spurious targets were not used as features either. Figure 4.1 gives an example training instances for the CRF with the corresponding label sequence and unigram features.

Label	Features
<b>o</b>	T=Alistair word
<b>H</b>	HOOK
<b>I</b>	T=-LRB- word
<b>I</b>	T=November word DATE
<b>I</b>	T=20 word DATE
<b>I</b>	T=, word
<b>I</b>	T=1908 word YEAR
<b>I</b>	T=- word
<b>I</b>	T=March word DATE
<b>I</b>	T=30 word DATE
<b>I</b>	T=, word
<b>I</b>	T=2004 word YEAR
<b>I</b>	T=-RRB- word
<b>I</b>	T=was word
<b>I</b>	T=an word
<b>I</b>	T=England word LOCATION
<b>I</b>	T=, word
<b>I</b>	T=English-born word
<b>I</b>	T=United word
<b>I</b>	T=States word
<b>I</b>	T=, word
<b>I</b>	T=American word
<b>T</b>	occupation
<b>o</b>	T=and word
<b>o</b>	T=broadcaster word occupation
<b>o</b>	T=. word

Figure 4.1: Example Training Sentence marked with features, for CRF training. Each line corresponds to an observation. Each word in the original sentence has an associated lexical unigram feature ( $T=X$ ), except for the Hook and the Target, where the lexical feature has been removed to prevent over-generalization. The “word” feature is present for all words, except for first and last names (none of which appear in this snippet), and for the hook and target.

### 4.2.1 Feature Set

For all models, the primary source of information is in lexical unigram features. For the CRF the text is additionally annotated with word features. The word features used were marked personal names (from a name list), month and day pairs, words consisting of four-digits, and place names. Additionally, all common nouns were given the “word” feature and markers for the start and end of a sentence were added.

This dissertation does not present an exhaustive look at possible feature sets for the CRF models. It would be straight forward to add additional features such as lexical bigram features, part-of-speech, dictionary features, the distance between hook and target, and parse features.

### 4.2.2 Target Set Models

To select candidate sentences, target set models are required. The birthday model allows for four different ways of expressing a birthday:

- 12 November
- November 12
- 12th of November
- November 12th.

Any four-digit number is assumed to be a possible year, which excludes years between 999 B.C. and 999 A.D. Occupations were created by generating a list of frequent person descriptions from the WordNet hierarchy and then culling this list to retain only occupations.

To create a birthplace model, a list of places was taken from WordNet and augmented by all US state names and two-letter state name acronyms. The US Census publishes a list of first names available on-line, and these were used to create a list of names for relationship extraction.

In extraction, candidate targets that were not present in our target set models  $A_r(y)$  were rejected. In some cases, this resulted in the system being unable to find the correct target for a particular relationship, since it was not in the target set. The birthplace model, the familial relationship models, and the occupation model were all missing targets.

### 4.3 Biographic Fact Extraction

One hundred fifty two semi-structured mini-biographies were downloaded from an online site ([www.infoplease.com](http://www.infoplease.com)), and simple rules extracted a biographic fact database of birth day and month (henceforth birthday), birthyear, occupation, birth place, and year of death (when applicable). An example of the data can be found in Table 4.2. The system normalized birthdays and performed capitalization normalization for the remaining fields. There was no further normalization, such as normalizing states to their two letter acronyms (e.g. California  $\rightarrow$  CA). Fifteen names were set aside as training data, and the rest were used for testing<sup>3</sup>. For each name, 150 documents were downloaded from Google to serve as the hook corpus for either training or testing, and later experiments suggest that downloading further documents would yield increasingly small gains (Section 4.5.5).

In training, documents were automatically annotated using people in the training

---

<sup>3</sup>The first 15 people in Appendix A were used for training and the rest for testing.

set as hooks. In testing, grading was performed by exact match to the database. This is a very strict method of evaluation for two reasons. First, the extractors might have retrieved information that was simply not present in the database but nevertheless correct (e.g. someone’s occupation might be listed as actor and the retrieved occupation might be director, which could also be true). Second, since the retrieved targets were not normalized, there might have been retrieved targets that were correct but were not recognized (e.g. the database birthplace is New York, and the system retrieves NY).<sup>4</sup>

The choices in the system design overall were to allow rapid bootstrapping of facts in new domains with little adaptation. This design decision was realized in having only small number of training facts, not performing extensive normalization, and eschewing complicated feature engineering.

The performance of each extraction system is measured with regard to per-extraction **precision**:

$$\textit{Pre-Fusion Precision} = \frac{\# \textit{ Correct Extracted Targets}}{\# \textit{ Total Extracted Targets}}$$

Also examined was the **Hit-Rate**, which calculates the average number of times per person the correct target was extracted, as an approximation to recall. It is difficult to calculate true recall, since without manual annotation of the entire corpus, it cannot be

---

<sup>4</sup>This also has implications for training, since the models will be trained on annotated data that has errors. This has the most pronounced impact for the negative exemplar prediction models. This phenomena of missing and inaccurate data was most prevalent for occupation and birthplace relationships, though it was observed for other relationships as well.

	Birthday	Birth year	Occupation	Birthplace	Year of Death	Avg.
PCL	<b>78.9</b>	35.5	<b>30.5</b>	<b>51.0</b>	<b>52.7</b>	<b>49.7</b>
NB+E	42.3	36.1	25.5	21.7	8.8	26.9
CRF	50.9	34.2	21.9	13.9	26.7	29.5
CRF+E	68.0	<b>65.4</b>	24.6	35.7	31.4	45.0

Table 4.3: Pre-Fusion Precision of extracted facts for the different extraction systems, trained on 15 people each with 150 documents, and tested on 137 people each with 150 documents. PCL and CRF+E have the best precision.

	Birthday	Birth year	Occupation	Birthplace	Year of Death	Avg.
PCL	4.8	1.9	1.5	1.0	0.1	1.9
NB+E	<b>9.6</b>	11.5	20.3	<b>11.3</b>	0.7	10.9
CRF	3.0	<b>16.3</b>	<b>31.1</b>	10.7	<b>3.2</b>	<b>12.9</b>
CRF+E	6.8	9.9	3.2	3.6	1.4	5.0

Table 4.4: Pre-Fusion Hit-Rate of extracted facts with the identical training/testing set-up as above. NB+E and CRF have the best hit-rate.

known for certain how many times the document set contains the desired information.<sup>5</sup>

$$\text{Pre-Fusion Hit-Rate} = \frac{\# \text{ Correct Extracted Targets}}{\# \text{ People}}$$

The precision of each of the various extraction methods is listed in Table 4.3. The data show that on average the PCL method has the best precision, whereas the NB+E extractor has the worst. Training the CRF with negative exemplars (CRF+E) gives better precision in extracted information than training it without negative exemplars (CRF). Table 4.4 lists the hit-rate or average number of correctly extracted targets per person. The results illustrate that the PCL has the worst hit-rate and the CRF trained without negative examples has the best hit-rate. The presence of a few excellent patterns allows PCL to have high precision,

---

<sup>5</sup>It is insufficient to count all text matches as instances that the system should extract. To obtain the true recall, it is necessary to decide whether each sentence contains the desired relationship, even in cases where the information is not what the biographies have listed.

	Birthday	Birth year	Occupation	Birthplace	Year of Death	Avg.
PCL	<b>69.6</b>	33.2	<b>27.2</b>	<b>28.6</b>	<b>30.6</b>	<b>37.8</b>
NB+E	12.3	10.6	5.8	6.3	6.3	8.3
CRF	37.4	8.3	4.1	5.5	7.3	12.5
CRF+E	45.7	<b>40.0</b>	18.1	22.4	14.1	28.6

Table 4.5: Pre-Fusion Per-person Deduplicated Precision of extracted facts for the different extraction systems, trained on 15 people each with 150 documents, and tested on 137 people each with 150 documents. Just as with Pre-Fusion precision calculated above, PCL has the highest deduplicated precision.

	Birthday	Birth year	Occupation	Birthplace	Year of Death	Avg.
PCL	90.5	64.2	55.5	35.8	38.9	57.0
NB+E	<b>95.6</b>	96.4	98.5	<b>77.4</b>	77.8	89.1
CRF	78.1	<b>97.1</b>	<b>99.3</b>	75.2	<b>100</b>	<b>89.9</b>
CRF+E	94.2	94.9	78.1	67.9	94.4	85.9

Table 4.6: Pre-Fusion Per-person Deduplicated Recall of extracted facts with the identical training/testing set-up as above. Just as with Pre-Fusion Hit-Rate, the CRF has the highest per-person deduplicated recall.

but the cost is that these patterns are infrequent and thus the overall hit-rate is lower.

In addition to the above metrics, it is of interest as to system performance when duplicates have been removed. Two metrics can be computed. **Pre-fusion per-person deduplicated precision** gives the average number of correct vs. incorrect distinct targets retrieved by each system for each person. **Pre-fusion deduplicated recall** measures whether or not a system retrieved a correct target for each person. The performance of the systems with these metrics can be seen in Tables 4.5 and 4.6. These metrics yield the same system rankings as the versions above which measure performance with duplicates.

To test how the extraction precision changes as more documents are retrieved from the ranked results from Google, the above experiments were run over test sets of 1, 5, 15, 30, 75, and 150 documents per person. The number of people are kept the same, while the number of documents about them are increased.

<p> <b>Birthday Extractions</b> </p> <p> ... Angelina <b>Jolie</b> Date of Birth <b>June 4, 1975</b> ... </p> <p> ... Alyce Faye <b>Wattleton</b> Planned Parenthood Alyce Faye <b>Wattleton</b> was born <b>July 8...</b> </p> <p> Don't miss this ... event honoring Barry <b>Levinson</b> on <b>October 4</b> . </p> <hr/> <p> <b>Birthyear Extractions</b> </p> <p> ... Angelina <b>Jolie</b> ( <b>1975</b> ... </p> <p> Ava <b>Gardner</b> Type : Actress .. Date of Birth: .. Year of Birth:<b>1922</b> </p> <p> The Story of Bonnie and Clyde by Bonnie <b>Parker</b> ( <b>1934</b> ) </p> <hr/> <p> <b>Birthplace Extractions</b> </p> <p> <b>Quinn</b> was born in <b>Mexico</b> ... </p> <p> [Burt <b>Bacharach</b> grew up in <b>New York</b> </p> <p> Eric McCormick waits for a table , as do Halle Berry and Benjamin <b>Bratt</b> ( <b>Vancouver</b> Sun , Canada ) . </p> <hr/> <p> <b>Occupation Extractions</b> </p> <p> Annie <b>Potts</b> , the <b>Actor</b> ... </p> <p> Bill <b>Moyers</b> is the <b>author</b> of ... </p> <p> In this 1929 silent, .. classic, <b>Keaton</b> plays a <b>fellow</b> named Elmer ... </p>
---

Figure 4.2: Example Extractions of PCL from a 150 document test set. The first two examples in each section are correct, and the third is an example of an incorrect extraction. Out-of-vocabulary words are replaced with the token “.”.

<p>Birthday Extractions</p> <p>See David Byrne in concert live <b>David Byrne</b> was born in Dumbarton, Scotland on <b>May 14, 1952</b>.</p> <p>Charles Bronson <b>Charles Bronson</b> ( <b>November 3,1921</b> - August 30, 2003) was an American actor of “tough guy” roles.</p> <p><b>David O. Selznick</b> (Film, Biography) Encyclopedia AllRefer Channels :: Health — Yellow Pages — Reference — Weather <b>December 17 ...</b></p> <hr/> <p>Birthyear Extractions</p> <p><b>Anita Bryant</b> Type : Musician Quotes Date of Birth : 3/25/1940 Year of Birth : <b>1940</b></p> <p><b>Arthur Fiedler</b> was born in Boston on December 17, <b>1894</b> , his background deeply rooted in European musical tradition</p> <p>Revival, Play, Drama Written by <b>Athol Fugard</b> Jun 1, <b>2003</b> - July 13, 2003</p> <hr/> <p>Birthplace Extractions</p> <p>Theater Season For more complex or filtered searches, click here <b>Billy Joel</b> ( b. Mar 14, 1947 Long Beach, <b>NY</b>, USA) ...</p> <p><b>Brian Douglas Wilson</b> ( born June 20, 1942, in Hawthorne, <b>California</b>)</p> <p>... “Real Sports with <b>Bryant Gumbel</b>” is TV’s best newsmag, but the host spends more time on the golf course than in the studio ( <b>Minneapolis Star Tribune</b> )</p> <hr/> <p>Occupation Extractions</p> <p><b>Brent Spinner</b> (born February 2, 1949 in Houston Texas) is an American actor</p> <p><b>Carol Channing</b> : A Who2 Profile CAROL CHANNING &amp; # 8226 ; <b>Singer / Actor</b></p> <p><b>Parker, Charlie</b> Multimedia 1 item Parker, Charlie (1920-1955) American alto saxophone player, a <b>founder</b> of the bebop jazz style and one of the most influential musicians in the history of jazz.</p>
--

Figure 4.3: Example Extractions of CRF+E from a 150 document test set. The first two examples in each section are correct, and the third is an example of an incorrect extraction.

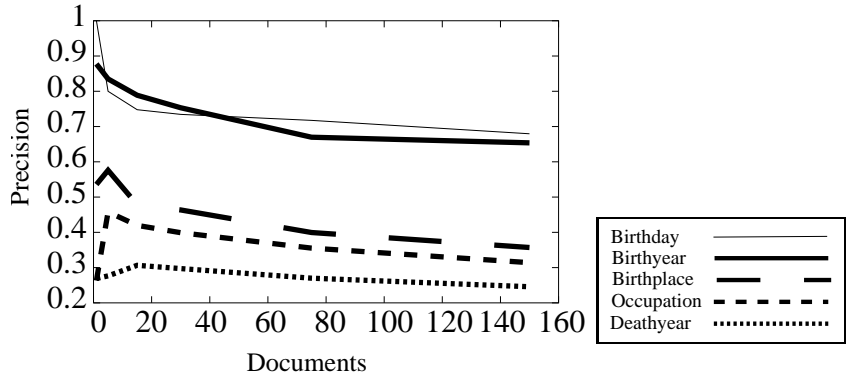


Figure 4.4: CRF+E Pre-Fusion Precision vs. Test Set Size. As test set size increases, pre-fusion per-extraction precision decreases.

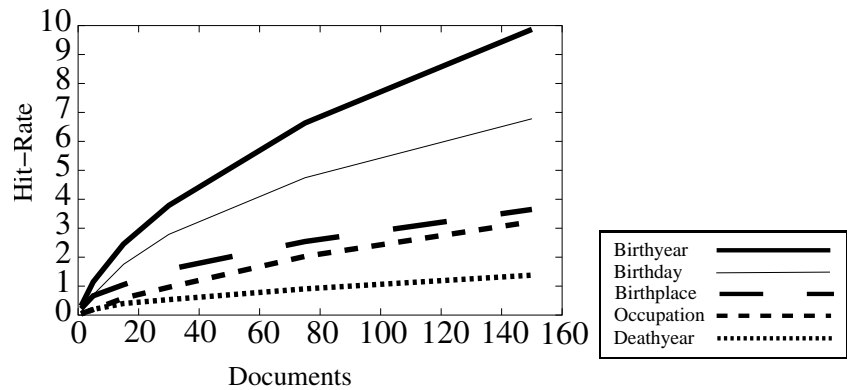


Figure 4.5: CRF+E Pre-Fusion Hit-Rate vs. Test Set Size. As test set size increases, pre-fusion hit-rate increases. Results on people still living were omitted from hit-rate results for year-of-death.

The data in Figure 4.4 suggest that there is a gradual drop in extraction precision throughout the corpus which may be caused by the fact that documents further down the retrieved list are less relevant and therefore less likely to contain the relevant biographic data.

Even though the extractor’s precision drops, the data in Figure 4.5 indicate that there continue to be instances of the relevant biographic data, as the hit-rate continues to increase.

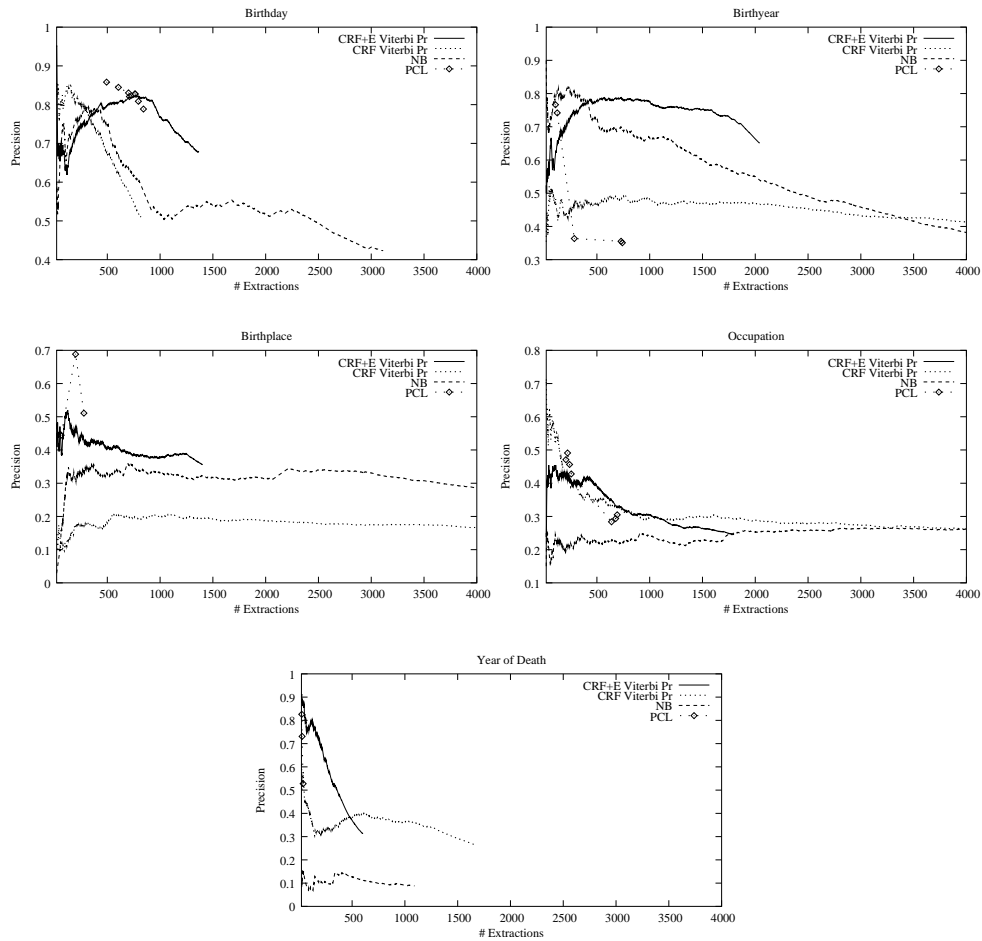


Figure 4.6: Precision vs. Number of Extractions for PCL, NB+E, CRF, CRF+E. PCL has a few very precise extractions, while CRF+E has a combined high precision along with a relatively large hit-rate.

As a final point of comparison, Figure 4.6 shows the precision of various extractors at various number of extractions per field. One of the most dramatic trends is that for the PCL, both the number of total extractions and the number of different points on the curve are very small. This results from relatively very few trained patterns being observed (e.g. 30 patterns for birthday). While the CRF trained with negative examples (CRF+E) shows the highest performance in general, in some cases, the CRF trained without negative examples (CRF) does better.

## 4.4 CRF Viterbi Sequence Extraction and Label Marginal Estimation

These experiments consider two alternative methods for using Conditional Random Fields for extraction. Both are trained in the same way, the difference comes in the method of application. In the first method, the Viterbi label sequence estimated by the CRF is taken to be the extraction from a given sentence. In the second, the confidence for each candidate to be the correct target is estimated using the Label Marginal computed by constrained forward-backwards (section 2.4.3). The prior experiments have demonstrated the performance of using the CRF Viterbi sequence for extracting facts. This section compares that performance with that of Label Marginal performance on per-fact extraction.

Figure 4.7 presents two example sentences, and shows for each candidate target, the Viterbi label sequence assignment and the probability assigned by the Label Marginal method. What the examples show is that the Label Marginal gives better back-off estimates than the Viterbi label sequence, which may be useful for improving recall.

Figure 4.8 shows the precision at various numbers of returned extractions for each fields. In the first curve, the extractions are returned by the Viterbi sequence and ranked by the probability of the Viterbi parse. In the second curve, extractions are returned by confidence estimate for each target given by the Label Marginal probability. The graph shows that the Viterbi is sometimes better than the label marginal probability estimate (in the cases of Birthplace, Occupation, and Year-of-Death), but that this increased precision is for a much smaller number of extractions than is obtained with the Label Marginal probability. The tails of the graph for the Label Marginal extractions have been removed

David **Hasselhoff** Birthname: David Michael Hasselhoff Height: 6' 4"  
Sex: M Nationality: American Birth Date: July 17, **1952** (*Label Marginal Probability .76, Viterbi Label = Target*) Birthplace: Baltimore, Maryland ...  
Wife: Pamela Bach **Hasselhoff** married in **1989** (*Label Marginal probability .0004, Viterbi Label = Background*) ... Mother Dolores **Hasselhoff** ...  
Daughter Haley Amber born in **1993** (*Label Marginal Probability .04,, Viterbi Label = Background*) ...

Jackie **Chan** ( born Kong-sang, as a tribute to his native Hong Kong ) can trace his origins to Hong Kong where he was born to parents Charles and Lee-Lee Chan on April 7, **1954**. (*Label Marginal Probability = 0.06, Viterbi Label = Spurious Target*)

Figure 4.7: Example of differences between Label Marginal and Viterbi Label Extraction Methods. Using label marginals instead of the Viterbi label sequence allows more candidates to be considered in each sentence. Sometimes, the correct target is already marked by the Viterbi label sequence, but often it is missed by the Viterbi label sequence but assigned a reasonably large probability by the underlying model.

for display, but extend for far longer (up to as much as 100,000 extractions). For example for birthday, there are 1,368 extractions retrieved by the Viterbi system and 24,978 from the Label Marginal<sup>6</sup>. The graphs show that the Label Marginal probability is a reasonably good method for ranking these extractions, though the strength of the Viterbi sequence suggests that noting that the candidate is present in the Viterbi labeling for the sentence might be advantageous.

## 4.5 Multi-Document Biographic Fact Extraction and Fusion

Per-extraction performance was presented in Section 4.3 but the final task is to find the single correct target for each person<sup>7</sup>.

---

<sup>6</sup>The Label Marginal method assigns a probability of being a target to all candidates (i.e. all strings for which  $A_R(y) = \text{true}$ ).

<sup>7</sup>This is a simplifying assumption, since there are many cases where there might exist multiple possible values, e.g. a person may be both a writer and a musician

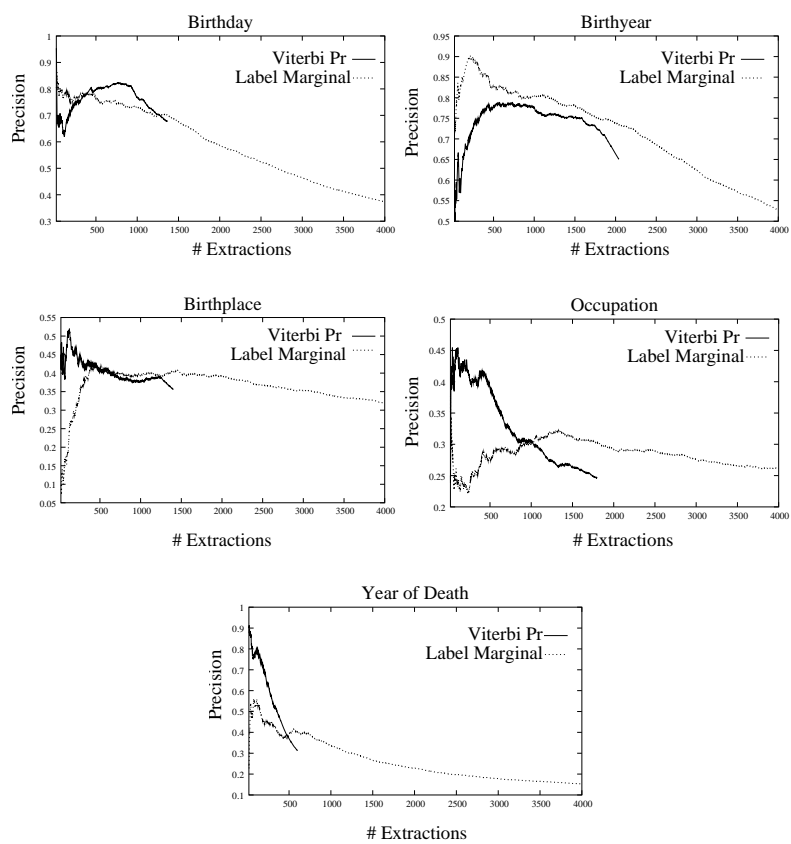


Figure 4.8: Precision vs. Number of Extractions. Though the Viterbi sequence in some cases has more precise extractions, the Label Marginal method always has significantly higher recall.

<b>PCL</b>	Phrase Conditional Likelihood (Section 2.4.1)
<b>NB+E</b>	Naïve Bayes Likelihood trained with negative examples (Section 2.4.2).
<b>CV</b>	Conditional Random Field Model, Viterbi Label Sequence (Section 2.4.3).
<b>CV+E</b>	CV with a spurious target model.
<b>CM</b>	CRF, Label Marginal
<b>CM+E</b>	CM with a spurious target model.

Figure 4.9: These six variant sentence extraction models are used in the remainder of the chapter.

Two main fusion scores are proposed. The first is a max:

$$F_r^{Max}(x, y) = \max_s C_r(x, y, s)$$

The second is a frequency measure which counts the number of times the target was extracted:

$$F_r^{Freq}(x, y) = |C_r(x, y, s) > 0|$$

The third is a weighted confidence measure:

$$F_r^{Weight}(x, y) = \sum_s C_r(x, y, s)$$

For CRF and CRF+E, there is the additional choice of whether the Viterbi label sequence probability is used, or the label marginal probability, and that distinction will be denoted as **CV(+E)** and **CM(+E)** respectively. Figure 4.9 gives the acronyms for each of the extraction methods used in this chapter and a short description.

The experimental setup used in the fusion experiments was the same as before: training on 15 people, and testing on 137 people. However, the grading post-fusion differs

from the pre-fusion grading. After fusion, the system returns one consensus target for each person and thus the grading should be on the **accuracy** of those targets. That is, missing targets are graded as wrong<sup>8</sup>. This is a harsh judgment because missing answers may not be as bad as wrong answers.

$$Post-Fusion Accuracy = \frac{\# \text{ People with Correct Target}}{\# \text{ People}}$$

In addition, the mean reciprocal rank (MRR)<sup>9</sup> is shown. With low accuracy extractors or for difficult relations, it is sometimes helpful to see the MRR scores in order to compare performance.

#### 4.5.1 Comparison of Fusion Methods

The first set of experiments looked at average performance on target extraction across all five fields with the various extractors and fusion methods. The data in Table 4.7 show the average system performance with the different fusion methods. Frequency voting gives anywhere from a 2% to a 20% improvement over picking the highest confidence candidate. CM+E Weighted (the CRF trained with negative exemplars using weighted confidence of label marginals) is the highest performing system overall.

For the remainder of the chapter, only the best performing fusion method for each sentence extractor is shown (i.e. Freq for PCL).

Figures 4.10 and 4.11 present examples of fusion for each of the extractors. Table

---

<sup>8</sup>For year of death, only cases where the person had died were graded. This is a lenient judgment because ideally the system should return “not-dead-yet” as appropriate. See [Feng and Hovy, 2005] for a possible solution to this.

<sup>9</sup>The reciprocal rank = 1 / the rank of the correct target.

	Best	Freq	Weighted
PCL	0.364	0.450	
NB+E	0.385	0.588	
CV	0.513	0.625	
CV+E	0.65	0.677	
CM	0.289		0.648
CM+E	0.412		<b>0.744</b>

Table 4.7: Average Accuracy of the Highest Confidence (Best) and Most Frequent (Vote) across five extraction fields. The CRF trained with both positive and negative examples, with output by label marginals, and which performs fusion by weighted confidence is the most successful extractor. Training using negative examples gives a 5% increase for the Viterbi frequency fusion methods, and a 10% increase for the weighted confidence fusion.

4.8 shows the results of using each of these extractors to extract correct relationships from the top 150 ranked documents downloaded from the Web. CM+E is the top performer for all relationships except for occupation in which the NB+E does the best. Both qualitatively from the examples, and quantitatively from the results, it can be seen that the poor recall of PCL is a severe disadvantage, from which its improved precision cannot recover. Even though it isn't precise, the raw frequency co-occurrence of the correct target with a hook allow the Naïve Bayes extractor to perform quite well.

### Confidence Thresholds and Fusion Accuracy

In the previous section, for the frequency methods (PCL, NB, CV, CV+E) a threshold manually tuned on the development data was used as a cut-off for judging whether or not a particular extracted fact was accepted. This section examines the effect of that threshold on fusion accuracy, in a post-fusion parallel of the pre-fusion experiments in at the end of Section 4.3.

Figure 4.12 shows the fusion accuracy across different recall thresholds. Recall is

Birthday		
	Fusion Accuracy	Fusion MRR
PCL	0.854	0.877
NB+E	0.854	0.889
CV	0.650	0.703
CM	0.810	0.861
CV+E	<b>0.883</b>	0.911
CM+E	<b>0.883</b>	<b>0.913</b>
Birthyear		
	Fusion Accuracy	Fusion MRR
PCL	0.387	0.500
NB+E	0.774	0.838
CV	0.796	0.86
CM	0.679	0.782
CV+E	0.861	0.894
CM+E	<b>0.883</b>	<b>0.925</b>
Occupation		
	Fusion Accuracy	Fusion MRR
PCL	0.299	0.405
NB+E	<b>0.642</b>	<b>0.751</b>
CV	0.606	0.74
CM	0.613	0.749
CV+E	0.416	0.552
CM+E	0.577	0.725
Birthplace		
	Fusion Accuracy	Fusion MRR
PCL	0.321	0.338
NB+E	0.474	0.586
CV	0.321	0.476
CM	0.416	0.540
CV+E	0.467	0.560
CM+E	<b>0.518</b>	<b>0.621</b>
Year-of-Death		
	Fusion Accuracy	Fusion MRR
PCL	0.389	0.389
NB+E	0.194	0.383
CV	0.750	0.840
CM	0.722	0.822
CV+E	0.750	0.827
CM+E	<b>0.861</b>	<b>0.906</b>

Table 4.8: Fusion Accuracy and Fusion MRR for biographic fact fields across Multiple Extraction Methods. CM+E, the Conditional Random Field with negative examples and confidence weighted fusion performs the best.

Birthday – Henry Fonda			
Candidate	PCL Frequency	Candidate	Naïve Bayes Frequency
<b>May 16</b>	4	<b>May 16</b>	9
December 3	1	May 26	6
	1	December 16	5
	1	May 18	4
		September 29	3
		September 23	3
		July 30	3
		December 19	3
		August 13	3
		September 11	2

Birthplace – Farah Fawcett			
Candidate	PCL Frequency	Candidate	Naïve Bayes Frequency
		<b>Texas</b>	18
		tx	5
		New York	3
		Lee	3
		Wa	2
		Rwanda	2
		Logan	2
		Alabama	2
		See	1
		Pittsburgh	1

Occupation – Garth Brooks			
Candidate	PCL Frequency	Candidate	Naïve Bayes Frequency
<b>Performer</b>	1	<b>Artist</b>	46
		<b>Singer</b>	10
		Challenger	6
		Reviewer	5
		Producer	5
		<b>Entertainer</b>	5
		<b>Musician</b>	4
		Author	3
		<b>Performer</b>	2
		Matador	2

Figure 4.10: Candidates ranked by PCL and Naïve Bayes for Fusion of extracted Birthday, Birthplace and Occupation facts. Candidates in **bold** are correct.

Birthday – Henry Fonda			
Candidate	Viterbi Frequency	Candidate	Weighted Confidence Estimate
<b>May 16</b>	8	<b>May 16</b>	6.15
August 12	2	August 12	1.78
June 16	1	December 19	0.95
December 19	1	June 16	0.81
		June 6	0.29
		February 17	0.20
		December 3	0.08
		July 29	0.04
		November 22	0.03
		January 1	0.02

Birthplace – Farah Fawcett			
Candidate	Viterbi Frequency	Candidate	Weighted Confidence Estimate
Logan	2	<b>Texas</b>	2.40
Lee	2	Logan	1.37
<b>Texas</b>	1	Lee	1.16
Florida	1	Austin	0.62
Austin	1	Jackson	0.40
		Lea	0.34
		Florida	0.29
		Nevada	0.21
		Maryland	0.16
		See	0.15

Occupation – Garth Brooks			
Candidate	Viterbi Frequency	Candidate	Weighted Confidence Estimate
Mason	1	<b>Artist</b>	2.64
		<b>Performer</b>	2.07
		<b>Singer</b>	1.77
		Author	1.67
		Investor	0.99
		<b>Musician</b>	0.82
		Actor	0.81
		Sponsor	0.66
		<b>Vocalist</b>	0.58
		Mason	0.56

Figure 4.11: Candidates ranked by Viterbi Frequency and Weighted Confidence Estimation for Fusion of extracted Birthday, Birthplace and Occupation facts. Candidates in **bold** are correct.

calculated over the number of extractions retrieved by each extractor and corresponds to a particular cut-off for selecting extractions for the purposes of frequency-based agglomeration. One of the striking aspects of the data is that the curves are relatively flat for CV+E (in particular for birthday, birthyear and occupation), which suggests that the method is relatively robust to the choice of a cut-off parameter. In general, higher levels of recall lead to improved fusion accuracy, and this suggests that a decent choice for a threshold without using any development data would be to simply take all extractions. This may be true due the fact that increased noise at higher levels of recall is likely to be uncorrelated and thus does not affect system performance.

### **Error Analysis**

The system so far described is a complicated pipeline of operations. Initially, pages are downloaded from the Web, cleaned, and run through a part-of-speech tagger. In training, a set of facts are used to automatically mark up documents which mention those facts, producing a set of example sentences, and those sentences then are used to train a statistical model. In testing, that model is used to extract information from a series of documents, where the information is then fused and a consensus answer is presented to the user.

At each of these steps, there is room for error. The wrong pages can be downloaded, either by fault of the automatic process in which the download requests are made, or alternatively because the name is polyreferent and pages belonging to a different person are downloaded. The various segmentation and tagging operations are highly effective though relatively brittle. In particular, the role of the automatic annotation in causing errors and

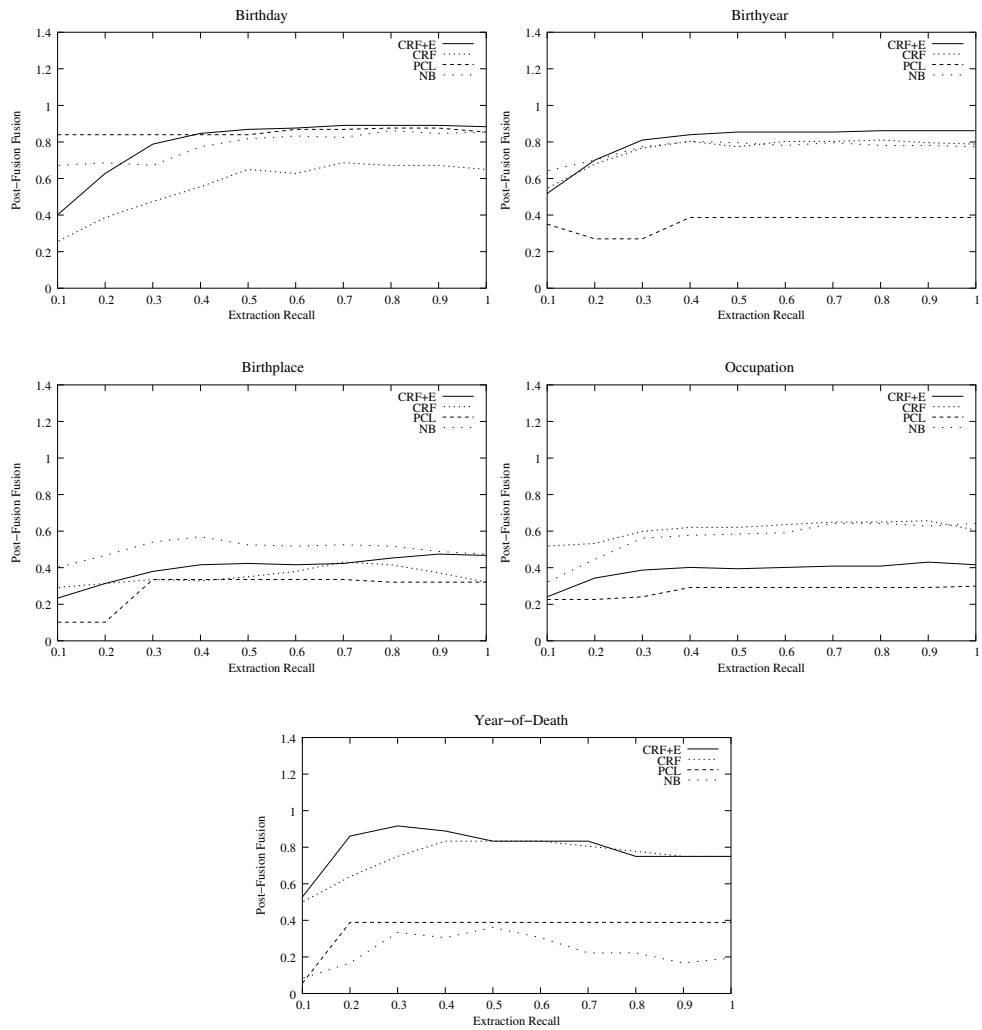


Figure 4.12: Precision vs. Number of Extractions for Post-Fusion Accuracy. CRF+E curves dominate for Birthday, Birthyear and Year-of-death, but for Birthplace, CRF performs slightly better at most points in the curve. For Occupation, NB and CRF perform about equally.

the relative merits and flaws of the extraction systems deserve a full accounting.

The process of automatic annotation leads to a set of errors which are somewhat novel. First, if the base information is incorrect, the marked up data will be incorrect. The facts used in the first part of the study (those dealing with biographical, non-relationship information) were taken from `www.infoplease.com` and there was no vetting or analysis of the facts to determine whether or not they were accurate. In one case, the experiments showed the information to be incorrect (Gwyneth Paltrow's birthday was listed as September 27, when most sources list it as September 28). As mentioned previously, the error of omission has additional consequences. For some fields like occupation, there are multiple possible correct ways to state the profession and missing one variant leads to many sentences marked as having a false target which are in fact correct (false negatives in training mode).

Automatic annotation errors are harder to categorize. These occur when a human would not mark a sentence as containing the fact, even though the sentence contains a hook and target. For example, the sentence

“December 6, 2004 - Angelina Jolie attended the premiere of 'Hotel Rwanda' at the Academy Theater in Los Angeles on December 2nd .”

does not imply that Angelina Jolie's birthplace is Los Angeles. However, when the system is automatically marking up the text, and is given the information that Angelina Jolie's birthplace is in fact Los Angeles, it will mark up this sentence as attesting this fact. It is precisely this kind of false positive that distinguishes human annotation from computer annotation. By training on a large corpus, the hope is that these errors will be drowned out by predominantly correct mark up, and those error contexts which are present will tend

to be uncorrelated noise, while the true positives will tend to exhibit a more concentrated signal.

While individual extractions are rife with errors, the extraction procedure becomes much more robust with respect to extraction errors after fusion. False positives (where an incorrect target is extracted) have a relatively small effect, since presumably the false positives will be uncorrelated, and thus will be overwhelmed by the true answer. False negatives (where a correct target isn't correctly extracted from a sentence) have the potential to lead to lower system performance since if the correct target isn't retrieved from any sentence the system cannot recover. Thus there is an asymmetry between false positives and false negatives (a precision/recall trade-off), so that it's better to collect more, noisier information than select a few answers which are known to be correct.

Part of the reason for the success of the weighted confidence scheme is that its recall is better than the Viterbi labeling. In cases where the Viterbi labeling doesn't return any candidates, the weighted confidence method gives weightings to all potential candidates, even if any particular case doesn't meet the threshold for extraction in one particular example. These additional erroneous candidates do not hurt for two reasons. First their confidence estimates on the whole are low, and second they are typically not distinctively peaked. That is, there are many candidates each of which has a very small probability. Another improvement comes in that weighted confidence estimates help resolve ties which happen occasionally in Viterbi extraction.

Another reason which became clear through the error analysis is that for birth-place, due to infrastructure improvements, the marginal label method had better agglom-

**Miles Davis** Biography : Jazz legend **Miles Davis** was born on May 25 , 1926 in Alton , Ill . to a middle-class ... MORE  
Buy CDs Add our **Miles Davis** XML feed to your blog ,  
website , or newsreader .

May 26, 1988 : **Davis**' autobiography says Prince put in an  
appearance at **Davis**' 62nd birthday party .

Figure 4.13: In Web data, often there are multiple hooks and targets, or spurious targets in a given automatically segmented “sentence”. Determining what to annotate as the hook and target or spurious target has an impact on performance.

eration of place names as compared to other methods and that improved agglomeration accounts for some of the improvement<sup>10</sup>. However, this benefit does not hold for the other fields and the improvement must come as well from the reasons above.

## 4.5.2 Training-Data Annotation Variants

One of the details in preparing the annotated data for training is the exact method for marking up the sentences. In particular, when there are a number of hooks in a particular sentence, choosing which hook to attach to which target (or spurious target) can be complicated, and as it turns out, important (see Figure 4.13 for an example). The AT&T finite-state machine library [Mohri et al., 2003] provided the backbone for training data annotation.

The initial model, and the model used for the majority of the experiments, is depicted in Figure 4.14 (A). In this finite state model, an arbitrary choice is made between the possible hook and target combinations, using the BestPath library call in the library. The next model chooses the hook and target/spurious target which are closest to each

---

<sup>10</sup>The CM+E was able to correctly segment “Las Vegas” as one place name, where earlier systems (occasionally) picked out “Vegas”, which would be graded as incorrect. This bug, which occurred only in the case of birthplace extraction, was discovered too late to correct.

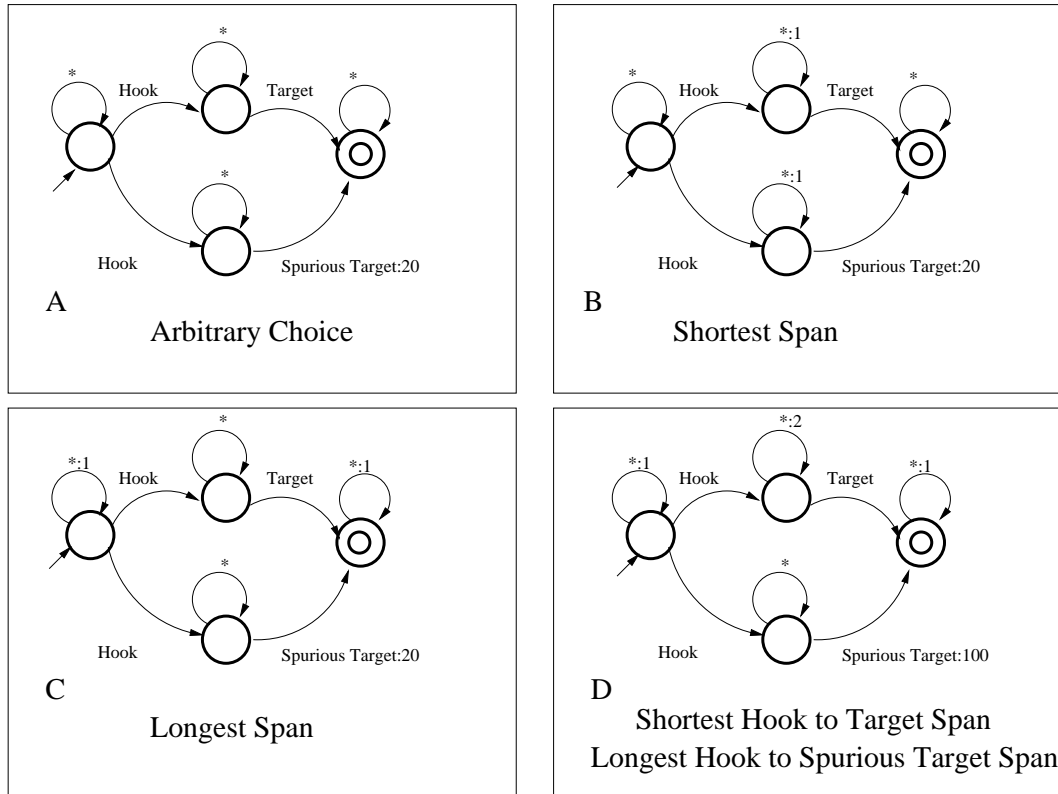


Figure 4.14: Hook and Target/Spurious Target markup strategies for when multiple hooks and targets/spurious targets appear in the same sentence.

other (Figure 4.14 (B)). This was produced by taking the BestPath over a finite-state machine where each interstitial word gets weight of 1 (BestPath looks for the path with the minimal weight). The next model picks the hook target/spurious target pair with the longest interstitial gap between them, as given in (Figure 4.14 (C)). The final method chooses the closest hook and target if the correct target is present in the sentence, and otherwise the hook/spurious target which are farthest apart (Figure 4.14 (D)).

The results indicate that the arbitrary strategy initially examined is not the best method for producing annotated sentences, and strategies which make more principled choices can improve performance. Out of the training annotation schemes, the best per-

	Birthday	Birthyear	Birthplace	Occupation	Year of Death	Avg.
Arbitrary	0.883	<b>0.869</b>	0.467	0.416	0.750	0.677
Longest-Span	0.832	0.839	0.526	0.606	<b>0.833</b>	0.727
Shortest-Span	<b>0.891</b>	0.825	0.409	<b>0.730</b>	0.806	0.732
Shortest-Target Longest Spurious	0.861	0.861	<b>0.555</b>	0.693	0.806	<b>0.755</b>

Table 4.9: Performance of various methods for Automatic Annotation as shown in Figure 4.14. The best strategy annotates the closest hook and target pairs, and the longest hook and spurious target pairs.

former appears to be the method which labels the shortest distances between hook and target and the longest spans between hook and spurious target. One possible reason for this is that it chooses the least noisy training data for the sequence which needs to be high precision (hook-target) and the most noisy for the sequence which functions in many ways as the background. This would collect more examples of “noise” unigrams which would help in smoothing. This analysis leaves unexplored the possibility of taking more than example from a given sentence.

### 4.5.3 Order-Invariant Models

The previous models have enforced an assumption that the hook precedes the target. This assumption is not necessary, and an obvious question arises as to whether performance would be improved with the relaxation of this constraint. In this section, the CV+E and CM+E models which have this constraint removed are explored with the expectation that the results will carry-over to the lower performing systems.

In the order invariant models, the state space must be expanded. Instead of tagging any interstitial sequence no matter whether the hook or target is first, tag sequences  $pA_2q$

and  $qA_3p$  must be tagged separately. If instead sequences were tagged  $pA_2q$  and  $qA_2p$ , using the same state for both orderings, the models could then mark up new sentences as  $pA_2p$ .

Figures 4.10 and 4.11 show the performance of the order-invariant (Shortest-Span) extractors compared to the Shortest-Span one-sided performance. The results suggest that for this set of relationships, enforcing a fixed order of hook followed by target improves performance. This may be caused by the noise of training data, where sentences which appear to encode a relationship in “target hook” order are often noise.

	Birthday	
	Fusion Accuracy	Fusion MRR
O-CV+E	0.679	0.696
CV+E	0.883	0.911
CM+E	0.883	0.913
O-CM+E	0.854	0.896
	Birthyear	
	Fusion Accuracy	Fusion MRR
CV+E	0.869	0.901
O-CV+E	0.511	0.556
CM+E	0.883	0.925
O-CM+E	0.803	0.867
	Occupation	
	Fusion Accuracy	Fusion MRR
CV+E	0.416	0.552
O-CV+E	0.197	0.212
CM+E	0.577	0.725
O-CM+E	0.613	0.735
	Birthplace	
	Fusion Accuracy	Fusion MRR
CV+E	0.467	0.560
O-CV+E	0.321	0.365
CM+E	0.518	0.621
O-CM+E	0.438	0.573
	Year-of-Death	
	Fusion Accuracy	Fusion MRR
CV+E	0.750	0.827
O-CV+E	0.250	0.400
CM+E	0.861	0.906
O-CM+E	0.333	0.560

Table 4.10: Models which expect a relation in a specific order (hook followed by target) are compared with models which accept either order. In general, the former perform better.

	Average Accuracy	Average MRR
CV+E	0.677	0.750
O-CV+E	0.392	0.446
CM+E	0.744	0.818
O-CM+E	0.608	0.726

Table 4.11: Average Order-Invariant Model Performance. On average, the models which predict a hook followed by a target perform better than those which accept either order.

	Spouse	Father	Son	Mother	Daughter	R-Avg	School	Avg
PCL	0.4	0	0.143	0	0	0.109	0.286	0.138
NB+E	0	0	0.143	0	0.2	0.069	0.714	0.176
CV	0	0	0.143	0	0	0.029	0	0.024
CM	0	0	0	0	0.2	0.04	0.571	0.129
CV+E	0.2	0	0.143	0.1	0	0.089	0	0.074
CM+E	0.3	0	0	0.2	0.4	0.18	0.714	0.269

Table 4.12: Familial Relationship Extraction – Fusion Accuracy.

	Spouse	Mother	Father	Son	Daughter	R-avg	School	Avg
PCL	0.4	0.1	0.152	0	0	0.13	0.286	0.156
NB+E	0.071	0.011	0.223	0.103	0.313	0.144	0.743	0.244
CV	0.135	0.083	0.188	0.113	0	0.104	0	0.086
CM	0.097	0.068	0.056	0.101	0.407	0.146	0.667	0.233
CV+E	0.283	0	0.159	0.15	0	0.118	0.071	0.111
CM+E	0.356	0.086	0.025	0.311	0.468	0.249	0.762	0.335

Table 4.13: Familial Relationship Extraction – Fusion MRR. On these data, the CM+E method works best on average.

#### 4.5.4 Familial Relationship Extraction and Fusion

In addition to the relationships detailed above, 6 other attributes of people were examined, college attended, and 5 nuclear family relations: spouse, mother, father, son, and daughter. These were not present in the mini-biographies listed above, but instead were manually discovered by research on Web pages.

Tables 4.12 and 4.13 display system performance at retrieving the other attributes. “r-avg” is the average system performance on the nuclear family relationships, and “avg” is the system performance averaged across all of the relations. The results corroborate what prior experiments have shown – the best model is CM+E, the CRF with weighted fusion over label marginals. The results also validate earlier results on the utility of training with positive and negative examples, where models with a negative exemplars out-perform the models trained without negative exemplars.

## Error Analysis for Relationships

It is also relevant to note the drop in performance with the relationship extraction in comparison with performance for the other attributes. Three causes seem to be preeminent.

First, the relationships are attested much more infrequently than the other attributes. For example, Gloria Steinem's father's name is Leo. The token Leo appears 9 times in the 150 document set, out of which 6 times the token actually refers to her father, and only 4 times is it co-sentential with a referent to Gloria Steinem, so unlike birth year which appears in appropriate contexts around 37 times, here the system has only 4 legitimate chances to extract the correct answer.

Second, there are more out-of-domain targets than there are for the other attributes. From all of the target names, 33/120 were not present in the list of names. While this doesn't necessarily translate to an upper bound of 72% since correctness is judged on simply getting one correct relationship from the set (e.g. finding one son), it shows that there is a big difference from the previous cases, where nearly all of the targets were recovered by the model.

Third, the targets are not necessarily the most frequent elements in their target set in a particular hook corpus. With the previous models, the frequency alone of the correct target allows even poor extractors to produce reliable results after fusion, as the pressure to model interstitial text is lower. An individual's birth day is often the most frequent day that co-occurs with that individual in Web pages. In contrast, in the Jackson Pollack retrieved document set Andy Warhol (a contemporary) appears 17 times, while Jackson

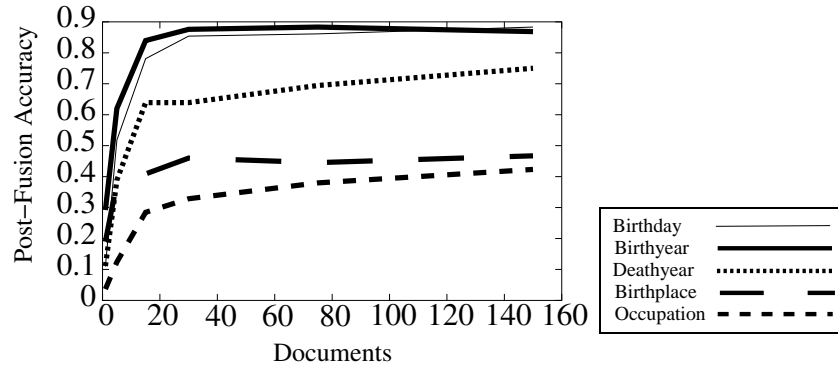


Figure 4.15: CV+E Post-Fusion Accuracy vs. Test Set Size. As test set size increases, post-fusion accuracy increases. Thus, even while precision of extracted facts decreases (Figure 4.4), the increased recall allows for improved performance (Figure 4.5).

Pollack’s father, LeRoy Pollack, appears only 3.

#### 4.5.5 Test Set Size and Performance

As with pre-fusion, a set of experiments with different test set sizes were performed all of which used the CV+E extraction system trained on 150 documents. The data in Figure 4.15 show that post-fusion accuracy improves as the test set size increases. Most of the gains come in the first 30 documents, where average field accuracy increases from 14% (1 document) to 63% (30 documents). Increasing the test set size to 150 documents yields an additional 5% absolute improvement.

Post-fusion errors came from two major sources. The first source is the misranking of correct relationships. The second is the case where relevant information is not retrieved at all.

$$Post-Fusion\ Missing = \frac{\# \text{ Missing Targets}}{\# \text{ People}}$$

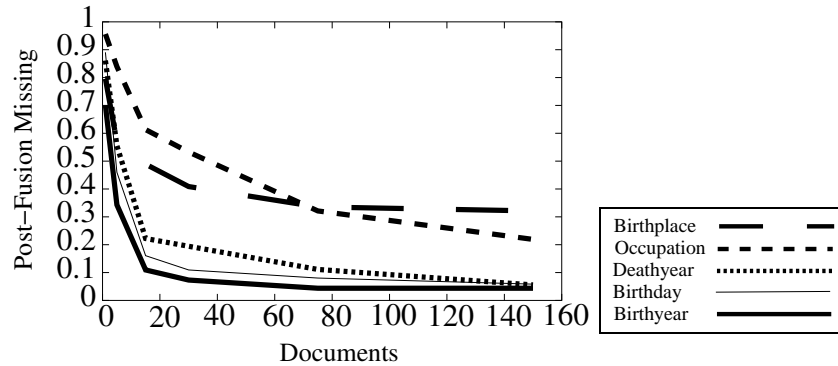


Figure 4.16: CV+E Post-Fusion Missing Values vs. Test Set Size. Part of the improved performance due to recall comes in the discovery of facts in later documents which are not present in the small test sets.

The data in Figure 4.16 suggest that the decrease in missing targets is significant contributing factor to the improvement in performance with increased document size. It may be a continuing bottleneck for birthplace, as more than half the errors (32% at 150 documents) come from missing targets.

Along with the above comparison of performance of CV+E against corpus size, another question is how the extractors perform with respect to each other as test set size increases. These results show that the trend over all extractors is the same as with CV+E.

#### 4.5.6 Training Set Size and Performance

One of the results from Section 4.3 is that lower ranked documents are less likely to contain the relevant biographic information. While this does not have an dramatic effect on the post-fusion accuracy (which grows with more documents), it suggests that training on a smaller corpus, with more relevant documents and more sentences with the desired information, might lead to equivalent or improved performance. A final set of experiments looked at system performance when the extractor is trained on fewer than 150 documents.

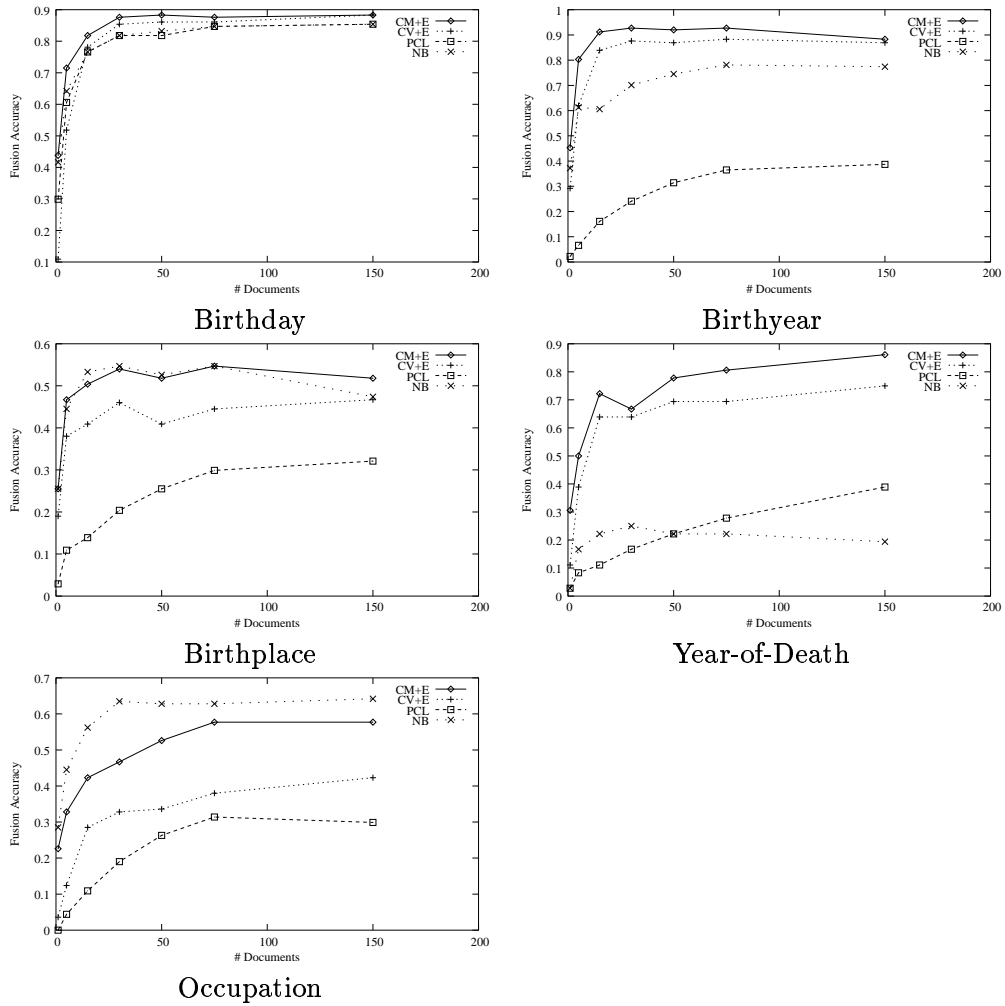


Figure 4.17: Post-Fusion Accuracy vs. Test Set Size across different extractors, CM+E, CV+E, PCL, and NB. The increased performance of post-fusion accuracy due to larger test set sizes is consistent across each of the extractors and the fields.

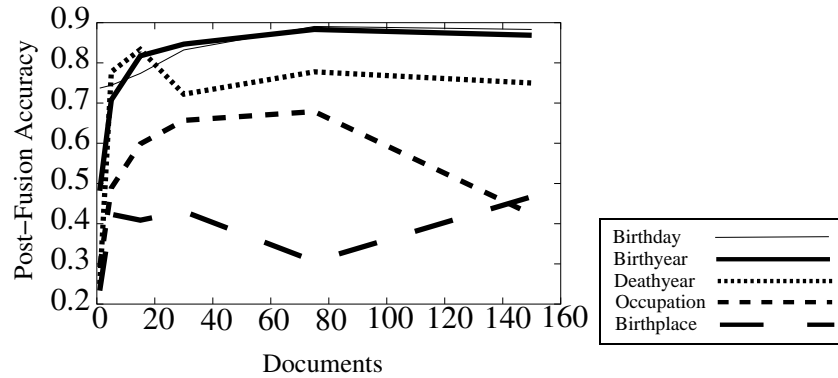


Figure 4.18: CV+E Post-Fusion Accuracy vs. Training Set Size. As with increasing the test set size, increasing the training set size leads to gains in performance. These gains are more erratic, suggesting that later documents have some negative effect in training, and may be introducing more noise into the training process.

The data in Figure 4.18 show that reducing the training set to 30 documents per person yields around the same post-fusion accuracy as a set of 150 documents per person. Average accuracy when the system was trained on 30 documents is 70%, while average accuracy when trained on 150 documents is 68%. Most of this loss in performance comes from losses in occupation, but the other relationships have either little or no gain from training on additional documents. There are two reasons why more training data may not help, and even may hurt performance.

One possibility is that higher ranked retrieved documents are more likely to contain biographical facts, while in later documents it is more likely that automatically annotated training instances are in fact false positives. That is, higher ranked documents are cleaner training data. Pre-Fusion precision results (Figure 4.19) support this hypothesis since it appears that later instances are contaminating earlier models.

The data in Figure 4.20 suggest an alternate possibility: the hit-rate decreases and increases erratically. This suggests that the selection thresholds are brittle. Thus for some

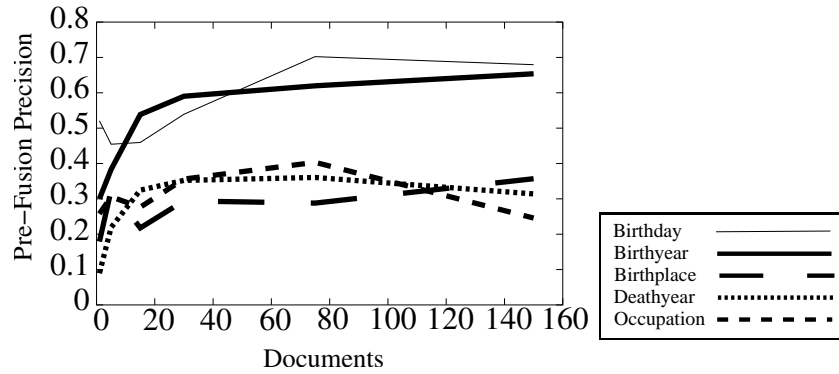


Figure 4.19: CV+E Pre-Fusion Precision vs. Training Set Size. As the training set size increases, the precision of the extractors generally increases, and then drops at the 150 documents. Presumably, more data yields better extractors, but as the document set gets larger, later instances contaminate the extractors and lead to lower precision.

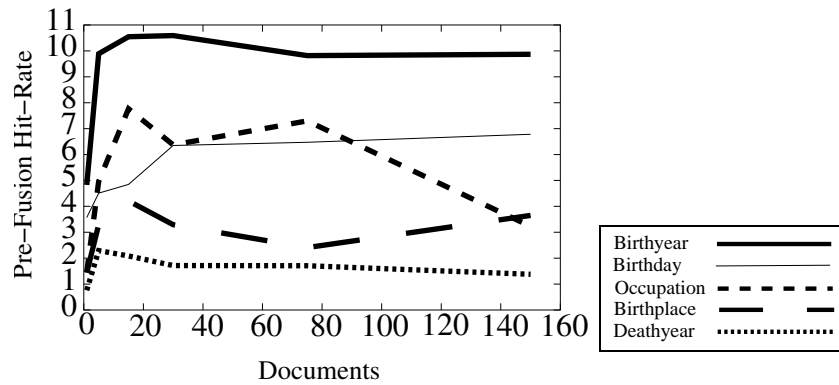


Figure 4.20: CV+E Pre-fusion Hit-Rate vs. Training Set Size. As the training set size increases, the recall decreases and increases erratically. This suggests that the recall thresholds are brittle.

training conditions, a reduced a hit-rate leads to lower post-fusion accuracy.

### More Names or More Pages?

One additional question is whether there is more benefit from adding more people to the training set or collecting more pages. Prior experiments have suggested that performance levels off at around 150 training data pages. For the next set of experiments, the number of training pages is fixed at 750 pages, and composed either of 150 pages for 5

	5 people, 150 pages	15 people, 50 pages
Birthday	78.1	86.1
Birthyear	35.0	86.9
Birthplace	34.3	40.8
Year-of-Death	63.9	69.4
Occupation	47.4	33.6
Avg.	51.74	63.36

Table 4.14: Post-Fusion Accuracy for training on 750 pages, either with 5 people and 150 pages or 15 people and 50 pages, with CV+E system. For the same number of pages, it is better to have them come from different people than to take pages further down the ranked document list.

people or 50 pages for 15 people.

The results in Table 4.14 suggest that having more people (example facts) may be more beneficial than more pages, as using more people giving an average benefit of 10%, where all relations show improvement with the exception of occupation where performance drops. There are two potential reasons for the improved performance. First, it is possible that pages further down the ranked list are less valuable. In support of this hypothesis, earlier experiments have shown that the precision of the extractors drops as the testing set size increases. A second possibility is that having more names allows for more genre diversity for training. It is likely that both of the effects have some part in the change in performance. Because of the large increase from using more people, it prompts the question of whether bootstrapping would lead to higher performance.

## 4.6 Bootstrapping

The prior section looked at methods which take a small set of facts, download pages, trains a model from the downloaded pages, and then extracts more facts. However,

in this context, a bootstrapping method (e.g. [Yarowsky, 1995]) might be applied. Once an initial model is trained from a seed set of facts, more facts could be extracted, and those facts in turn used to mark up data and build a better model. The improved model could then be used to extract the remaining unselected facts.

The most closely related work in fact extraction via bootstrapping comes in [Jones et al., 1999, Agichtein and Gravano, 2000]. In those papers, a set of information extraction patterns is provided, and the goal is to find appropriate patterns and facts (either semantic tags in the former or tuples in the latter) by the following procedure.

1. Start with a small seed set.
2. Learn a set of information extraction patterns from those seeds.
3. Use the information extraction patterns to extract new seeds.
4. Go to step 2.

In the above work, heuristic measures for selecting appropriate patterns and facts are used.

In particular [Jones et al., 1999] picks good patterns by:

$$score(pattern_i) = R_i \times \log_2(F_i)$$

where  $F_i$  is the number of known target facts which were extracted,  $N_i$  is the number of facts (known and unknown) that were extracted, and  $R_i = \frac{F_i}{N_i}$ . Good facts  $t$  are then chosen

as:

$$score(t) = \sum_{k=1}^{P_t} 1 + (.01 \times score(pattern_k))$$

where  $P_t$  is the number of patterns that pick  $t$  as a fact, and  $pattern_1..pattern_k$  are those patterns. These heuristics attempt to iteratively pick out good patterns and good facts.

The highest performing sentence extractors CV+E and CM+E abandon the phrasal extraction patterns used in PCL. Therefore the above bootstrapping heuristics will not work in their present form (since it isn't possible to compute the score for a particular pattern). However, there are a number of alternatives to the above heuristics in this case.

One option is to use the frequency information provided by the CV+E method which has shown to be effective for cross-document fusion. In this case, all of the potential facts to add would be ranked by their frequency of extraction and the facts with the highest frequency would be added to the training set for bootstrapping.

$$R^{Freq}(x, y) = F_{cf}^{Freq}(x, y)$$

Another option is to use the weighted confidence estimate method with the CM+E method:

$$R^{Weight}(x, y) = F_{cf}^{Weight}(x, y)$$

Given these two metrics, a primary question is how well they each do at ranking in isolation to see how well they would pick out correct answers from incorrect ones. Figure 4.21 plots the precision vs. recall of the various confidence measures using the **Shortest Target, Longest Span** training method. In the figure, the recall is computed over the post-fusion fused extractions (i.e. for each person, their fused answer is assigned a confidence

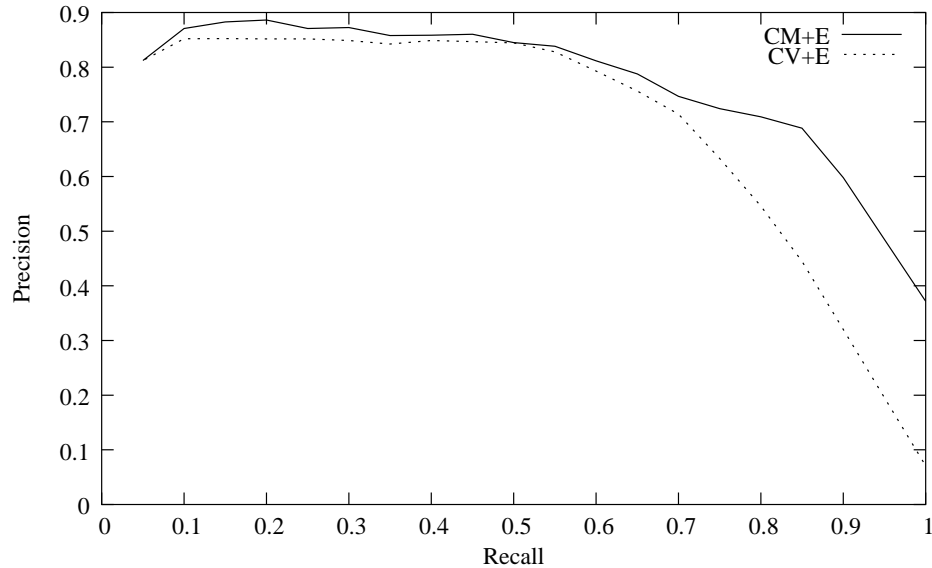


Figure 4.21: Precision/Recall Curves for CV+E and CM+E Ranking Methods. CM+E weighted confidence gives better precision at all recall levels than CV+E Viterbi frequency.

weight and these confidence weights are ordered and ranked.). As seen in Figure 4.21, the highest performance measure based on the precision recall graphs is CM+E with weight confidence fusion (as it dominates the CV+E curve), so we pursued experiments which performed bootstrapping using that measure as a heuristic for seed selection.

At each iteration in bootstrapping, the system adds the top 5 facts as ranked by the above statistics, mimicking the procedure advocated by prior research [Jones et al., 1999]. Only five iterations of bootstrapping were run as the computational cost at each iteration is considerable. An alternative method would be to have used a development set to estimate the appropriate number of bootstrapping iterations to use.

#### 4.6.1 Experimental Results

Figure 4.22 and Table 4.15 show the results of bootstrapping. As can be observed, each field has a distinctly different performance change, with the largest improvement com-

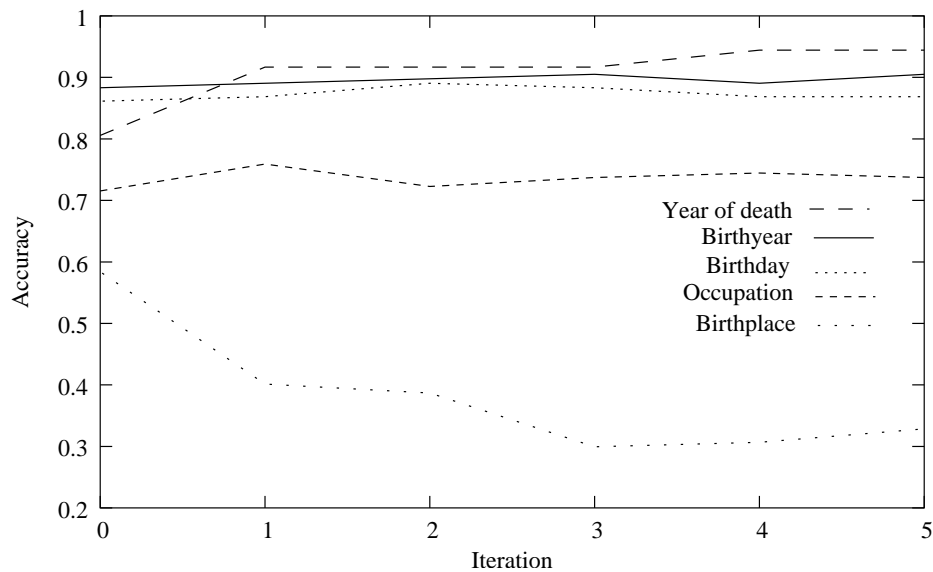


Figure 4.22: Precision vs. Recall by Fusion score averaged over all 5 fields. For all fields except birthplace, there is a slight gain in performance by bootstrapping.

Field	Initial	After 5 Iterations
Birthday	88.3	90.5
Birthyear	86.1	86.8
Birthplace	58.4	32.8
Occupation	71.5	73.7
Year-of-Death	80.5	94.4

Table 4.15: Bootstrapping Results : After 5 iterations there is a slight gain in all fields except birthplace, which had the lowest starting performance.

ing in the Year-of-Death extraction and the only drop in performance coming from Birthplace. One possible way to explain these differences is that for Birthplace, the new examples are too noisy to give reliable additional information (supported by the fact that birthplace has the lowest rate of extraction among all of the fields). In contrast, Year-of-Death is able to be reliably extracted, but in general suffers from a lack of training data (as it is infrequently mentioned).

The question remains as to why there wasn't a larger gain in performance from bootstrapping for the other fields. One possible explanation for the lack of more improvement from additional is that the new examples aren't correct, and thus add noise to the estimator. However, from inspection of the data the predominant trend is that the example accuracy doesn't degrade considerably. Thus, the lack of improvement must come from a different avenue.

Another possibility is that the feature space is sufficiently estimated by a small set of training examples and adding more examples doesn't yield additional information. Perhaps by switching to a richer feature space more improvement could be gained. These experiments and line of investigation will be left to future work.

## 4.7 Coreference : all gender-matching pronouns

The previous sections have only used exact last name matches in the text. This section considers what would happen to system performance given a simple coreference scheme while doing the training and testing extraction. Our coreference scheme uses a simple pronoun gender matching method, where every pronoun in a sentence which shared

Field	PCL		NB+E		CV+E		CM+E	
	-	Coref	-	Coref	-	Coref	-	Coref
Birthday	.85	.85	.85	.84	<b>.88</b>	<b>.88</b>	<b>.88</b>	<b>.88</b>
Birthyear	.39	.39	.77	.75	.86	.86	.88	<b>.89</b>
Occupation	.3	.34	.64	.57	.42	.65	.58	<b>.67</b>
Birthplace	.32	.41	.47	.46	.48	.50	.52	<b>.58</b>
Death Year	.39	.67	.19	.44	.75	.81	<b>.86</b>	<b>.86</b>
Avg. Gain		.08		.07		.06		.07

Table 4.16: Effects of Coreference on Biographic Fact Extraction. The “-” column holds the performance of the system without coreference. On average, a simple coreference resolution methods leads to a small gain in post-fusion accuracy for every method.

the same gender of the person in question is replaced by the person. This scheme will obviously increase the overall noise of the data, but since the noise will be uncorrelated, it will cancel out and the signal will appear more strongly.

To test out the coreference scheme, coreference substitution was performed over selected test sets of 1, 5, 15, 30, 50, 75, and 150 documents for the initial 5 fields.

As can be in seen in Table 4.16, there are gains across all extraction methods. Even for CM+E there is still some additional gain in performance by using this coreference scheme. The relations which benefit from the most gain are death year and occupation, which typically have the lowest performance among all of the relations.

Figure 4.23 show the effects of increased test set size on performance with respect to the three fields that shows the most improvement with coreference (birthplace, occupation, year-of-death). These graphs suggest that coreference improves performance at all values of the test set. Moreover, the weakest extractors are helped the most.

Tables 4.17 and 4.18 show the effect of performing coreference on the second field set (familial relationships + schooling). The observations from these experiments concur with results from the prior experiments. Adding coreference information yields a consistent

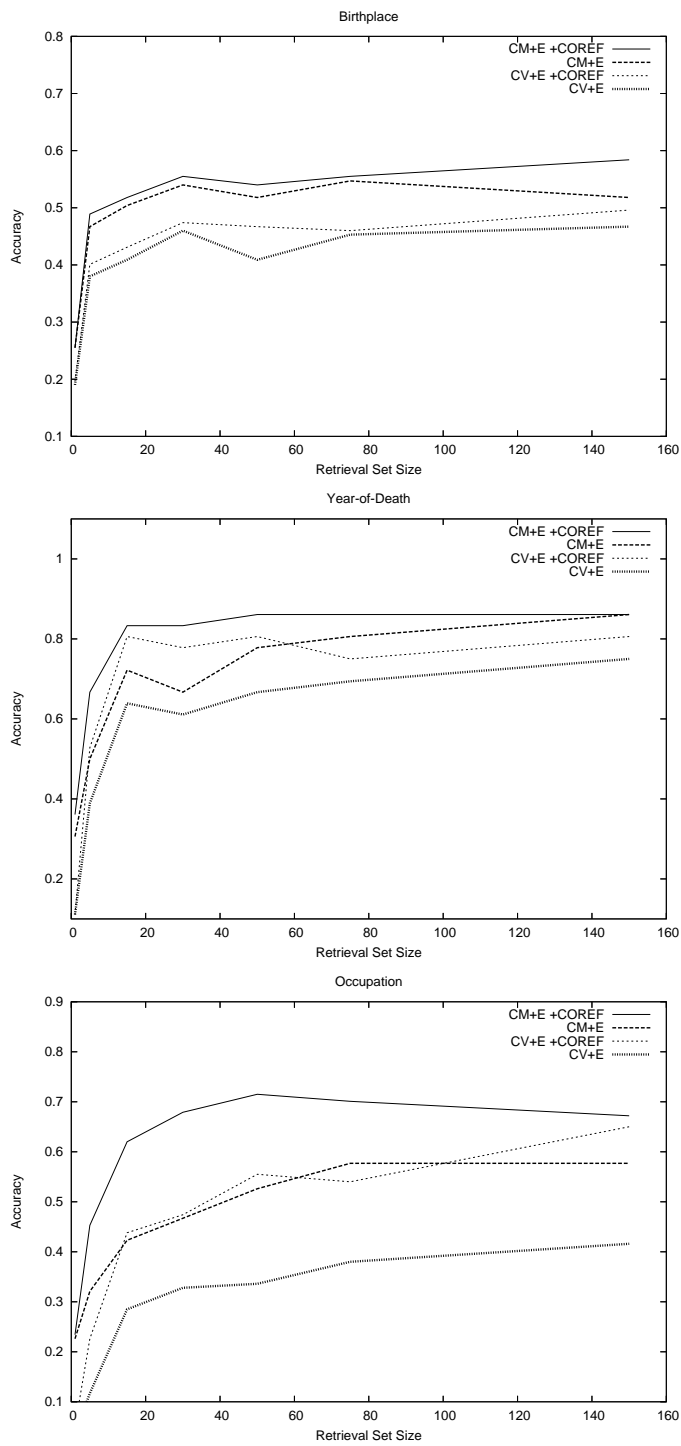


Figure 4.23: Effect of Coreference on occupation extraction with increased test set sizes. Performance improves at all test set size values.

Field	PCL		NB+E		CV+E		CM+E	
	-	Coref	-	Coref	-	Coref	-	Coref
Spouse	0	0.1	0.4	0.6	0.2	0.3	0.3	0.7
Son	0.143	0.143	0.143	0.143	0.143	0.143	0	0
Father	0	0	0	0.1	0	0.1	0	0.2
Daughter	0.2	0.2	0	0	0	0	0.4	0.4
Mother	0	0.1	0	0.2	0.1	0.2	0.2	0.4
School	0.714	0.714	0.286	0.571	0	0.286	0.714	0.571
R Avg. Gain		.04		.1		.06		.16
Avg. Gain		.03		.13		.1		.11

Table 4.17: System Accuracy for familial relationship extraction with and without coreference. The “-” column holds the performance of the system without coreference.

Field	PCL		NB+E		CV+E		CM+E	
	-	Coref	-	Coref	-	Coref	-	Coref
Spouse	0.071	0.234	0.4	0.6	0.283	0.4	0.356	0.7
Son	0.223	0.238	0.152	0.152	0.159	0.176	0.025	0.046
Father	0.011	0.005	0.1	0.303	0	0.3	0.086	0.347
Daughter	0.313	0.217	0	0	0	0	0.468	0.502
Mother	0.103	0.202	0	0.3	0.15	0.25	0.311	0.501
School	0.743	0.786	0.286	0.643	0.071	0.286	0.762	0.679
R Avg. Gain		.035		.14		.11		.17
Avg. Gain		.036		.18		.12		.13

Table 4.18: System performance by Mean Reciprocal Rank for familial relationship extraction with and without coreference. The “-” column holds the performance of the system without coreference.

improvement to the performance, which is born out both from results in precision and in improvements in extraction MRR (mean reciprocal rank). Another item to note is that performance improvement from coreference on relationships is in the same magnitude as performance improvement on the other fields.

## 4.8 Cascaded Fact Extraction and Fusion for Multi-Relationship Extraction

Biographic facts co-occur frequently and having a system which extracts them simultaneously might make sense. One option is to create a joint-field method in which multiple fields are simultaneously extracted. However, for a joint-field model, there are a number of additional modeling choices that need to be made. In the above scenarios, the best performing CRFs essentially work as classifiers and decide between whether the desired relationship or something else exists (when trained on negative examples). It doesn't seem clear how to specify a multi-field extraction problem in this way, and thus the gains realized by the models trained to predict spurious targets would be lost.

One solution proposed by [McDonald et al., 2005] is to create separate classifiers for each of the separate relations and then fuse the results. In that work, distinct relations (e.g. separate parts of a complicated CEO transition event) are extracted, and then a fused model chooses the combined assignment which provides the maximally probable assignment. This model isn't clearly applicable, since there are not strong constraints between the distinct relations in this situation.

Another alternative solution would be to have one CRF which marks both birthyear and birthday when they occur in the same sentence. There are two potential problems with this type of model. First, this model would need a large number of states to appropriately model the interstitial texts (e.g. for two relations in the same sentence it would need distinct states for between the hook and birthyear, between birthyear and birthday, between hook and birthday, birthday and birthyear), which would make the model dramatically

more sparse. Second, this model would need further modifications to take into account the cross-sentence and cross-document fusion which has proven extremely useful.

This section proposes an process where a cascade of fact extraction and fusion steps iteratively extracts a target for a given hook (e.g. birthday), marks up the text with the discovered information, and then extracts another relationship for the same hook (e.g. birthyear). In the model proposed in the next few pages, the order of the cascade is fixed. For example, to build a birth year extractor given knowledge of the birthday, given tuples  $birthday(x_i, d_i)$  and  $birthyear(x_i, y_i)$ , in training the hook corpus  $D_i$  is marked with the appropriate birthday  $d_i$  and the target birth year  $y_i$  and an additional feature is added to the CRF if the birthday occurs somewhere in the sentence.<sup>11</sup> This feature is not location-specific (i.e. the feature just indicates that the found birthday appears somewhere in the sentence, not either before-or-after the candidate birthyear).

In testing, for each hook the system finds the birthday using the methods presented in the previous sections, annotates sentences which contain that birthday with a *found-birthday-in-sentence feature*, and applies the fused birth year CRF. Figure 4.24 shows an example of the iterative markup that occurs when the system uses birthday information to learn birth year. In the case of using multiple relations, features which indicate the presence of each fact alone and features which indicate presence of two facts are added (e.g. if the found birthday and occupation both occur in the sentence, added features would be: birthday-in-sentence, occupation-in-sentence, birthday-and-occupation-in-sentence).

Table 4.19 shows the effect of using this cascade. Based on the relative performance

---

<sup>11</sup>The CRF state model doesn't change.

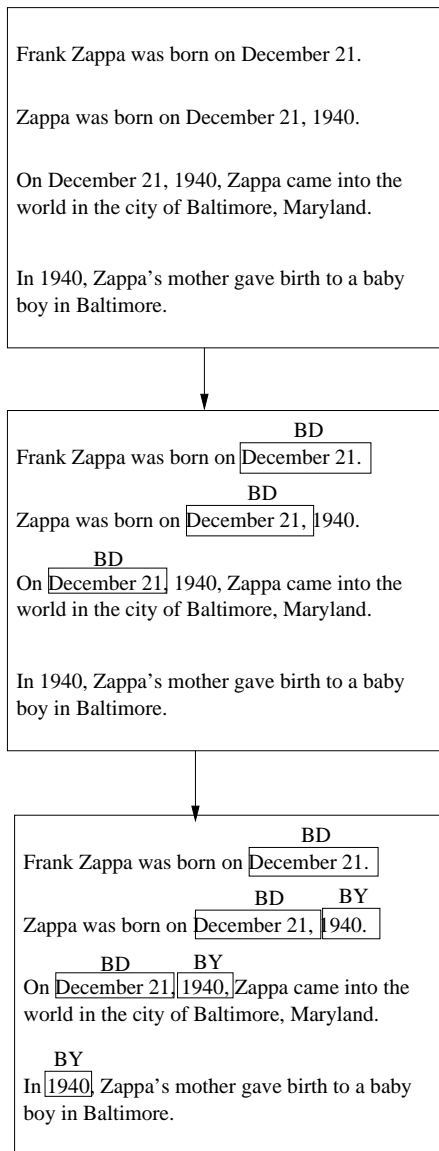


Figure 4.24: An Example of Cross-Field Bootstrapping : First the birthday (BD) is extracted and the text marked (middle diagram). From this annotated corpus, birth year (BY) is discovered and the text marked (right diagram). The discovered birth year may appear in contexts where the discovered birthday does not and improve extraction of further data such as birth place.

Birthday		
	Extraction Precision	Accuracy
CV	50.9	65.0
+ birthyear	69.9	75.9
CV+E	68.0	<b>88.3</b>
+ birthyear	77.3	87.0
Birth year		
	Extraction Precision	Fusion Accuracy
CV	34.2	79.7
+ birthday	47.2	86.1
CV+E	65.38	86.9
+ birthday	<b>80.9</b>	<b>89.1</b>
Occupation		
	Extraction Precision	Fusion Accuracy
CV	21.9	60.6
+ birthday	21.7	56.9
+ birthyear	21.7	59.8
+ birth year(f)	21.9	59.9
+ all	21.4	59.1
CV+E	24.6	42.3
+ birthday	32.5	57.7
+ birthyear	<b>39.3</b>	65.7
+ birth year(f)	<b>38.7</b>	<b>67.2</b>
+ all	38.2	64.2
Birthplace		
	Extraction Precision	Fusion Accuracy
CV	13.9	32.1
+ birthday	15.8	37.2
+ birthyear	15.1	34.3
+ birth year(f)	15.6	35.0
CV+E	35.7	46.7
+ birthday	35.0	47.4
+ birthyear	27.3	35.0
+ birth year(f)	29.4	35.0
+ occupation(f)	31.4	35.4
+ all	<b>36.2</b>	<b>53.2</b>

Table 4.19: Performance of Cascaded Fact Extraction and Fusion Models. (f) indicates that the best fused result was taken. birth year(f) means birth years were annotated using the system that discovered the most accurate birth years.

of each of the individual extraction systems, the following order for the cascade was selected: 1) Birthday, 2) Birth year, 3) Occupation, 4) Birthplace.<sup>12</sup> In the case of birthday, this schedule was revised to allow first the extraction of birthyear and then birthday. There are gains in accuracy for birth year, occupation and birthplace by using cascaded information extraction and fusion. The performance of the plain CV+E averaged across all five fields is 67.4, while for the best cascaded bootstrapped system it is 74.6, a gain of 7%.

Another area in which the impact of cascaded extraction can be seen is in the number of people who have all facts about them correctly extracted. When using cascaded extraction and fusion, the percentage of summaries that are completely correct improves greatly: to 37% from 13.8%, a gain of 24% over extracting the fields in isolation. Additionally, bootstrapping does not appear to hurt in cases where incorrect information is extracted. Performance in the bootstrapped system for birthyear, occupation and birth place when the birthday is wrong is almost the same as performance in the non-bootstrapped system.

## 4.9 Related Work

Finding information on public figures has received prior attention in the NLP research community. In particular [Schiffman et al., 2001] and [Cowie et al., 2000] create biographic summaries by composition of phrases or sentences from pages. This is not an information extraction task, but really an extractive summarization task, as it does not look for specific pieces of information, but rather summarizes what is available. [Chen and Bian, 1998] extract email addresses, URLs and proper nouns associated with

---

<sup>12</sup>This system has the extra knowledge of which fused method is the best for each relationship. This was assessed by inspection on the test data.

people mentioned on-line. Unlike the above approaches, information extraction is used to pick out key facts (e.g. birthday, birthplace) about individuals.

The closest work to the fusion methods presented here is [Skounakis and Craven, 2003] which presents a method for combining evidence from multiple instances to determine whether certain facts are true or not. They look at the case of protein interactions and attempt to classify whether or not two proteins interact using multiple possible extractions throughout the text. In biographic fact extraction there is a distinction in that the goal is to find the target for a particular hook, as opposed to classifying separately whether a relationship holds between any two possible hooks and targets. [Downey et al., 2005] follows in the same vein and presents a method for judging whether a particular fact is true given its presence in a large set of extracted facts. Another related study in fact extraction and fusion is [Masterson and Kushmerick, 2003], which performs multi-document fact extraction and resolves conflicts by choosing the single most confident extraction for a particular template slot. [Riloff and Jones, 1999a] demonstrates ways to build up facts of arity one (e.g. “city(Boston)” ) using multiple extraction patterns and bootstrapping.

Bootstrapping of extraction systems has proven very effective in other cases, but this chapter has shown mixed results which may stem from a few causes. First, the cases in which bootstrapping was least effective in this work was in cases where the underlying extractor was of very low precision (e.g. for the birthplace field). Prior work has only used high-precision patterns for bootstrapping. Second, bootstrapping in other cases has allowed the system to improve recall. In this system, there is a fixed corpus and a fixed set of test cases, and there simply might not be the available data in the test set to extract. Finally,

prior methods have primarily used PCL-variants to perform bootstrapping – methods which have been shown to be high precision and low-recall. By gaining more patterns, the recall is able to be improved. With the methods proposed here, the recall is already high – it is the precision that could be improved.

Somewhat more distantly related is [Stevenson, 2004] which presents evidence that processing single sentences alone is insufficient in order to retrieve information which is contained over multiple sentences.

A number of systems have used information fusion for multi-document summarization. Notable systems are SUMMONS [Radev and McKeown, 1998] and RIPTIDE [White et al., 2001], which perform closed domain summarization using an information extraction system whose output is assumed to be perfect. [Barzilay et al., 1999] presents another example of information fusion from multiple documents. Their system does not explicitly extract facts, but instead picks out relevant repeated elements and combines them to retain the semantics of the original.

In recent research into question answering, information fusion has been used to combine multiple candidate answers to form a consensus answer. [Clarke et al., 2001] use frequency of n-gram occurrence to pick answers for particular questions. Another example of answer fusion comes in [Brill et al., 2001] which combines the output of multiple question answering systems in order to rank answers. [Dalmas and Webber, 2004] use a WordNet cover heuristic to choose an appropriate location from a large candidate set of answers.

[Nahm and Mooney, 2002] presents a study on multi-field extraction. Their system learns word-to-word relationships by doing data mining on information extraction results.

[McDonald et al., 2005] proposes a method for combining the results of distinct extractions into one joint event. [Prager et al., 2004] uses temporal constraints on year of birth, year of death, and years when an person produced artifacts (e.g. wrote a book) to predict when people were born, died, and produced artifacts. In their study, the added constraints improved performance, though it is not clear on which facts the improved performance came from. Using the CRF extractors in our data set, this heuristic did not yield any improvement for detecting year of death from birth year. Other indirectly related work for multi-field extraction is [Sutton et al., 2004] and [Bunescu and Mooney, 2004] that suggest methods for combining information in graphical models across multiple extraction instances.

## 4.10 Conclusion

This chapter has demonstrated that the general training framework proposed in Section 2.2 can be usefully applied to extracting biographic facts (Section 4.3) and that fusion improves the accuracy of the extracted facts (Section 4.5). In particular, the weighted confidence method using confidence estimates from a CRF with negative examples gives the best performance. Test set size experiments showed that 30 of the top highly ranked documents give performance equal to performance on the whole 150 documents (Section 4.5.5). As with other forms of extraction learning, bootstrapping has allowed some improvement, where learned facts are added to the training data to allow for retraining (Section 4.6). A cascaded fact extraction model whereby an initial set of facts is learned and then that information is added back into the model in order to improve extraction of further facts also yields improvement (Section 4.8). Finally, coreference experiments demonstrate that a

slight gain can be realized by a simple coreference resolution algorithm.(Section 4.7).

In summary, this chapter demonstrates that biographic fact extraction can be reasonably performed by fact extractors trained by example. Chapter 6 explores these methods for a different domain and shows that they hold there as well. Together, these two results suggest that these techniques will be applicable for extraction of facts as defined in Section 1.1 whenever there is a corpus of redundant information. While the extracted facts presented in this chapter are useful in isolation, the next chapter suggests that they may also be useful as a part of a larger system.

## Chapter 5

# Fact Extraction for

# Cross-Document Coreference

This chapter presents the problem of cross-document coreference. [Bagga and Baldwin, 1998] popularly introduced this problem in the context of news corpora, and presented vector-space models to solve the disambiguation problem (See Section 5.3). They developed a corpus of articles from the New York Times which mentioned John Smith, and Section 5.7 presents the performance of standard vector-space models on these articles. This chapter, however, focuses on the distinct task of cross-document coreference in Web pages. A set of experiments in Section 5.5 shows that the facts extracted in the Chapter 4 can be used to improve cross-document coreference for a set of pseudonyms. Section 5.6 presents evidence that the types of facts used for pseudonyms do not improve performance with regards to a set of real polyreference, and suggests that pseudonyms do not necessarily reflect real polyreference. Across all of the experiments, Centroid Clustering matches or

exceeds the performance of Group Average Clustering and Single Link Clustering.

## 5.1 Name Polyreference

Reuters (March 13, 2003) observed the problem of name ambiguity to be a major stumbling block in personal name Web searches. Aside from the context of the World Wide Web, the general problem of proper noun disambiguation and cross-document coreference is an open problem and a difficult one. While word senses and translation ambiguities may typically have 2-20 alternative meanings that must be resolved through context, a personal name such as “Jim Clark” may potentially refer to hundreds or thousands of distinct individuals.<sup>1</sup> Each different **referent** typically has some distinct contextual characteristics. These characteristics can help distinguish, resolve and trace the referents when the surface names appear in online documents.

A search of Google shows 76,000 Web pages mentioning Jim Clark, of which the first 10 unique referents are:

1. Jim Clark - Race car driver from Scotland
2. Jim Clark - Clockmaker from Colorado
3. Jim Clark - Film Editor
4. Jim Clark - Netscape Founder
5. Jim Clark - Disaster Survivor
6. Jim Clark - Car Salesman in Kansas
7. Jim Clark - Fishing Instructor in Canada
8. Jim Clark - Computer Science student in Hong Kong
9. Jim Clark - Professor at McGill
10. Jim Clark - Gun Dealer in Louisiana

This chapter presents a set of methods for distinguishing the real world referent of a given name in context. In certain cases, facts extracted by means described in the

---

<sup>1</sup>A search in the phone book at <http://peoplesearch.lycos.com/> returned 622 entries for “Jim Clark” and 3708 entries for “James Clark”

preceding chapter may be used to improve performance on the cross-document coreference task. Other types of features such as Web link structure, text in pages linked to or by a particular page, and wide-context lexical features are not pursued.

This work focuses on the problem in which one name links to multiple referents. The converse, where multiple names correspond to one referent has been explored by [Ravin and Kazi, 1999] and [Wacholder et al., 1997].

## 5.2 User Scenarios

Unlike Chapter 3, which was concerned with a specific task where a user had a question and wanted to find an exact answer for that question, the methods in this section disambiguate referents which can then serve in variety of user tasks.

One of the most immediately applicable tasks is simply given a query on a Web search engine, return a clustering of Web pages where each cluster is mono-referent. For disambiguation, each cluster could be labeled with relevant information (e.g. occupation and birthplace) for the referent of interest in order to facilitate finding the cluster desired. This is the main scenario examined in this dissertation.

However, often the user knows ahead of time some characteristics about the person they're looking up, for example their age. Instead of performing a clustering without any information, the system could use this information to aid in clustering, or alternatively to rank pages which are most relevant to this particular person. The experiments in Section 5.5.3 explore this scenario.

The previous section referred to situations which start with a user query. Alter-

natively, the system could work as a filter on top of Web pages, where each Web page is annotated in some fashion with links to additionally relevant pages. In this case, you might skip from a page which mentions a particular referent to other pages about that same referent. In this case, the entire contents of the referring Web page can serve as added information to find the referent of interest. Section 5.5.4 gives a demonstration of this idea.

### 5.3 Cross-Document Coreference Method

This section lays out the cross-document coreference method that will be used as a basis for the experiments in the rest of the chapter. In this work, one of the first assumptions which is made is a that there is a single referent per-name per-document. That is, it is assumed that there won't be more than one distinct "John Smith" in each document. With this assumption, in order to asses whether or not two names in separate documents refer to the same person, the systems test how similar the two documents are.

[Bagga and Baldwin, 1998] presented an online linking algorithm, whereby referents in new documents are linked to the most similar previous document to create chains of coreference. The similarity measure used in those experiments is a form of local vector-space model. Each document is represented as a vector of words, where each word is assigned a weight according to some weighting scheme. The similarity of two documents is the cosine distance between the two vectors:

$$\cos(a, b) = \frac{a \cdot b}{\|a\| \times \|b\|}$$

That work uses TF-IDF as a weighting scheme, where in a collection of documents of size

$N$ ,  $a_i$  (weight of word  $i$  in document  $a$ ) is:

$$a_i = \text{freq}(a, i) \log \frac{N}{n_i}$$

where  $n_i$  is the number of documents the word appeared in, and  $\text{freq}(a, i)$  is the number of times the word  $i$  appears in the document  $a$ .

To create vectors for each document, that work suggests using only local context around the name mentions to fill vectors for a particular document. The motivation for selecting only local context is that the local context for a given word is going to be more indicative of the referent than the entire document. This chapter considers 50 and 100 and entire document contexts for building document vectors.

Following [Mann and Yarowsky, 2003], this section proposes a unsupervised document clustering model, using both TF-IDF and a new weight measure NNP+MI. The second weighting scheme NNP+MI is chosen to be better suited to the task of cross-document coreference. The weighting scheme selects only the proper nouns in the corpus and the most relevant words to a document collection. Here relevance is calculated as the mutual information between the word  $w$  and the corpus  $c$ :

$$I(w; c) = \frac{p(w|c)}{p(w)}$$

An outside collection of Wall Street Journal newswire (from 1987) was used to calculate  $p(w)$ . The system only retains words which appear more than 20 times in the collection and have a  $I(w; c)$  greater than 10. These words are added to the document's feature

vector with a weight equal to  $\log(I(w;c))$ . This weighting scheme was chosen because association between people (and between places) was thought to be especially important for cross- document coreference, and the mutual information of words with a particular referent corpus would be likely to pick out words that were most correlated with referents, as opposed to particular documents.

This chapter explores the use of three clustering methods. Single Link Clustering, Group Average Clustering (proposed in [Gooi and Allan, 2004]) and Centroid Clustering (following [Mann and Yarowsky, 2003]). For all of these clustering algorithms, at each step, the system merges the two most similar clusters.

For Single Link Clustering, the similarity between two clusters is the largest similarity between pairs of documents for each cluster. For clusters  $A$  and  $B$  with documents  $A_0..A_j$  and  $B_0..B_k$ , the single link distance is :

$$S_{sl}(A, B) = \max_{a \in 0..j, b \in 0..k} sim(A_a, B_b)$$

In Group Average clustering, the clusters with the highest average similarity between documents are put together.

$$S_{ga}(A, B) = \sum_{a=0}^j \sum_{b=0}^k sim(A_a, B_b)$$

In Centroid Clustering, after a merge, the term vector for the resultant cluster is recalculated as the average of terms from all of the constituents clusters and vectors ( $A^*$ ). Then similarity

is computed as similarity between those vectors.

$$S_{cc}(A^*, B^*) = sim(A^*, B^*)$$

The above methods describe an ordering for clustering, but do not describe how to pick the number of clusters. This section reports performance given the optimal possible threshold on the development and test sets and from a threshold optimized on the development set for the test set.

### 5.3.1 Applications of Facts for Cross-Document Coreference

Information extraction techniques have the potential to add high precision, categorical information to less precise bag-of-words methods. This categorical data can support or exclude candidate name↔referent matches with higher confidence and greater pinpoint accuracy than via simple context vector-style features alone.

This chapter presents a simple method for using extracted biographic facts to improve clustering of coreferent entities. In this method, biographic fact extractors are applied over all of the documents to extract facts about a particular individual. Then, all documents which have matching features are merged to create seed clusters. For each of these seed clusters, a centroid for the cluster is calculated (by averaging the document term vectors of all constituent clusters), and then pages from which no biographic facts were extracted are put into one of these seed clusters. Because of their high degree of precision and specificity, documents which contain similar extracted features are virtually guaranteed to have the same referent. In addition to improving disambiguation performance, these

extracted features help distinguish the different clusters and provide information about the different people.

In order to extract the baseline biographic facts, the CV+E and CM+E extractors built in Chapter 4 were applied over the various corpora. For the CM+E extractors, any fact extracted with confidence higher than 10% was included.

The proposed method does not explicitly model mismatched extractions, where two documents have different values for a specific relationship (i.e. two mismatched birthdays). Appropriately utilizing this information is not straightforward, as the source of conflicting information can be different referents or bad extraction. For the extractors built in Chapter 4 and the data examined in this chapter, it is the case that after the clusters have been separated by matched features, the majority of remaining fact mismatches come from bad extraction, and thus a mismatch model will likely have little or no effect.

An alternate method for using extracted facts, which is considered in [Mann and Yarowsky, 2003] but not reported on here, gives higher weight to words which have ever been seen as filling a pattern in the term vector. For example, if 1756 is extracted as a birth year from a syntactic-based pattern for the polysemous name, then whenever 1756 is observed anywhere in context (outside an extraction pattern), it is given a higher weighting and added to the document vector as a term.

This chapter uses a subset of the information extracted above. In particular, only the most reliable and broad-coverage information is used, with the expectation that it is this reliable and frequent information that is most likely to aid in disambiguation. Only birthday, birthyear, and birthplace are considered. Occupation is excluded because of the

frequency of illegitimate matches, and year of death is too infrequent to be of use.

Email addresses, another potentially useful feature, is excluded for a number of reasons. A preliminary study suggested that email address mentions are not broad-coverage enough to make a significant difference. Celebrities don't have publicly available email addresses, so collecting email addresses for these cases will only lead to errors. Training an email address extractor via the methods in Section 2.2 is difficult because of the challenge of choosing appropriate seeds. Email addresses are rarely presented co-sententially with names (as in "Mrs. Luna email:luna@stellakins.com"), instead being more typically recoverable from wider contexts, something outside of the ability of the extraction system used in this dissertation.

In general, the space of potential extracted facts is quite large, and there may be other facts useful for cross-document coreference that are unexplored in this dissertation. The system developed in this chapter is a proof-of-concept demonstration that extracting a minimal set of facts is useful and can lead to higher performance, though the cost for extracting a large set of facts is prohibitively high. Scaling up information extraction methods in order to be able to automatically extract multiple different kinds of information from wide contexts is a goal of future work.

## 5.4 B-Cubed for Cluster Evaluation

[Bagga, 1998] proposed the B-Cubed method for evaluating coreference chains as an improvement over the MUC-6 evaluation measure, and that evaluation metric has since been widely adopted by the cross-document coreference community. Their method uses the

mean per-document precision and per-document recall, where per-document precision and recall are calculated for each document  $i$  as:

$$\text{Precision}_i = \frac{\# \text{ documents correctly clustered with document } i}{\# \text{ documents in cluster containing } i}$$

$$\text{Recall}_i = \frac{\# \text{ documents correctly clustered with document } i}{\# \text{ documents that should be clustered with } i}$$

This method was originally stated in terms of coreference chains, but as there is no ordering information in the chains, the chains were evaluated in the same way a cluster without internal chaining would be. Therefore, it can be suitably applied to the clustering methods discussed herein.

## 5.5 Web Pseudonym Experiments

This section examines performance on disambiguating referent mentions on Web pages. In order to investigate this phenomenon, this section examines performance on disambiguation of pseudonyms and artificially creates a test set from automatically retrieved Web pages.

Pseudonyms are constructed from two distinct names (e.g. Woody Harrelson and Miles Davis), to create a new pseudonym (WoodyMiles HarrelsonDavis). To create a document set corresponding to this pseudonym, pages for each of the two individuals are downloaded, and all instances of each name are replaced by an otherwise unique token (e.g. XXXs) or removed. The goal is then to determine which pages came from which original queries. There is no manual effort required, as the correct labeling is known. If one of

the names in the pseudonym is polyreferent, then the resulting pseudonym would only be able to distinguish between incorrect merges, as splits might in fact be correct, as the system might have correctly split two different referents. However, pseudonyms created from celebrities are unlikely to create this problem as the names themselves are unusual and the celebrity referent overwhelms other referents with the same name.

The pseudonyms in this section are taken from celebrities in the IP152 set (Appendix A). Two sets of names were selected, a development set of names (16-26 in IP152) and a test set of names (27-36 in IP152). 150 documents were collected for each name in both the development and test sets, and within each of these retrieved document sets, the pages with no mentions of the name in question were removed, and top 50-ranked documents from the resulting set were selected. All mentions of the name in question were then removed from each of these 50-document sets.

In both the development set of names and the test set of names, every possible pair of names was used to create the pseudonyms. In some cases, this created pseudonyms of different genders, similar to occasional cases of real polyreference for gender-ambiguous names such as Chris and Robin. Though gender information was not explicitly used for disambiguation, it is possible that it would be a useful disambiguation cue.

The use of pseudowords for word-sense disambiguation was introduced in [Gale et al., 1992] and [Schutze, 1992] as an inexpensive way for generating training and testing data. [Gaustad, 2001] suggested that performance on pseudowords is markedly better than on real ambiguous words.

For this task, pseudonyms seem to be an appealing way to generate training and

test data for a disambiguation phenomenon that has not received the attention that has accompanied word-sense disambiguation. Along with this study, [Niu et al., 2004] also use celebrity pseudonyms to test disambiguation techniques. This section explores the use of pseudonyms and presents some initial findings which suggest that extracted features can make a difference in cross-document coreference.

The next section repeats these experiments in the case of real instances of polyreference on the World Wide Web in newspapers and shows that genre has a dramatic effect on task difficulty and on the method applicability. While the results in this section give some indication of the utility of facts for cross-document coreference, the improved performance results in this section is tempered by the decreased performance results reported in subsequent sections.

### 5.5.1 Unsupervised Clustering

This section compares the performance of Single Link and Group Average Agglomerative Clustering as suggested in [Gooi and Allan, 2004] with Centroid Clustering proposed in [Mann and Yarowsky, 2003]. Following common practice, instead of clustering all of the words from the documents, each document vector was composed of a 50 word window surrounding the name mention. Along with TF-IDF, NNP+MI weights were investigated to compute similarity between clusters. All of the methods were evaluated by the B-Cubed metric.

Given a 100-document set composed of 50 pages from each of two celebrities, the system automatically disambiguates the pages by referent. Baseline performance on this task is 66% which is achieved by placing all of the pages in one cluster, as the B-Cubed

F-measure is then the harmonic mean of 50% and 100%.

The results report the best B-cubed F-measure for each method, with both the optimal stopping threshold and stopping thresholds estimated from training data. Many of the methods which didn't use extracted facts were unable to beat this baseline. The fact that few systems were able to beat the baseline in this test set is demonstrative of the difficulty of this particular test set.

The main reason for the difficulty of this test set (as opposed to others which will be examined later) stems from the properties of Web pages and Web page mentions as compared to newswire. First, these Web pages are relatively short: newswire pages have on average 2000 words per document, while these Web pages have on average 850 words per document (test and development set). Perhaps more significantly, celebrities often appear in very similar contexts online (e.g. Amazon Web hits for items involving the person), while in newswire the contexts are quite different for different people.

Figure 5.1 shows performance of the systems on the development set, with the optimal stopping threshold. Figure 5.2 shows the performance of the systems on the test set, with the optimal stopping thresholds. Figure 5.3 shows the performance of the system on the test set, with the stopping thresholds optimized on the development set.

For this test set, Group Average Clustering and Centroid Clustering yielded significantly better average performance than Single Link Clustering (69.1% and 68.7% vs. 66.6%)<sup>2</sup> NNP+MI works significantly better than TF-IDF (69.6% vs. 66.6%), suggesting that filtering out words is effective for noisy Web pages. Furthermore, the use of 50- and

---

<sup>2</sup>For this test and the following statistical tests, a paired t-test is used (p=.95)

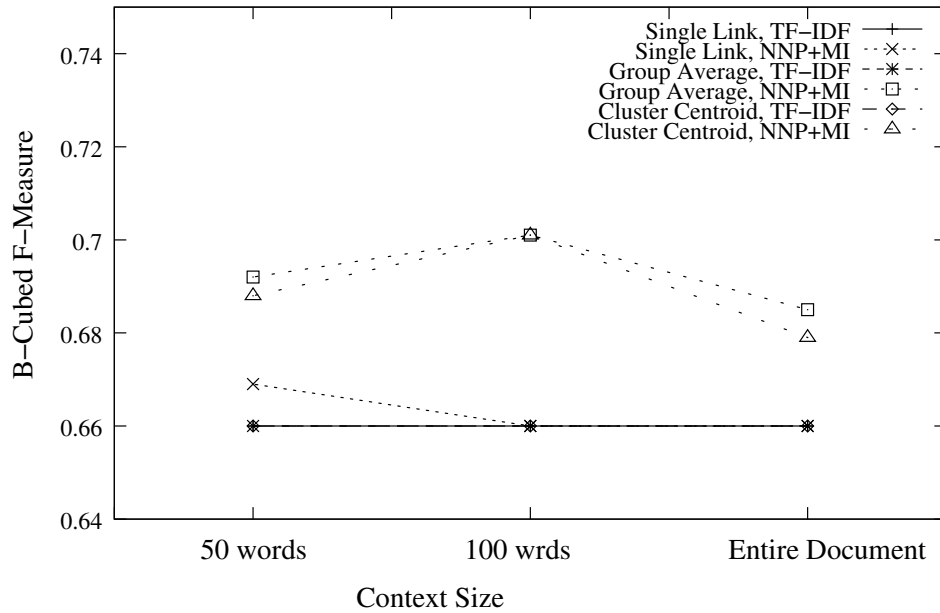


Figure 5.1: Effect on Context Size for Clustering Pseudonym Documents (Dev Set). The 100-word context using either Centroid Clustering or Group Average with the NNP+MI weighting scheme achieves the best performance. Baseline performance is 66%.

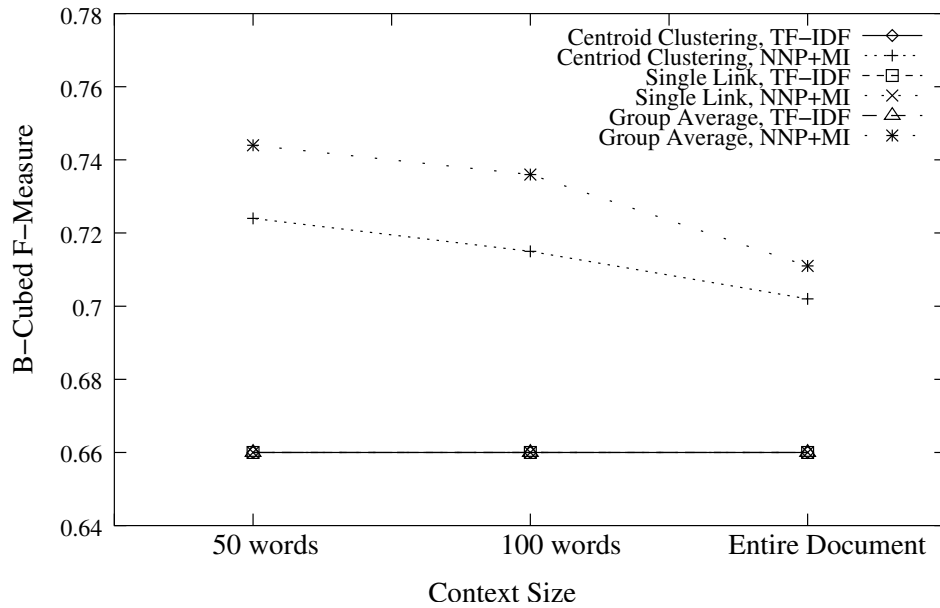


Figure 5.2: Effect on Context Size for Clustering Pseudonym Documents (Test Set), with the oracle stopping threshold. The 50-word context using Group Average with the NNP+MI weighting scheme achieves the best performance. Most of the other techniques don't perform better than baseline.

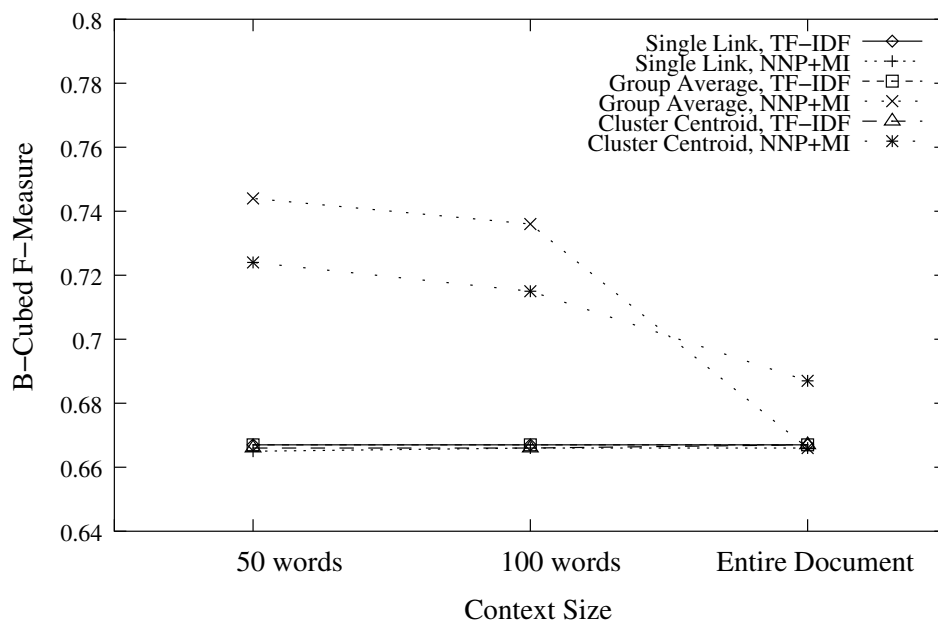


Figure 5.3: Effect on Context Size for Clustering Pseudoname Documents (Test Set) with the stopping threshold determined by the training data. The 50-word context using Group Average with the NNP+MI weighting scheme achieves the best performance. Most of the other techniques don't perform better than baseline.

100-word context windows is significantly better than the use of the entire document (68.8% and 68.6% vs. 66.9%). On the test data set, the optimal stopping threshold gave no better performance than the stopping thresholds optimized on the development set.

Figures 5.4 and 5.5 show the effect of varying the stopping threshold for each of the clustering methods. Figure 5.4 illustrates performance with the NNP+MI weighting method, while Figure 5.5 illustrates performance with TF-IDF weighting.

### 5.5.2 Seed Clusters from Fact Extraction

In the next set of experiments, the system described in Chapter 4 is used to extract biographic facts (birthday, birthyear and birthplace) for each document in the pseudoname collection (Section 5.3.1). By merging documents which share any feature, seed clusters are

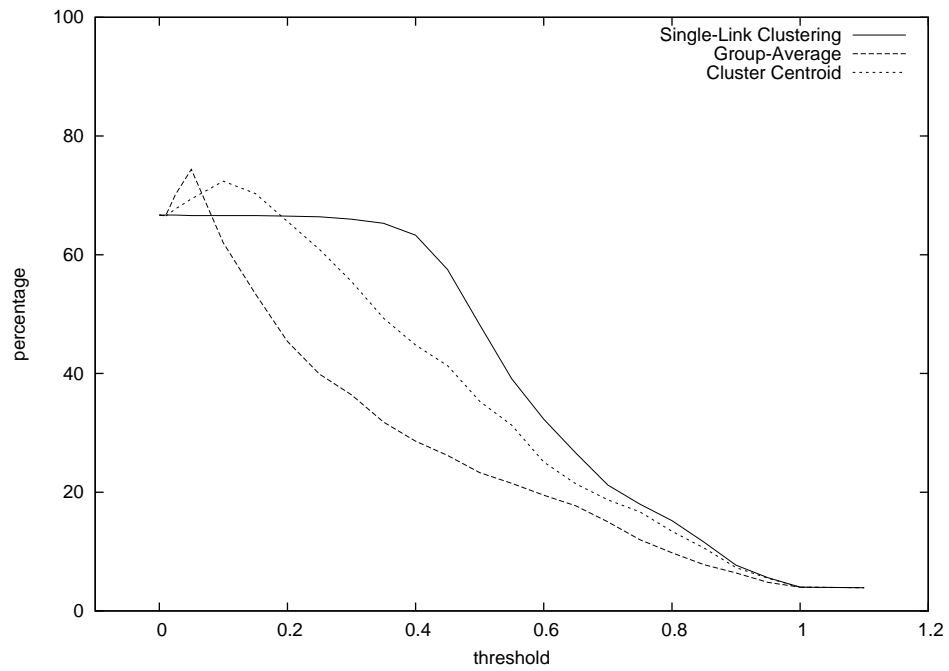


Figure 5.4: Stopping Thresholds for Clustering Methods over the Pseudonym Test Set using NNP+MI weighting.

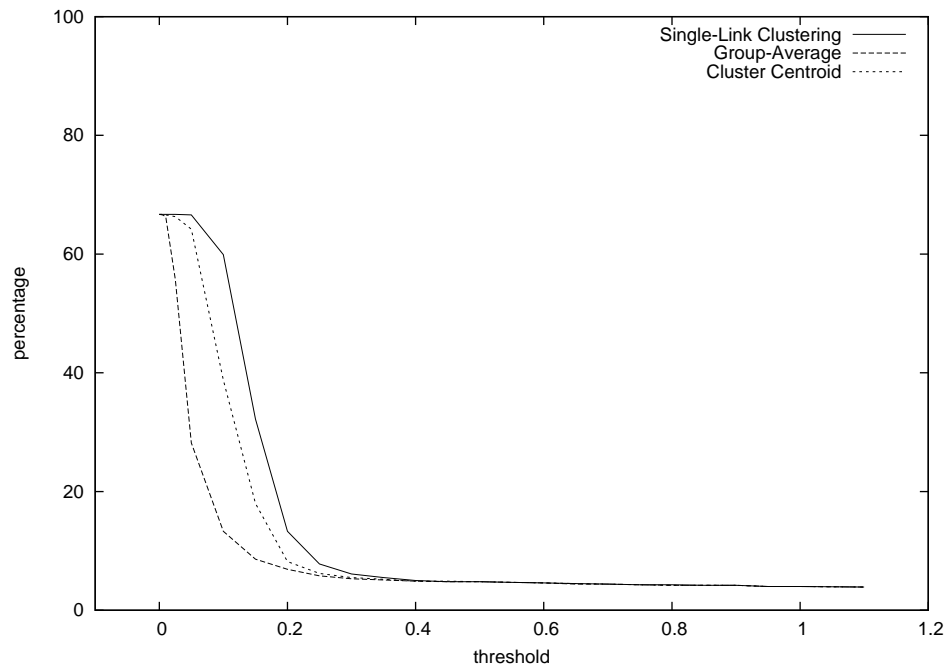


Figure 5.5: Stopping Thresholds for Clustering Methods over the Pseudonym Test Set using TFIDF weighting.

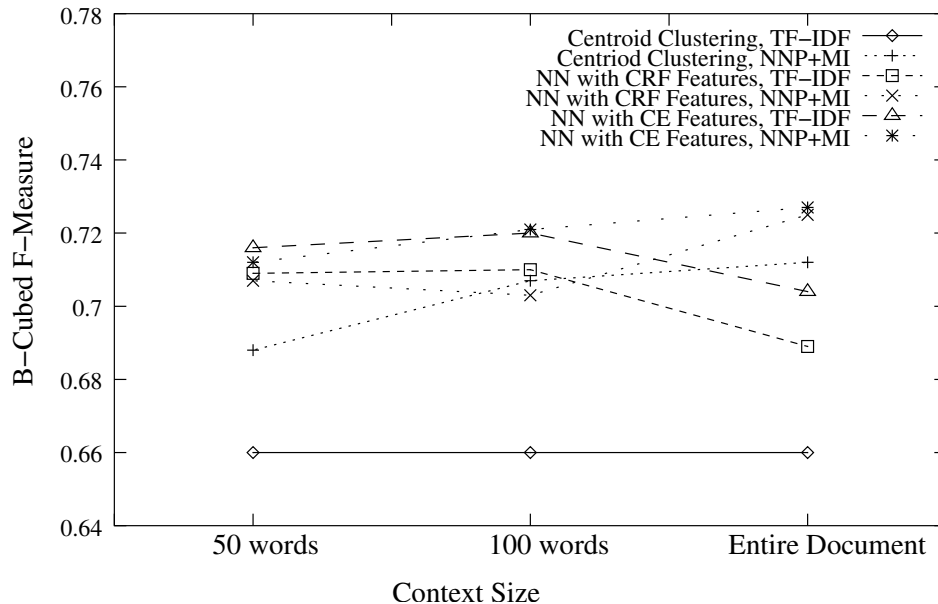


Figure 5.6: Effect of extracted facts on Unsupervised Pseudoname Clustering (Dev Set). Methods which use automatically extracted facts consistently beat the baseline, with the CM+E extraction method achieving the highest performance.

formed. Each unmerged page is then assigned to one of these seed clusters.

For the IP152 development set, each referent had an average of 23 pages containing extracted facts (or around 46% of pages had an extracted fact), and for the IP152 test set, each referent had an average of 25.1 pages containing extracted facts (or around 50% of pages had an extracted fact).

Figures 5.6 and 5.7 illustrate the improvement in performance from using these features. All of the runs with extracted facts beat the baseline, and the best system with extracted facts gave performance around 2% higher than any system without features. Moreover, the performance of the systems with nearest neighbor assignment to seed clusters do not require a threshold for stopping.

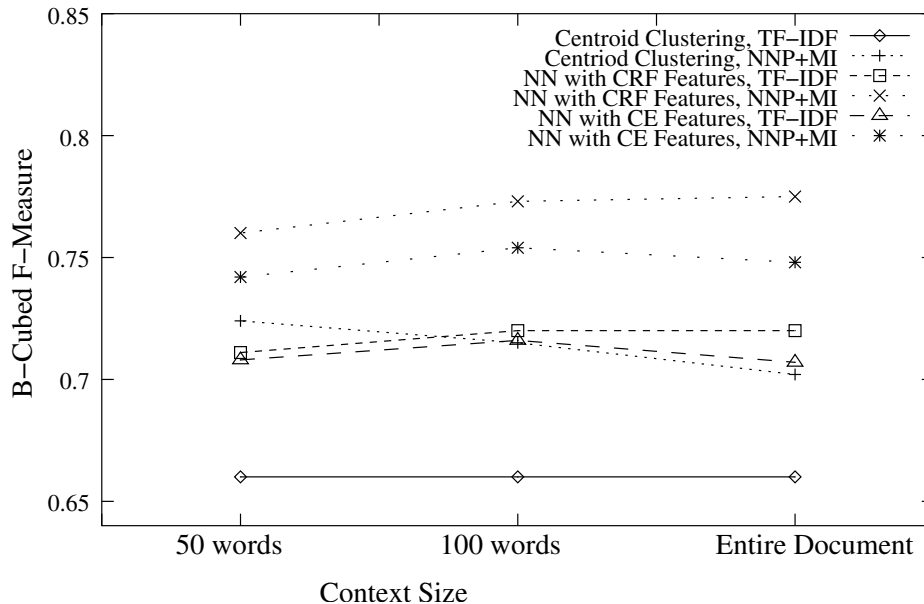


Figure 5.7: Effect of extracted facts on Unsupervised Pseudoname Clustering (Test Set). In the held-out test set, the CV+E extracted features achieve the highest performance.

### 5.5.3 Seed Clusters from Oracle Facts

The next set of experiments starts from complete knowledge of the correct biographic facts for each referent in the pseudoname, and these known form seed clusters. This corresponds to a task in which the user knows biographic information about an individual to start. All pages with matching extracted facts are assigned to the seed cluster they matched most (or to none if none of the features matched). Cluster centroids are then computed for these initial seeds, and all unassigned pages are clustered to these seed clusters.

Figures 5.8 and 5.9 show the results of these experiments for the two pseudoname sets. As can be seen from the table, having the true biographic facts aids clustering. There are many possible explanations for this. First, erroneous features do not form new seed clusters. Furthermore, knowing the complete set of features for each individual helps unify pages from which only one or two correct facts might be extracted. Finally, in cases where

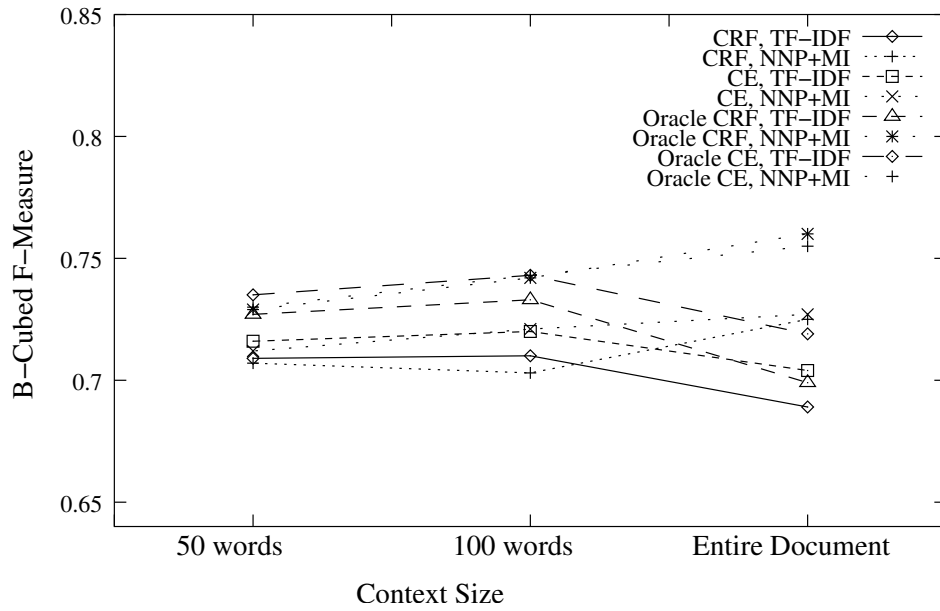


Figure 5.8: Oracle Experiments where for each referent, the true biographic facts for each referent is used to build a seed cluster, and then pages were clustered to those seeds (Dev Set). Overall, the CV+E extractor yields the best performance on this task.

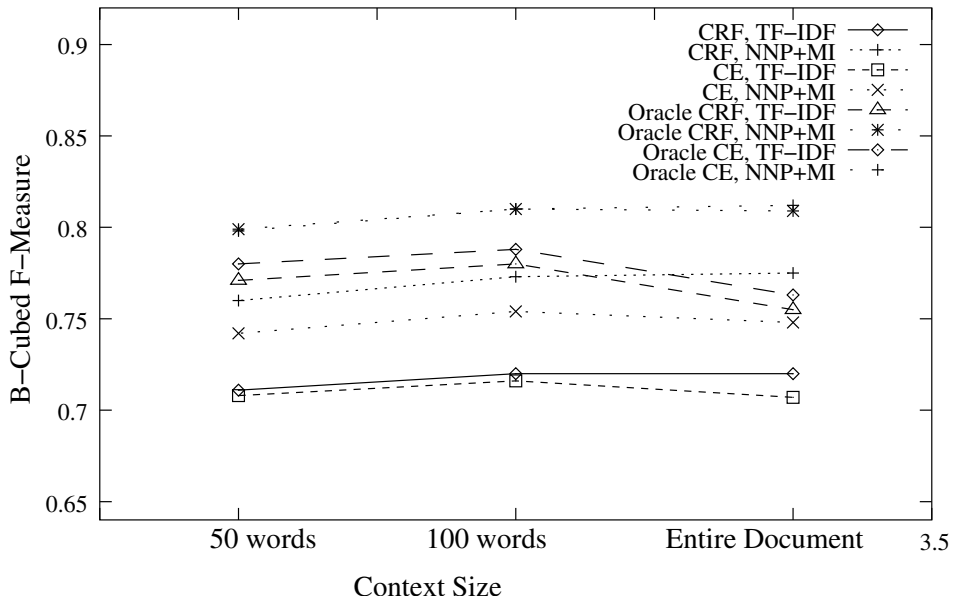


Figure 5.9: Oracle Experiments where for each referent, the correct features were used to build a seed cluster, and then pages were clustered to those seeds (Test Set). The CV+E and CM+E extractors with oracle feature sets yield the best performance on this task.

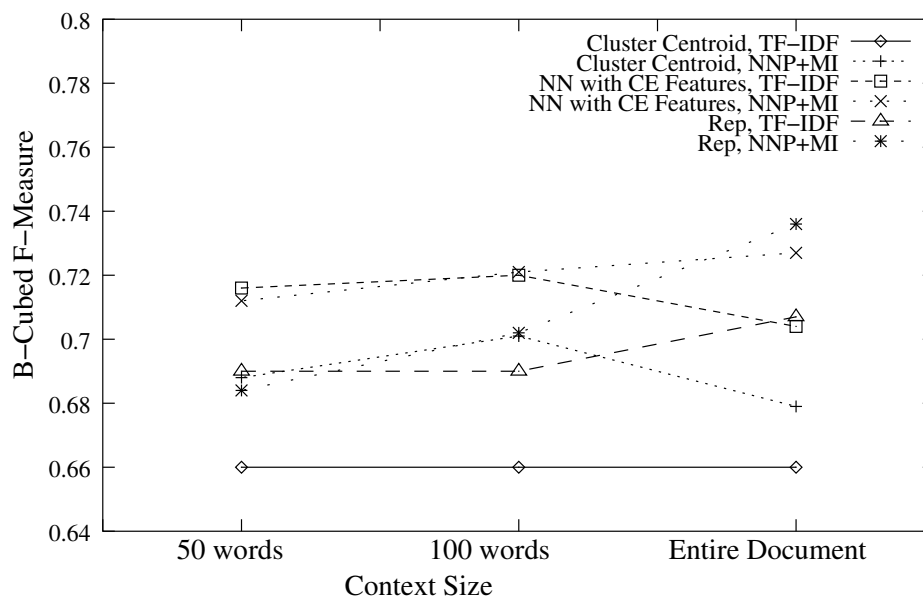


Figure 5.10: Representative Pages Experiment (Dev Set). Representative pages are used to form seeds for clusters to which all remaining documents are then clustered.

two people have misleadingly identical features (like sharing a birthyear), having the truth for each person allows documents with multiple pieces of information to be placed in the correct cluster, and additionally there are two well-formed clusters as opposed to one giant cluster which represents both people.

### 5.5.4 Representative Page Initialization

In the final set of experiments for the pseudoname pairings, we selected a representative page for each person in the pair, and used these pages to disambiguate the remaining pages. To find the representative page, a set of reference pages were selected by hand from either the Wikipedia ([www.wikipedia.com](http://www.wikipedia.com)) entry for a particular individual or another page which was significantly biographic in nature. This task is related to the task of finding other pages which mention the same referents as is on the current page.

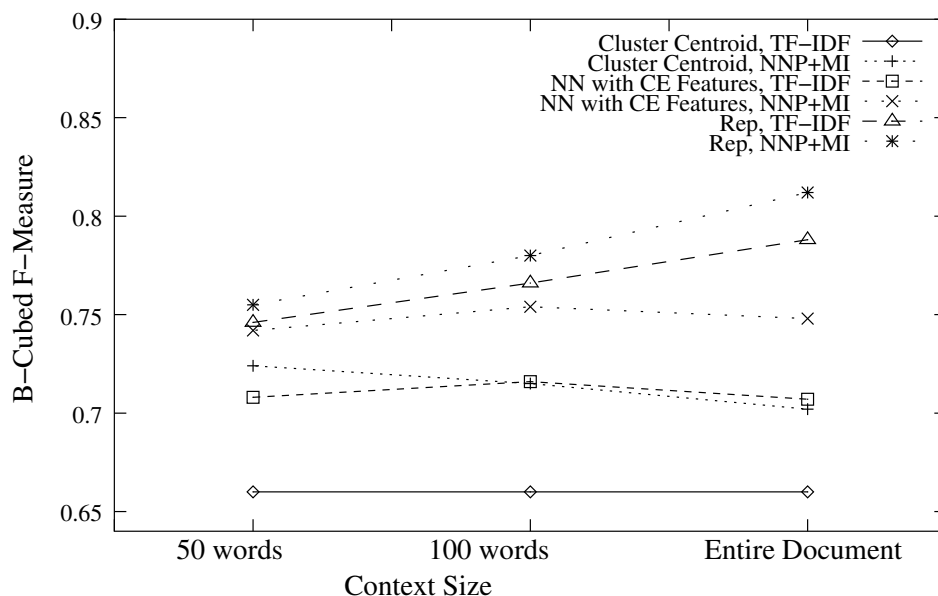


Figure 5.11: Representative Pages Experiment (Test Set). Representative pages are used to form seeds for clusters to which all remaining documents are then clustered.

The results in Figures 5.10 and 5.11 suggest that using representative pages can improve performance, while the performance of the unsupervised system with extracted features comes close to the performance of the system with those representative pages.

## 5.6 Web Polyreference Experiments

The above discussion has been focused on an artificially constructed situation, where a pseudonym was created by combining the data for two unique individuals, where there were equal numbers of pages about each individual in each disambiguation set. These pseudonyms and disambiguation sets do not necessarily represent true cases of polyreference for two main reasons. First, the people chosen are celebrities. Second, the distribution of number of referents for a given name and the number of pages per referent do not necessarily match those found on the Web in real cases of polyreference.

The first difference concerns the disparity in number of mentions between celebrities and non-celebrities, and has in some places been argued to be insignificant [Niu et al., 2004]. This appears not to be the case on the Web. In particular, celebrities frequently appear in biographical contexts on the Web, where articles discuss their birth, death and other factors of a personal nature. Non-celebrities are rarely mentioned in these kinds of contexts on the Web, while they are more frequently mentioned as members of organizations (companies, civic associations, entrants to a marathon).

The second set of distinctions, in number of referents per name, and in number of pages per referent, has also been little explored in the literature. The number of referents for a given name is closely related to the population size sampled – a larger population will inevitably increase the number of referents per name. As the number of pages on the Web increases, there will be a corresponding increase in the number of referents per name, with an increase in the problem of polyreference. This is in contrast with the relatively static world of word senses.

One way to approach the problem of predicting the number of referents for a particular name is to use population statistics to estimate the popularity of names and determine the probability of a given name. This probability can then be used to determine whether the name is relatively common or infrequent, or given an estimate of the population size being sampled, can be used to directly estimate the number of unique referents for a name. Figures 5.12 and 5.13 are graphs of the probability of first and last names taken from the US Census data [Census, 1990]. Here, the probability is the number of expected referents for a given name.

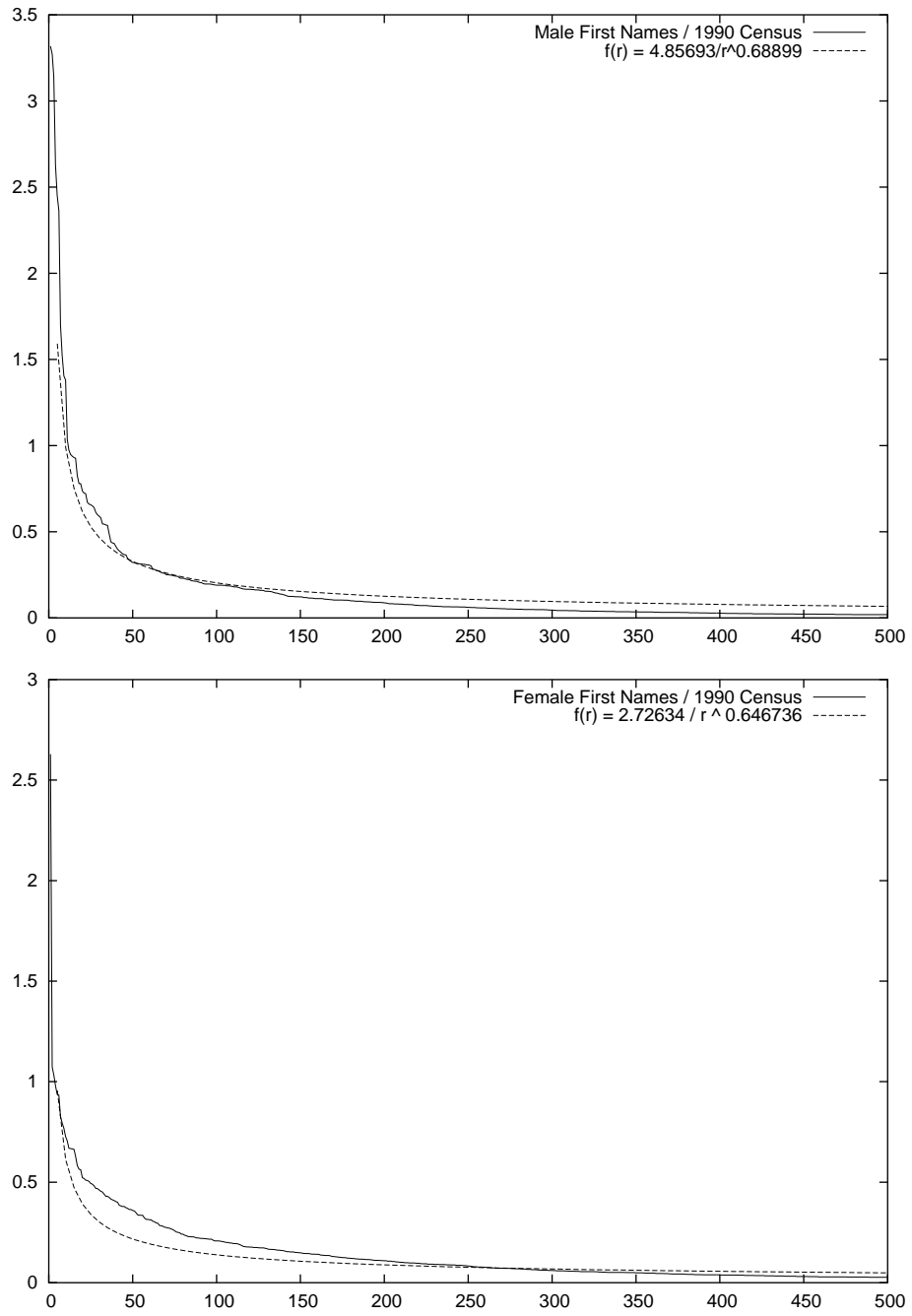


Figure 5.12: Male and Female First Names Distribution, the relative number of referents per name, fit to a power law. Only the top 500 names are shown.

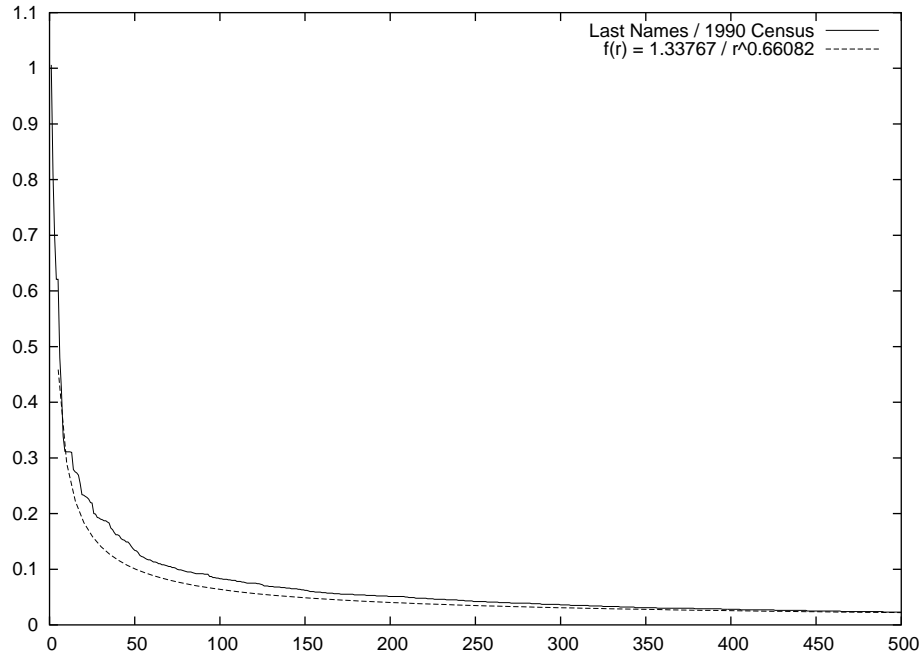


Figure 5.13: Last Name Distribution, the relative number of referents per name, fit to a power law. Only the top 500 names are shown.

Zipf's law:

$$f = k/r$$

(where  $f$  = the frequency of the word, and  $r$  = rank in word frequency, and  $k$  is a constant) has been applied to probabilities of words in corpora. It predicts that a word's frequency will decrease inversely with the rank of the word. Applied to name distributions, this formula would predict that the number of referents for a given name will decrease inversely with the rank of popularity of the name. The data in Figures 5.12 and 5.13 are fit to a related power law:

$$f = k/r^a$$

where both  $k$  and  $a$  are fitted. As can be seen, the fit and the data match closely.

### 5.6.1 Web Polyreference Data Set

A natural question then becomes, does Zipf’s law hold for people on the Web? Prior to this work, there was no Web polyreference data available for a relatively unbiased sample of names. In order to test the above methods, a set of Web polyreference pages was created (Appendix B, W03). The method of construction follows.

First names and last names were sampled independently from the U.S. Census distribution. All names that either did not appear on the Web or were ambiguous with common nouns or places (e.g. “X Boston” and “X Art”) were thrown out. 32 names were selected from that list, and for each name a query for that exact name (e.g. “Abby Watkins”) was issued to Google. The top 100 pages (at maximum) were downloaded for each name. In total there were 212 referents, and 882 pages.

From this collection, the pages retrieved for each name were manually disambiguated as to which referent they discussed, and any pages which did not contain a mention of the name in question were discarded. Pages without the name mention would occasionally occur as a result of changes in a page since Google had indexed the page, when the name appeared in a JavaScript block but didn’t appear on the downloaded page, or if the name appeared in anchor text but not in the page source.

In order to perform the manual annotation, a conservative coreference strategy was used. Two pages were judged to be coreferent if there was information on the two pages that could be used to connect the pages (e.g. employment at the same place, living in the same city). If there was no information which convincingly linked them, they remained

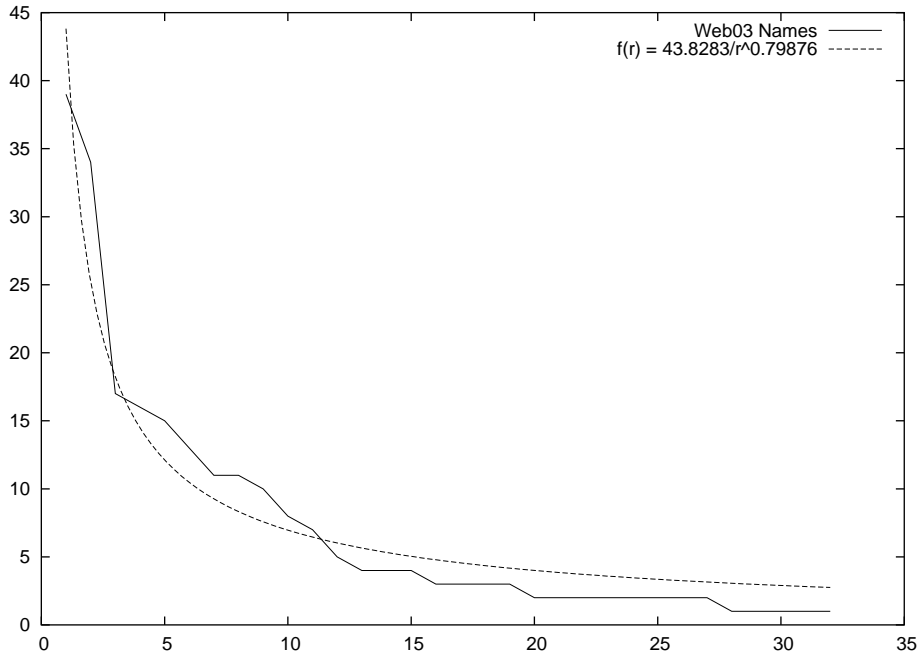


Figure 5.14: Number of Referents per Name for W03 with a power law fit.

separate. This process itself is prone to errors, as the true relation between the pages is not always knowable. This process of disambiguation took approximately 3-4 hours per 100 pages.

Figure 5.14 plots the popularity of each name (i.e. the number of referents per name). Figure 5.15 plots the number of pages per referent. For both of the distributions, a power law appears to hold.

Figure 5.16 shows the number of referents which are mentioned on a given number of pages. The most striking trend revealed by the graph is the number of singletons (people mentioned on only one Web page). Out of all 212 referents, 155 (73%) were singletons. However, these singletons only account for 17% of the pages (155/882). Another trend which is relevant to note is that out of 882 pages, 636 (72%) refer to the majority referent.

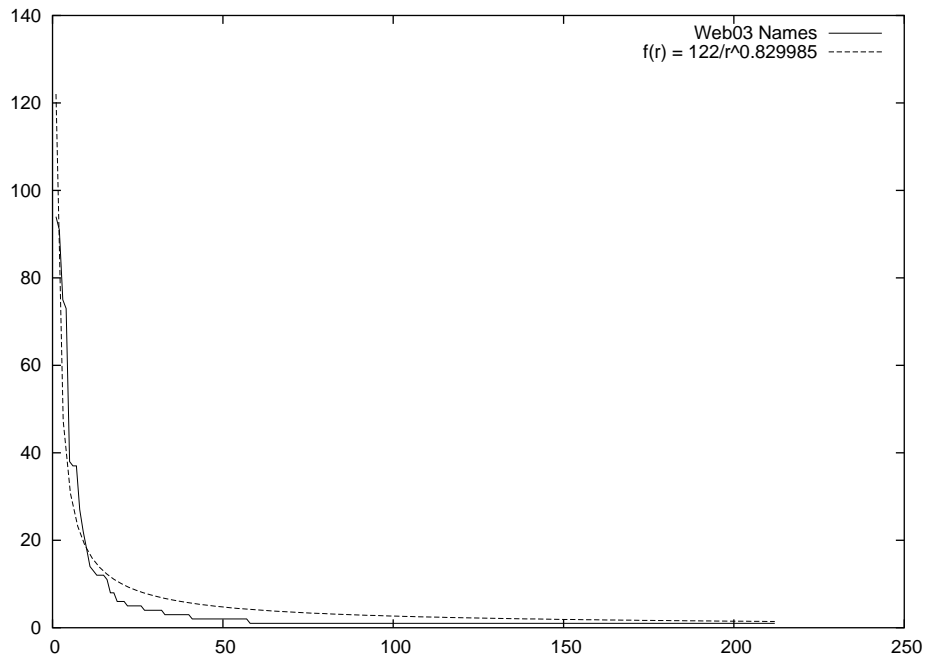


Figure 5.15: Number of Pages per Referent for W03 with a power law fit.

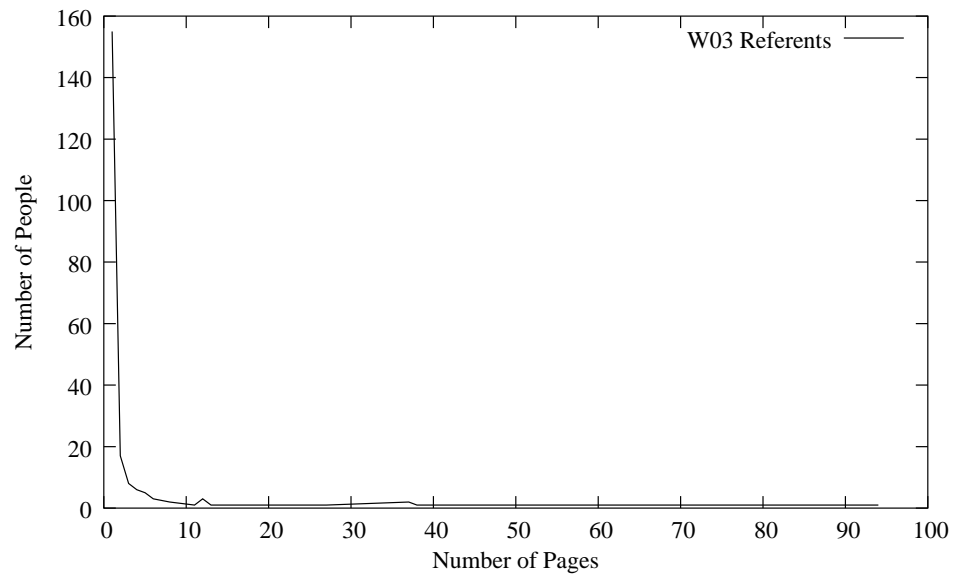


Figure 5.16: Number of Referents with a given number of pages for W03.

## 5.6.2 Unsupervised Clustering

Following the method proposed in the previous sections, this section explores cross-document coreference for real polyreferent names. As with the celebrity pseudonyms, the first set of experiments concerns the unsupervised disambiguation of Web pages to provide single-referent Web page clusters.

Figure 5.17 presents results on unsupervised clustering for the polyreferent names in W03, with two-fold cross validation, finding the optimal stopping threshold on one partition, and using that stopping threshold weights on the other partition. The Cluster Centroid method achieves the highest performance overall for both the TF-IDF and NNP+MI weighting methods, with Group Average the next after that.

Overall, the Cluster Centroid method is significantly better than Group Average clustering and Single Link clustering<sup>3</sup> (and between Group Average and Single Link clustering, there is no significant difference in performance). There is no significant difference in performance between the use of different context windows (50 word, 100 word, or entire document) for the creation of context vectors. Nor is there a significant difference in performance between using TF-IDF and the NNP+MI weighting methods. The estimated stopping thresholds perform significantly worse than the oracle, with a 1% decrease in performance.

The above performance is calculated using the B-Cubed measure which takes into account all of the referents for a given polyreferent name. For many user scenarios, there is one majority referent which takes up most of the Web pages. Across all of the names,

---

<sup>3</sup>In this and the following tests, a paired t-test is used, with  $p = .95$ .

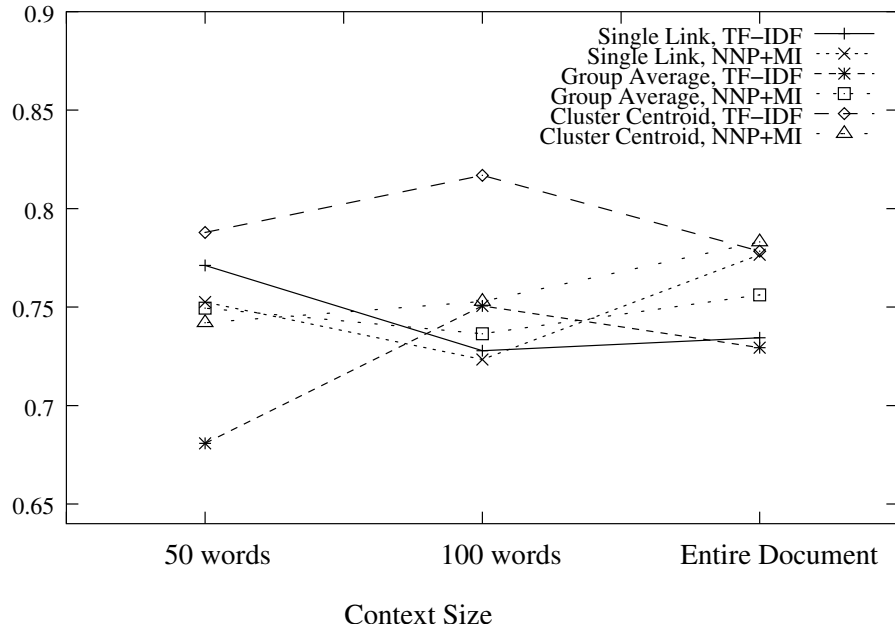


Figure 5.17: Unsupervised Clustering of Manually Found Polyreference Cases using two-fold cross-validation. For each model, the stopping threshold is optimized on one half of the collection, and used on the other half. Cluster Centroid performs significantly better than the other clustering methods.

636/882 pages refer to a majority referent, and these majority referents are therefore quite important with respect to the overall performance. Overall performance is lower on the majority referents than on the dataset taken as a whole, averaged across clustering methods, context sizes and weights (75% to 73%), again using two-fold cross-validation. This drop in performance is significant.

Figures 5.18 and 5.19 show the effect of varying the stopping threshold for each of the clustering methods. Figure 5.18 illustrates performance with the NNP+MI weighting method, while Figure 5.19 illustrates performance with TF-IDF weighting.

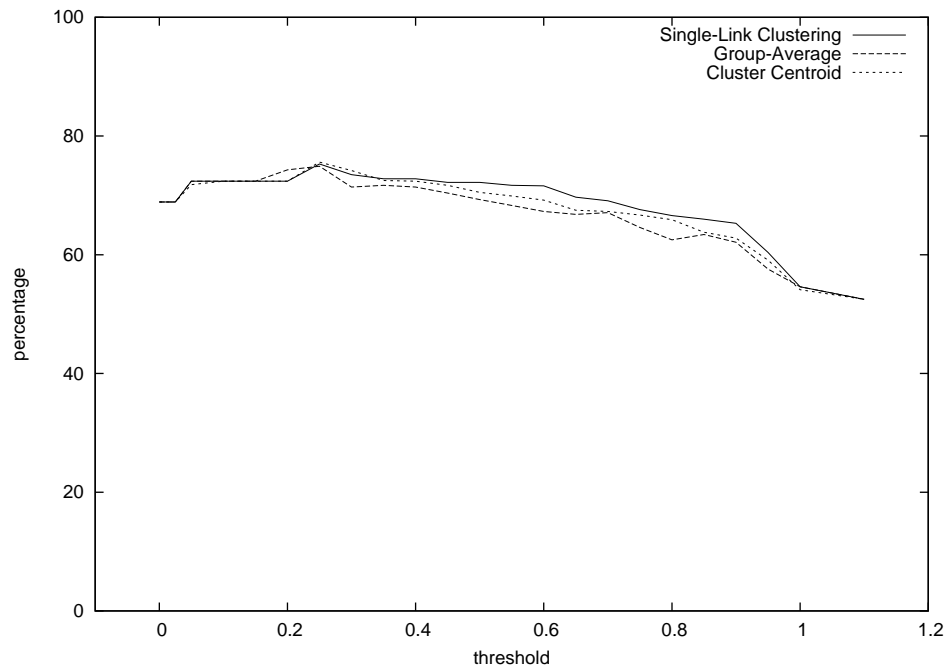


Figure 5.18: Stopping Thresholds for Clustering Methods over the Polyreference Test Set using NNP+MI weighting.

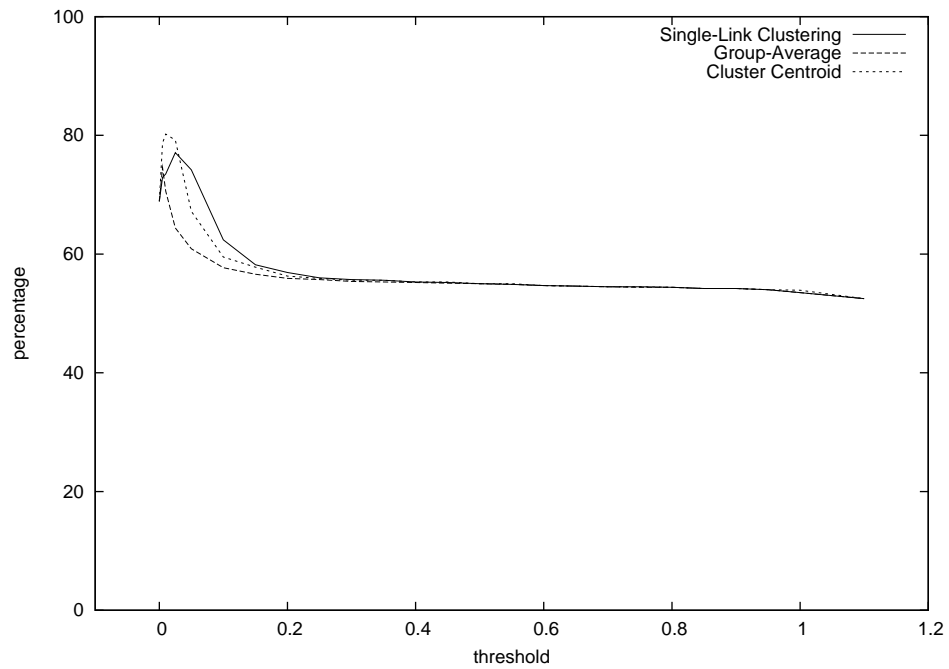


Figure 5.19: Stopping Thresholds for Clustering Methods over the Polyreference Test Set using TFIDF weighting.

### 5.6.3 Seed Clusters from Fact Extraction

As was done in the previous section, for these polyreferent cases, features were extracted from the pages in order to form seed clusters to which the remaining pages were assigned. However, unlike the previous section, there were dramatically fewer total extractions per referent. For the IP152 development set, there was an average of 23 pages with features on them per referent, and for the IP152 test set, there were an average of 25.1 pages with features on them per referent. In contrast, for the W03 set, there were an average of **.72** pages with extracted features on them per referent.

This is a dramatic difference from the Web celebrity case, and when run on the Web polyreferent names, gives results which do not beat the baseline. There are several likely causes for the different rates of extracted features for non-celebrities as compared with the Web celebrities.

First, there are fewer pages per referent for the real polyreference case than there are in the pseudoname case. Unlike the pseudoname case (but similar to John Smith), there are frequent numbers of singleton pages. On average, there were 4 pages per referent for the polyreferent names. However, the baseline difference between the number of pages per referent for the polyreference case and the number for the pseudoname case does not fully account for the difference in the number of pages with extracted features. The percentage of pages with extracted features for the IP152 development and test set are 0.46 and 0.50 respectively, while for W03 is it 0.17.

Second, it is likely that celebrities and non-celebrities are discussed on the Web in very different ways. The kinds of documents that discuss celebrities are very different from

pages that mention non-celebrities. There are few biography pages about non-celebrities. Non-celebrity mentions often occur in a setting in which they are not the main focus, for example a page of marathon results, a listing of employees for a particular company, or a posting to the comments page of a product.

Third, since celebrities are more central topics, the highest ranked Google pages are more likely to be topical. Celebrities are a major focus of conversation on the Web, and therefore the high-ranked pages returned by Google in response to a query on their name center around them. In contrast, for non-celebrities, the highest ranked pages returned by Google mention the individuals only in passing. Thus, Google works quite well for finding celebrities, but not nearly as well for finding biographical information about the rest of us.

Furthermore, non-celebrities typically only occur in contexts where they are uniquely identified by locality to that context. Since celebrities appear in a sort of global context, disambiguation is meaningful. However, non-celebrities typically only occur in contexts in which the audience is expected to be able to disambiguate the individuals using the context.

No matter what the cause, this decrease in the number of pages with extracted information per referent leads to a drastic decrease in clustering performance with the added extracted facts (Figure 5.20). Since there was, on average, less than one page per referent with extracted features, seed clusters could not be built for each referent. Even if the seed clusters could be built, this rate of feature extraction would result in little effect from extracted features, as features would only very rarely help to merge pages (there typically isn't even one extracted feature per referent).

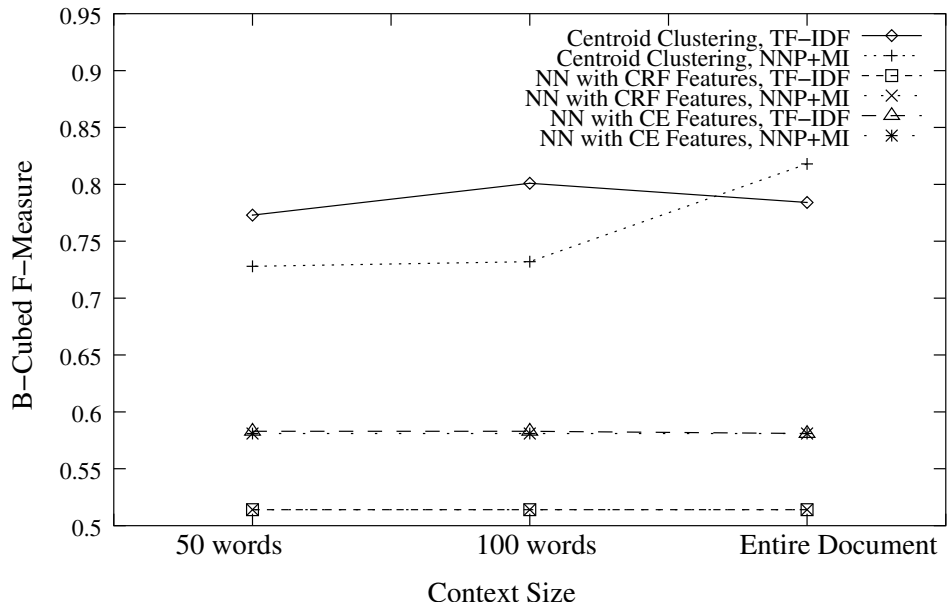


Figure 5.20: Unsupervised Clustering of Manually Found Polyreference Cases with Extracted Features. Unlike in the pseudoname case, extracted features do not appear to help in clustering.

## 5.7 “John Smith” Experiments

In addition to the Web corpora examined in the prior sections, this section looks at system coreference disambiguation performance on the “John Smith” Corpus. This corpus, introduced in [Bagga and Baldwin, 1998], is made up of 197 articles from New York Times printed between 1996 and 1997, which contain the name John Smith, possibly with a middle initial or middle name. Those authors manually resolved all of the “John Smith” references in the articles, and found that out of the 197 documents, there are 35 different John Smiths named, where 24 appeared only in one document, and the 11 other John Smiths took up the remaining 173 documents.

The baseline for the task is B-Cubed F-Measure of 37%, where all documents are in one cluster (Precision 23%, Recall 100%). If all documents are placed in separate clusters,

the baseline performance is 30% (Precision 100%, Recall 17%). Figure 5.21 demonstrates that on this corpus of John Smith articles, the Centroid Clustering outperforms Single Link and Group Average methods. For the TF-IDF weighting scheme, Centroid Clustering method achieves the highest performance at 92% in comparison with 86% for single-link clustering and 85% for group-average. 92% performance on the John Smith task is the highest reported result on this corpus, with the comparison being to 81% for on-line Single Link clustering in [Bagga and Baldwin, 1998], and 88% for Group Average clustering reported by [Gooi and Allan, 2004].

[Niu et al., 2004] purport to give results on the “John Smith” corpus, but it appears to be a subset of the John Smith corpus. Their corpus has 68 documents with 29 different referents, in contrast to the corpus used for these experiments, which has 197 documents and 35 referents.

One of the trends revealed by data analysis is that Centroid Clustering is able to pick out sparse clues more effectively than Group-Average and Single-Link Clustering. In particular, “Captain John Smith” appears in passing in a number of articles. While these passing references are missed by Group-Average Clustering and Single-Link Clustering, Centroid Clustering connects them. Since passing references, where few words are directly about the target referent, are very sparse, they might match better to a cluster mean vector than they would with any particular document.

The high performance of Centroid Clustering relative to the Group Average and Single Link Clustering methods on this task supports the use of Nearest Neighbor clustering as a method for integrating features into the clustering method.

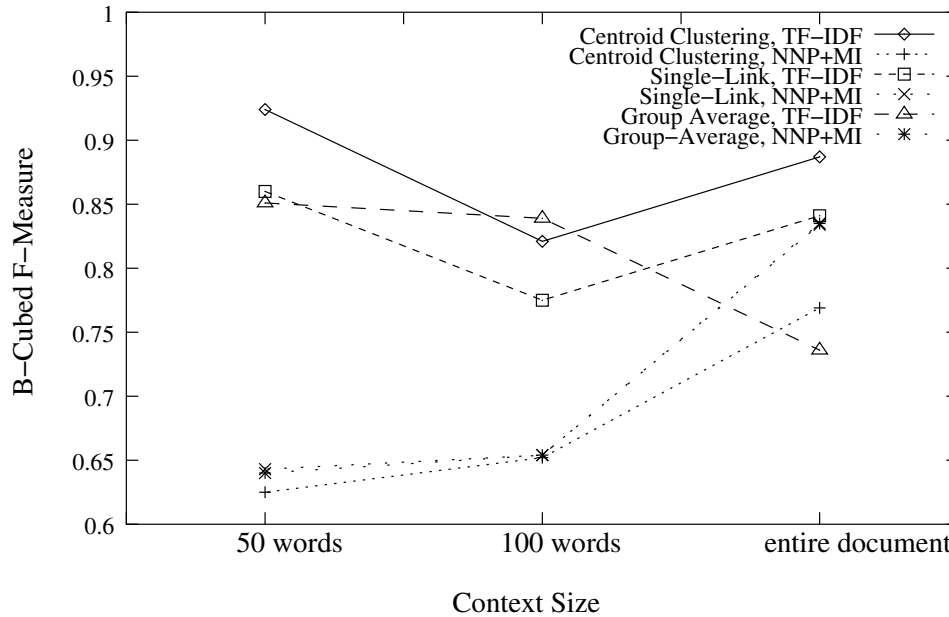


Figure 5.21: John Smith corpus coreference resolution. The 50-word context using Centroid Clustering and TF-IDF achieves the best performance.

On this corpus, TF-IDF for weighting term vectors is more effective than NNP+MI. NNP+MI performance is especially bad for small context windows and improves with larger context windows. This suggests that the likely cause of the deficits in performance of NNP+MI are that in small contexts, eliminating terms which are not either proper nouns or “relevant words” leads to extremely sparse term vectors which prevents formation of coherent clusters. Newswire is cleaner than web pages, and so in general more words are relevant. For more noisy corpora such as the Web corpora discussed in this section, the filtering of NNP+MI improves performance.

Figures 5.22 and 5.23 show the effect of varying the stopping threshold for each of the clustering methods. Figure 5.22 illustrates performance with the NNP+MI weighting method, while Figure 5.23 illustrates performance with TF-IDF weighting.

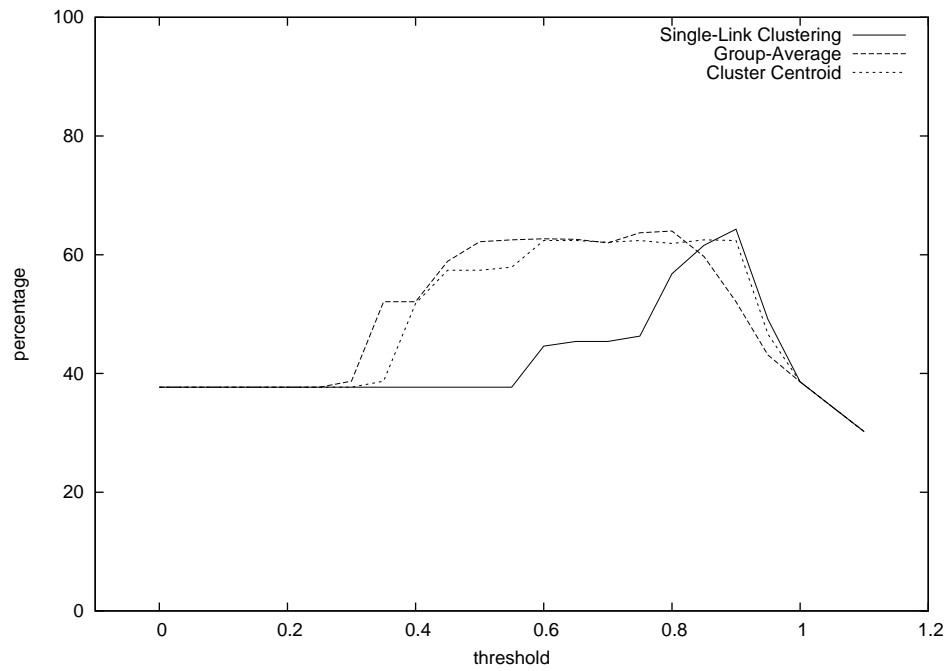


Figure 5.22: Stopping Thresholds for Clustering Methods over the John Smith Corpus using NNP+MI weighting.

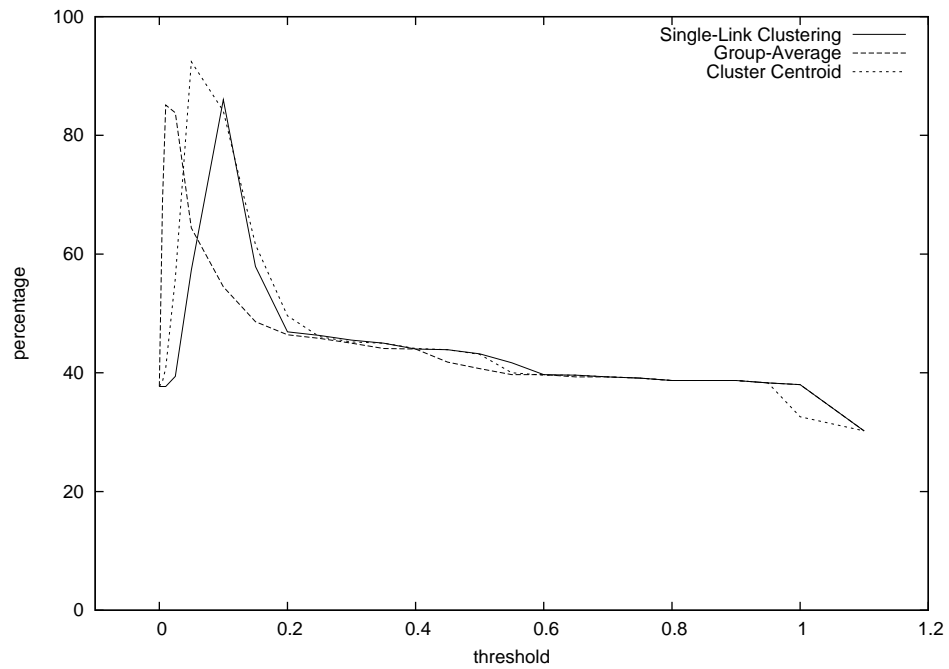


Figure 5.23: Stopping Thresholds for Clustering Methods over the John Smith Corpus using TFIDF weighting.

### 5.7.1 Fact Extraction for “John Smith”

For the “John Smith” pages, extraction of the five facts given in Chapter 4 works poorly for clustering, yielding performance no higher than the baseline. As with the Web polyreference case, part of the reason stems from a dearth of extracted facts on which the algorithm could work. Only .21 of the “John Smith” pages had facts extracted from them (compared with .45 and .5 for IP152 development and test sets, and .17 for W03). [Niu et al., 2004] report gains using extracted facts on the “John Smith” corpus and much of the difference between the systems can be attributed to their information extraction system. They use InfoExtract for extraction, a hand-built commercial extraction system<sup>4</sup> which extracts around 30 different types of facts about individuals. Their increased coverage over the space of facts as well as their increased recall and precision are likely reasons for their increased performance on the “John Smith” task.

## 5.8 Genre Effects on Cross-Document Coreference Resolution

One of the trends that comes across the most clearly in these experiments is the dramatic effect genre has on the ability of systems to perform cross-document coreference. In particular, people in newspapers, celebrities on the Web, and non-celebrities are all discussed in different ways. When celebrities appear on the Web, they frequently have birth information or occupation, especially when they are the focus of the Web page. Newspapers make a concerted effort to disambiguate people when they appear, typically presenting rele-

---

<sup>4</sup>Which is currently sold by Janya Inc. under the name Semantex.

vant properties such as occupation, age, place of residence, or employer, but not necessarily with the same features that appear when celebrities are discussed. These corpus differences can be observed by the performance of the fact extractors on the various corpora. While the extractors were able to find facts on around .5 of the celebrity pseudoname pages, only .2 of the newspaper pages had these facts, and only .15 of the Web polyreference pages did.

When non-celebrities appear on the Web, they typically are not disambiguated at all by the context. Web page authors may neglect to do this either because they suspect that the context performs all the disambiguation that is necessary (e.g. a John Smith listed on a page for “Bob’s Tire Service”), or alternatively with the supposition that these minor players in the Web space will not appear on any other page (and so attempting to cross-reference them requires too much effort). This latter hypothesis may be correct, especially with the observed Zipfian distribution of pages per referent, such that 73% of all referents are singletons (that is 73% of all referents only appear on one Web page).

This analysis carries over to the world of business, where in fact it appears in a more graduated setting. The most famous of executives are treated like celebrities, and the same kinds of biographical information can be found about them. As a reader inspects and searches for increasingly less famous people in the business world, the corresponding information about these people becomes more scarce, with semi-famous executives meriting a mini-biography with a list of recent employment positions, and upper-level executives garnering only a job description, and all the rest getting only a side-mention.

In light of this, there are a number of observations and conclusions from this research. First, and perhaps foremost, the use of synthetic test sets for polyreference may

produce misleading results. Second, for the problem of Web-based coreference resolution, the extracted features that work for celebrities on the Web do not appear to work for non-celebrities or people in news articles, because those features are simply not present. Further work on disambiguating and tracking non-celebrities on the Web must rely on wider-context information, such as context of the Web page (e.g. “Bob’s Tire Service in Detroit, MI”) which can place the individual, pages nearby in hyperspace which can serve this purpose, and the individuals with which the person in question appears. By disambiguating an entire cohort, a particular person may be more easily placed.

Finally, these results may suggest a more general message about genre effects on natural language processing. Most natural language processing applications have been tested and trained on news text, for example the extensive literature on part-of-speech tagging and parsing the Wall Street Journal. While there are exceptions (e.g. recent parsing efforts on speech data), this research suggests that a switch in genre to Web pages from news text may have dramatic effects, and that in order to perform well on Web pages, more research is necessary.

## 5.9 Related Work

There is a long history of clustering methods used for unsupervised learning, and in natural language processing, clustering has been an important tool. Clustering has been used for a variety of lexical analysis tasks including word similarity [Brown et al., 1992], [Pereira et al., 1993] and word sense disambiguation [Schutze, 1998]. Additionally, apart from cross-document coreference, there has been extensive work in clustering for NLP ap-

plications of which the closest is [Cardie and Wagstaff, 1999], where a clustering model is used to resolve intra-document coreference. The above list is only a very small subset of the related clustering work in natural language processing. For a thorough recent review of clustering work in natural language processing see [Pantel, 2003].

[Bagga and Baldwin, 1998] initiated the use of clustering for cross-document coreference, and since the publication of [Mann and Yarowsky, 2003], there has been a good deal of work on this problem. [Gooi and Allan, 2004] gave the first results suggesting that Group Average clustering gives better performance than Single Link clustering. Many groups have looked at alternate feature sets and clustering algorithms. [Bekkerman and McCallum, 2005] use the information bottleneck approach. [Li et al., 2005] create a generative probabilistic model of the entire corpus, and estimate the assignment of mentions to referents which maximizes the entire corpus. [Pedersen et al., 2005] use the method of Repeated Bisection, a hybrid divisive/agglomerative algorithm to cluster co-referent documents. [Fleischman and Hovy, 2004] estimate a Maximum Entropy model for document-document similarity.

Perhaps the most closely related work to this chapter is [Niu et al., 2004]. That paper also uses a Maximum Entropy document similarity model to judge distance between papers. As part of the distance measure, one feature is a match or mismatch between extracted biographic facts between people on distinct pages, which is weighted by the type of feature. Thus a match on certain features (e.g. spouse name) can have a greater impact than a match on others (e.g. birthplace). Along with a better similarity model, their system also benefits from a hand-crafted extraction system which extracts many more biographic

facts than presented in this chapter (e.g. age, employee-of, and employer-of). Their system achieves high performance on two newspaper disambiguation tasks. The first is celebrity pseudoreference disambiguation (88%) and the second task comprises two real polyreference disambiguation tasks (85% and 96%). The high performance of this system on real polyreference tasks supports the work in this chapter on fact extraction for cross-document coreference, though the utility of extracted facts for cross-document coreference on the Web hasn't been shown.

## 5.10 Conclusion

This chapter investigated the problem of unsupervised proper name disambiguation, which is also known as cross-document coreference. The use of extracted features for coreference proposed in Section 5.3.1 and evaluated in Sections 5.5.2 and 5.6.3 is the main contribution of the work, and suggests that extracted facts may be useful as building blocks for other applications. Genre differences between these corpora account for differing performance contributions from extracted facts for Web celebrity pseudonyms (Section 5.5), and Web polyreferent names (Section 5.6), and news text polyreferent names (Section 5.7). While extracted facts yield different levels of improvement, the use of Centroid Clustering and Group Average Clustering as opposed to Single Link Clustering shows significant gains across all three genres.

## Chapter 6

# Time-Bounded Facts and Timeline Construction

Chapter 4 presented the problem of fact extraction and fusion for a set of biographic facts. This chapter explores the property of time-boundedness in certain kinds of facts. As a particular instance of time-bounded facts, the chapter investigates facts which specify roles filled by only one entity during a given span of time. These role-filler facts and the relationships between the facts constitute a semantic network of relationships. In order to extract time-bounded role-filler facts, and semantic networks in general, Section 4.8 proposes a method of cascaded information retrieval, extraction, and fusion. The system takes multiple steps which retrieve pages with the relevant information, extract that information from the corpus, and fuse the extracted facts to generate high confidence information. Extractors further down the pipeline are able to use information discovered in earlier steps to build up an entire network of information. The eventual output is a timeline as can be seen

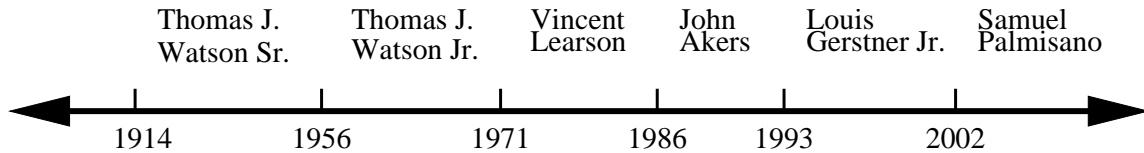


Figure 6.1: Timeline of CEO Succession History for IBM, since its founding in 1914. Tics mark the years of transition between adjacent CEOs (equivalent to Figure 1.2).

in Figure 6.1.

The problem of time-bounded role-filling has a number of interesting aspects. First, the constraint of being time-bounded is a common property of information, and reasoning about absolute and relative times is central to human information processing. Second, role-filling relationships are an important type of information for a wide variety of domains.

## 6.1 Time-Bounded Facts in Text

Most facts about specific entities in the world hold only for a given time span. Some of the biographic facts discussed in Chapter 4 specify these bounding times (e.g.  $\text{born}(\text{person}, \text{date})$ ) while other facts are inherently assumed to be within a particular time context (e.g.  $\text{brother}(x, y)$  can only be true after both of the individuals have been born). This chapter gives a more in-depth look at time-bounded facts, examining methods to determine the time spans of certain facts as well as discover relative orderings of sets of facts. This section reviews a model of temporal events and describes methods for determining time boundaries of events, both directly from text and by inference in cases of correlated time-bounded facts.

Recent work on the TimeML specification language [Pustejovsky et al., 2003] and

[Setzer, 2001] have laid out a framework for annotating temporal events in text. The specification allows for annotation of events and states as well as the relationships between the various events. All times in this specification are defined as a span between two points, where a given event has a beginning point A and an ending point B.

Of particular relevance to the work presented here are TLINKs, which represent temporal relationships which hold between pairs of events or between events and times. Of those, the following four are the most closely relevant for the work presented in this dissertation.

1. Span points (e.g.  $\text{start}(A,x)$ ).
2. Duration ( $\text{during}(A,z) \leftarrow \text{start}(A,x), \text{end}(A,z), x \leq z \leq y$ )
3. Immediate sequence ordering ( $\text{succeed}(A,B) \leftarrow \text{end}(A,x), \text{start}(B,x)$ )
4. Relative ordering ( $\text{after}(A,B) \leftarrow \text{end}(A,x), \text{start}(B,y) \ x \leq y$ )

The other inter-event relationships they specify occur between overlapping events, which are not considered in this section.

### **6.1.1 Extraction Methods for Temporal Information**

Different textual cues exist to extract information corresponding to each of these different types of information. Chapter 4 discussed inter-sentential methods for span point extraction. In that chapter, models for a person's birth and death are trained from example relationships. The models then extract specific dates from text such as "Miles Davis was born in 1926".

Immediate sequence models can be trained in the same way. Given a known set of sequential pairs, a fact extractor can be trained from text with mentions of the two entities in the training data. For example, a training sentence might read “**Joe** succeeded **Bob** as President of Foo Corporation”, where Joe is marked as the succeeding member of the pair, and Bob as the preceding. When given a new pair of undetermined order, this model then chooses between labelings which imply alternate orderings. In this chapter, immediate succession ordering is assumed, but presumably relative ordering models could be trained in similar ways.

### **Duration Models**

More than for previous types of information, there exist a rich set of alternative cues for duration models. First, text may literally specify duration moments (e.g. “Miles Davis was alive in 1950”), and for this set of cues, the above types of statistical models can be applied. Alternatively, the document may be associated with a time- stamp. In that case, the tense of the statement of the fact gives a lot of useful information. If the fact is mentioned in present tense, then the fact can be assumed to hold at the time at which the document was dated. If the fact is mentioned in the past tense, the event occurred before the time-stamp. If the fact is stated in the future or conditional, it occurred (or will occur) after the time-stamp of the document. Another possibility is implicit tense (e.g. “the late trumpeter Louis Armstrong”).

Two places in which this time-stamp might appear are in Web page header stamps such as “last-modified-on” and “date”, and date-line marking of a news article. However, extracting the date from these two contexts is non-trivial. Dates contained in Web page

headers are very noisy, and in a sample of headers collected in May 2005, nearly 50% had no information on their modification date, and of the remaining headers, 50% had modification dates within the past year, suggesting that their modification date might correspond to the time in which the ads or the format changed, not the relevant content. The remaining 25% of headers had modification dates primarily in the past few years, with few having modification dates prior to 2000.

Date-line marking of news articles is typically more reliable than Web page header modification times. However, this marking is not typically present in other genres. On top of this, finding the year and date of a news article is in itself a non-trivial information extraction task, as each different newspaper has a different format for its dateline marking. Finally, it isn't straightforward to train a system by example to extract datelines from text. For these reasons, the development of such a model is left to future work.

### **6.1.2 Reasoning about Temporal Events with Uncertain Information**

As stated above, the temporal model treats all events as having a start point and an end point. When attempting to build a completely specified time representation from extracted information, often only partially specified information is available from the extraction, typically from each of the four types of information as presented above. Time is explicitly specified in some facts (e.g. someone's birthday is the year 1926), while in others it may be unknown or bounded. These bounds on temporal events may be derived by reasoning, for example, given someone's birth year, their year of death is known to be some later date.

Each of these different types of information permits different inference operations.

Knowing the immediate sequence ordering and one of the points in the ordering leads to knowledge of the other point in the ordering ( $\text{succeed}(A,B), \text{end}(A,x) \rightarrow \text{start}(B,x)$ ). Knowing even one point at which an event was true allows inference of relative order ( $\text{after}(A,B) \leftarrow \text{during}(A,x), \text{during}(B,y), x < y$ ). This second principle allows for the heuristic methods applied in Section 6.6.3.

Often instead of incomplete information, a large set of uncertain and partially erroneous information is available. In this situation, the same kinds of constraints may be used to improve the estimates. In the case of immediate sequence ordering ( $\text{succeed}(A,B)$ ), since it is known that  $\text{end}(A,x)$  and  $\text{start}(B,y)$ , the system can synthesize information separately extracted for  $\text{end}(A,x)$  and  $\text{start}(B,y)$  to arrive at a consensus answer for  $x$ . Section 6.6.5 proposes a method to arrive at this consensus using linear interpolation to calculate a better estimate of transition years from extraction of varied start and end dates.

## 6.2 Management Succession

The particular case of time-bounded role-filling relationships that this chapter looks at is that of corporate management. This chapter proposes and demonstrates a method for creating a timeline of CEOs over a company's lifetime. Like biographic fact extraction, this is a relevant problem and one whose output would be useful to a broad community. Unlike biographic facts, which are often collected in large databases of people, there are no large public repositories of this information<sup>1</sup>. Management succession data are scattered across many sources, and often no more than one CEO is present in a given

---

<sup>1</sup>Dunn and Bradstreet sample this information yearly, and SEC filings also contain partial information of this sort.

document.

There are a number of complexities in this task, beyond the inherent complexity of the problem of time-bounded role-filling relationships in general. The exact name of the highest position within a corporate hierarchy many change over time. In the earlier part of the century, the title of the highest person in management was typically 'President'. In the later part, with the move to corporations managed by a board of trustees, the highest position is typically the Chief Executive Officer (CEO), who also sometimes holds the title Chairman. In some cases, a corporation may have multiple Presidents each of whom is distinct from the CEO, and where the Chairman of the Board is yet a different person. In these situations it is difficult for a person to determine the highest ranking officer, not to mention a computer. This chapter considers only CEO extraction, and does not attempt to find the true "highest ranking officer", but even there the CEO position title may be "CEO and Chairman" or "President and CEO".

Additionally, like the familial relationships discussed in the prior section, there is a large amount of flexibility in referring to people, and potential confusion can result. Anheuser-Busch CEO August A. Busch III (1975-2000) was the third August Busch to run the company and the fifth Busch. This phenomenon is not unusual for family-owned businesses. There are seven McGraws that have run the McGraw-Hill company, with most recent being Harold "Terry" McGraw. Added to that is the confusion between the CEO's last name and the name of the company. For the largest conglomerates (e.g. General Electric), this isn't typically a problem.

## Web Data on CEOs

As in the previous sections, a set of facts was manually collected for training and evaluation. Appendix C lists the corporate succession facts that were manually gathered for training and evaluation purposes in this thesis. The entire Web was used to create this set of facts using all pages on the Web (not necessarily those collected in the next step) from companies listed in the Forbes 500.

One of the most notable properties of this data is that some of the CEOs are more difficult to find than others. In particular, recent CEOs are widely discussed in Web text, while older CEOs are not. One of the starting points for the work is in CEO extraction, which section 6.5 discusses, proposing a corpus collected for that purpose. Figure 6.2 mean CEO mention is plotted against the last date at which the CEO was known to be in office, and shows that while there is copious information about recent CEOs, earlier CEOs are very infrequently mentioned. One possible reason for this may be that managers are likely to appear on company documents and contemporary newspaper and periodical articles. Managers from the 1970s would have been referred to in the periodicals of that time, but since those documents have not been digitized and posted online, information about earlier CEOs is sparse on the Web.

Because earlier CEOs are mentioned either very infrequently or not at all in our corpus, they are outside the scope of this dissertation. This dissertation focuses on information which is redundantly accessible and available, and provides methods for extraction and analysis given this base condition. Our work primarily focuses on CEOs from the recent past, in particular CEOs since 1990.

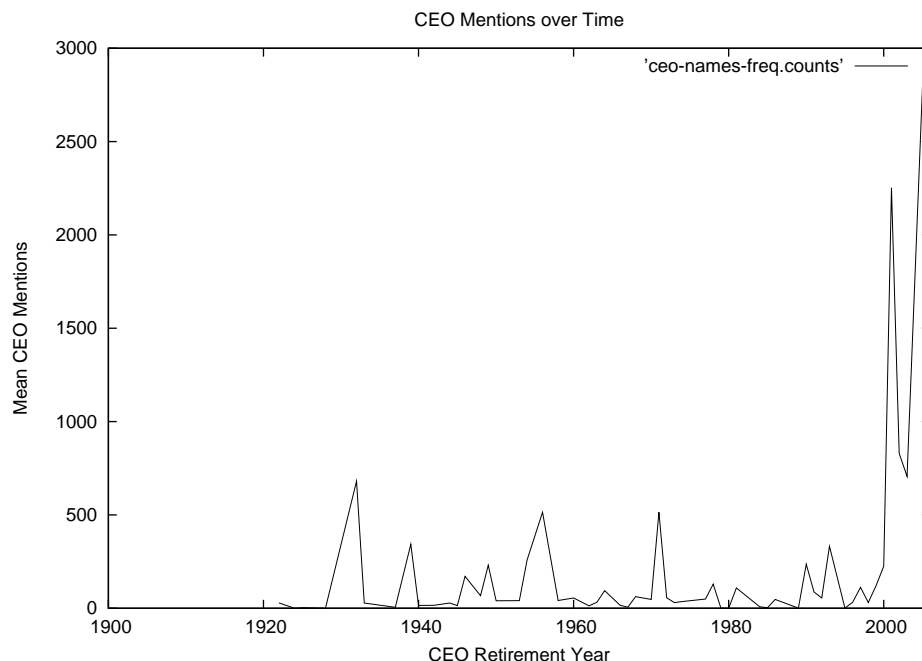


Figure 6.2: CEO Mentions over time. CEOs from the past decade are mentioned much more frequently than CEOs from earlier times.

### 6.3 Corporate Entities

This chapter treats corporations as a basic entity, in much the same way that Chapter 4 treated people. The crucial shared linguistic quality between celebrities and corporations is that their names are both relatively unambiguous<sup>2</sup> Inference is made significantly easier because almost every time the name of a celebrity or company is mentioned, those lexical tokens refer to the entity in question, and the sentence can be assumed to be relevant to that entity. For polysemous common nouns, this is not the case. Other entities where these methods are likely to work are other types of organizations (e.g. governing bodies or athletic teams), locations, and artifacts with titles (e.g. books or movies).

---

<sup>2</sup>For celebrities, even if the name is not unique, on the Web the celebrity referent for the name outnumbers all other referents so that the name is virtually unambiguous. For corporations, ambiguity can come from subsidiaries or from matching the founder's name.

There is biographic information for corporations which is similar in nature to the biographic facts extracted for people. One key fact about companies is the date of a company being founded, and to show that this information could be extracted a set of experiments was designed. In order to extract this information, first a set of company founding dates (18 companies and their founding dates) were collected manually from the Internet. This took around 2 hours. Next the system automatically downloaded 1000 pages for each of the companies in the set.<sup>3</sup> Six companies were assigned to be training examples, 4 development and 8 testing. Following the method proposed in Section 2.2, a extraction system was trained, and then answers were extracted for all of the companies. Here unigram and bigram features were used. For this new system evaluated on the development data the fusion accuracy was 50% and the fusion MRR was 75%. These results are comparable to results in Chapter 4 for experiments with similarly few training examples (see Section 4.5.6).

## 6.4 Cascaded Extraction and Fusion for Timeline Creation

The remainder of the chapter will build up in stages a series of models for inducing a CEO timeline for a particular company. The cascaded information retrieval, extraction and fusion system is as follows:

1. The system starts with a given company, and downloads a collection of pages. From this set of pages, managers are extracted and through fusion, a set of candidates is selected. (Section 6.5)

---

<sup>3</sup>1000 pages is much larger than the 150 for celebrities. In the case of companies, “biographic” type of information is much more sparse than in the case of people.

2. The top candidate is then treated as a known CEO, the next top candidates are placed into pairs with this known CEO, and a second set of documents is downloaded. From this set of documents, an ordering of two candidates is selected. (Section 6.6.2)
3. The CEOs in this ordering are inputs to a next stage in which the tenure midpoint for each CEO is estimated. This provides a check to the ordering proposed in the previous step. If the ordering by the tenure midpoints contradicts the ordering by succession, the tenure ordering is selected. (Section 6.6.3).
4. Start and end years are extracted for each of the CEOs in the succession relationship (Section 6.6.4). Those years are then combined to arrive at a transition year estimate (Section 6.6.5).
5. Finally, a visualization is created from all of the available information. (Section 6.6.6).

As each model in the above sequence is described, experiments on the development set will show performance. Section 6.7 will present results of using this system on the test set companies.

## **6.5 CEO Extraction**

The problem of manager extraction examined in this section, defined as finding the complete list of chief executives (CEOs) for a particular company (e.g. Boeing), is the first step in determining management succession. A complete or partial list provides information which can then be filtered and reordered. To solve the problem of CEO succession, a system was built in the same way as the biographic fact extraction system proposed in Section

2.2 and demonstrated in Chapter 4. Starting with examples of the relevant facts, the system generates a *hook* and queries Google to create a set of *hook documents* or *training set*. The training set is then automatically annotated with the known *targets* and CRF extractors are trained. In extraction, a hook is constructed and Google is queried as before to construct a *retrieval set*. The CRF is then applied over the retrieval set to extract the desired information, and the facts are fused across all of the documents to come up with the consensus answer.

This problem is different from biographic fact extraction in several ways. First, the hooks here are more complicated than person names. Instead of being people, the hooks are company-positions, and often the company name is complicated to identify. For example, often a Boeing Corporation subsidiary (e.g. “Aviation Partners Boeing”) is confused with the parent company. The complexity of the hooks has repercussion in automatic annotation, where instead of simply tagging each name mention in the text, each company-position phrase must be tagged, a process carried out by the weighted finite-state transducer shown in Figure 6.3. At the same time that the WFST marks up hooks, it also marks up with the “extra” tag words in the hook phrase which might indicate that the phrase is in fact not the true hook.

The above machines mark up sequences of a person and a company-position, and distinguish between person company-position relationships which are instances of the relationship desired, and those relationships which are not. It uses the weighting scheme proposed in Chapter 4, where Shortest Target/Longest Spurious which was shown to be the best weighting scheme. The machine in Figure 6.4 marks up the training data to extract

```

EXTRA = ε:<extra> (<ORGANIZATION>|<LOCATION>|<NNP>|<NNS>)*
ε:</extra>
COMPANY = (<company> EXTRA) | (EXTRA <company>)
TITLE = <title>+ (and <title>)?
MARK-COMPANY-POSITION = (
    Σ/1
    | (ε:<company-position>
      ( (COMPANY+ ('s)? TITLE)
        | (TITLE ((of the) | for)
          COMPANY+))
      ε:</company-position>)
    )*

```

Figure 6.3: WFST which marks company-positions with in-line XML tags. Example output might be : “<company-position> Boeing/COMPANY Chief/title Executive/title Officer/title < /company-position>”. All strings in < and > are semantic tags with which words are annotated, not actual text strings.

this information.

Like biographic fact extraction, these facts concern a constrained set of basic objects: people, titles and companies. To appropriately tag people, a Named-Entity tagger was built which achieves 83% F-measure on person extraction in the MUC7 test set. This is not state-of-the-art performance on the MUC7 test set which is just above 90%, but was sufficient for the task.

Since people are at the heart of the problem, there was additional work needed to resolve name ambiguity. To address this, a simple program parsed each name candidate into the fields of first name, middle name, last name, and honorific. Then the program attempted to unify two names, allowing for middle initials to match appropriate middle names. To match names with nicknames (a ubiquitous problem in person extraction), a nickname list extracted from the Oxygen Web site was used<sup>4</sup>.

---

<sup>4</sup>Thanks to P. Driscoll (2005), personal communication.

```

CEONAME = (<ceofirst>? <PERSON>)? <ceolast> <PERSON> ?
NAME = <PERSON>+
COMPANY-POSITION = <company-position>+
IGN =  $\Sigma/1$ 
IGN2 =  $\Sigma/2$ 
MARK-CEONAME-TRAIN =
     $\epsilon$ :<o> IGN  $\epsilon$ :</o>
    ( (  $\epsilon$ :<position> COMPANY-POSITION/100
 $\epsilon$ :<position>
     $\epsilon$ :<1> IGN2*  $\epsilon$ :</1>
     $\epsilon$ :<ceoname> CEONAME  $\epsilon$ :</ceoname> )
    (  $\epsilon$ :<>false-position> COMPANY-POSITION/100
 $\epsilon$ :<>false-position>
     $\epsilon$ :<2> IGN2*  $\epsilon$ :</2>
     $\epsilon$ :<>false-name> NAME  $\epsilon$ :</false-name> ) |
    ( (  $\epsilon$ :<ceoname> CEONAME  $\epsilon$ :<ceoname>
     $\epsilon$ :<3> IGN2*  $\epsilon$ :</3>
     $\epsilon$ :<position> COMPANY-POSITION
 $\epsilon$ :</position> )
    (  $\epsilon$ :<>false-position> NAME  $\epsilon$ :<>false-position>
     $\epsilon$ :<4> IGN2*  $\epsilon$ :</4>
     $\epsilon$ :<>false-position> COMPANY-POSITION/100
 $\epsilon$ :</false-position> ) )
     $\epsilon$ :<o> IGN  $\epsilon$ :</o>

```

Figure 6.4: WFST which produces training data for CEO extraction. True managers for the given company have previously been annotated with ceofirst and ceolast tags, and other names of people as with the PERSON tag. In the above machine, IGN stands for ignore.

Unlike for biographic fact extraction, there were always multiple correct targets. However, for each company, the database included the complete list of managers and could correctly annotate all training sentences. The impact in extraction and evaluation will be discussed and evaluated in the next section.

### 6.5.1 Development Set Performance

Using the above methods, a system was trained as described above to extract company CEOs using the management history from Anheuser-Busch, McGraw-Hill, Lennar, and Raytheon. As a development set for the Extractors, Boeing, Heinz, Staples, and Textron were used. Weighted confidence estimation (Section 2.4.3) for a CRF trained with positive and negative examples (Section 2.3) was employed. The performance on the development set is shown in three tables, Table 6.1, 6.3 and 6.4, and 6.2. Table 6.1 lists the rank of the correct answers in a list of targets ranked by the confidence method. As can be observed on the list, the first two recovered targets for all of the companies were in fact CEOs. However, the recall of the system is lower than the precision. Typically, only the most recent CEOs were recovered. This reflects the corpus, which is primarily concentrated on current information about companies, as opposed to corporate history.

Table 6.2 shows the normalized score (mean weighted confidence estimates) of the top 10 most confidently extracted CEOs. As can be seen, the normalized scores are comparable across the different companies, which suggests that the confidence estimate works not only on the sentence level but that the levels may be useful for comparing extractions across corpora.

Tables 6.3 and 6.4 list the top 10 target executives retrieved by the system for

CEO	Rank
Boeing	
Philip M. Condit	1
Harry C. Stonecipher	2
Frank Shrontz	9
Thornton A. Wilson	-
Heinz	
Anthony J.F. O'Reilly	1
William R. Johnson	2
Staples	
Thomas G. Stemberg	1
Ronald L. Sargent	2
Textron	
Lewis B. Campbell	1
James F. Hardymon II	2
Beverly F. Dolan	8
Joseph Collinson	-
Royal Little	-
William Miller	-
Robert P. Straetz	-
Rupert Thomson	-

Table 6.1: Rank of Correctly Found CEOs. For every company, the rank of the correct CEOs in the retrieved CEO list is given. For each company, the top ranked answer is in fact a valid CEO for the company.

Person	Position	Normalized Score
<b>Lewis B. Campbell</b>	Textron CEO	0.1362
<b>Thomas G. Stemberg</b>	Staples CEO	0.1111
<b>Philip M. Condit</b>	Boeing CEO	0.1053
<b>Ronald L. Sargent</b>	Staples CEO	0.0879
<b>William R. Johnson</b>	Heinz CEO	0.0648
<b>Harry C. Stonecipher</b>	Boeing CEO	0.0639
Alan R. Mulally	Executive Vice-President of Boeing	0.0563
<b>Anthony J.F. O'Reilly</b>	Heinz CEO	0.0289
<b>James F. Hardymon II</b>	Textron CEO	0.0278
Stephen A. Giliotti	Textron Financial Corp. CEO	0.0259

Table 6.2: Top 10 Most Confident CEO Extractions across all companies. When the retrieved CEOs are ranked by weighted confidence, 8/10 are CEOs for the hook company, which suggests that weighted confidence is reliable across corpora.

Person	Title
Boeing	
<b>Philip M. Condit</b>	Boeing CEO
<b>Harry C. Stonecipher</b>	Boeing CEO
Alan R. Mulally	Executive Vice-President of Boeing
James F. Albaugh	Boeing Integrated Defense Systems CEO
Mike Marino	Aviation Partners Boeing CEO
USA Today	?
James A. Bell	Boeing Interim CEO
William Gates	Microsoft CEO
<b>Frank Shrontz</b>	Boeing CEO
<b>Phillip M. Condit</b>	misspelling
Heinz	
<b>William R. Johnson</b>	Heinz CEO
<b>Anthony J.F. O'Reilly</b>	Heinz CEO
Neil Harrison	CEO of Heinz Frozen Food Company
Joseph Jimenez	CEO of Heinz Europe
Messrs. Berger	?
<b>Anthony J.F. O'Reilly</b>	misspelling
Joe Heinz	?
<b>Heinz CEO William Johnson</b>	bad extraction
Peter Lucas	CEO of Heinz Australia and New Zealand
<b>William R. Johnson Philip G.</b>	bad extraction

Table 6.3: Top 10 Found CEOs for Boeing and Heinz, ranked by confidence weight. People in **bold** were CEOs for that company. Misspellings of CEOs names are also marked in **bold**.

each company. One important thing to note is that people on this list who are incorrectly extracted are often closely associated with the company, though often they are not the chief executive. For Boeing, Alan Mulally is the President and CEO of Boeing Commercial Airplanes, and Mike Marino is the CEO of Aviation Partners Boeing. Setting aside the similarity of these titles with that of the true chief executive, on the Web Alan Mulally is incorrectly referred to as Boeing's CEO, not as CEO of Boeing Commercial Airplanes, a subsidiary. Furthermore, as a testament to the quality of the named-entity extraction, the extracted names are remarkably well-formed, with very few exceptions in the top 10 list.

Person	Title
Staples	
<b>Thomas G. Stemberg</b>	Staples CEO
<b>Ronald L. Sargent</b>	Staples CEO
<b>Thomas Stemberg Staples</b>	bad extraction
Gordon Clapp	?
Martin Hanaka	Staples COO
<b>Thomas Sternberg</b>	misspelling
Willard Mitt	?
David Brock	CEO Media Matters for America
<b>Mr. Ron Sargent</b>	bad name unification
<b>Ron Sargant</b>	misspelling
Textron	
<b>Lewis B. Campbell</b>	Textron CEO
<b>James F. Hardymon II</b>	Textron CEO
Stephen A. Giliotti	Textron Financial Corp. CEO
Sam Licavoli	Textron Industrial Products CEO
Jake Hirsch	Textron Fastening Systems CEO
Steve Loranger	Textron COO
Sam Lakavoli	Textron Industrial Products CEO
<b>Beverly F. Dolan</b>	Textron CEO
Mary Howell	Textron Executive Vice President
William B. Sturgill CEO	President of Golden Oak Mining

Table 6.4: Top 10 Found CEOs for Staples and Textron, ranked by confidence weight. People in **bold** were CEOs for that company. Misspellings of CEOs names are also marked in **bold**.

### 6.5.2 Biographic Fact Extraction

The methods presented in Chapter 4 for biographic fact extraction were additionally evaluated on the development set of CEOs, except that for these pages the hook query included both the person’s name and the company name (e.g. “Harry-Stonecipher Boeing”). The performance on this set of people was significantly lower than for the celebrity list given earlier. The results were 25% correct for birthday, 37.5% correct for birthyear, and 0% correct for birthplace, though all of the occupations were correct descriptions of this set of CEOs.

There are two primary reasons for the lower performance. First, the celebrities from the previous chapter are mentioned much more frequently than the CEOs in this chapter. As was shown earlier, a smaller retrieval set leads to decreased performance in extraction and fusion. For each of the celebrities, at least 150 pages mentioning them were found on the Web. For these CEOs, there were often far fewer Web pages available which mentioned them in the context of the company name.

On top of the lower number of mentions on Web pages, the Web is less likely to mention birthdays and associated biographic information for CEOs. Their occupations (as can be seen in Table 6.8) were found on the Web, but for most CEOs, birthplaces and birth dates are not as generally reported as it is for celebrities. However, often their ages are, and extractors for this information might be worthwhile, and are left for future work.

Heinz	<b>Anthony O'Reilly</b>	May 7	2.917
Boeing	<b>Harry Stonecipher</b>	May 16	0.994
Staples	Thomas Stemberg	June 30	0.014
Textron	Lewis Campbell	September 11	0.012
Textron	James Hardymon	May 1	0.01
Staples	Ronald Sargent	September 17	0.009
Boeing	Philip Condit	August 28	0.004
Heinz	William Johnson	June 24	0.004

Table 6.5: Extracted Birthdays for CEOs

Heinz	<b>Anthony O'Reilly</b>	1936	2.923
Boeing	<b>Harry Stonecipher</b>	1936	1.439
Boeing	<b>Philip Condit</b>	1941	1.163
Heinz	William Johnson	2005	0.188
Staples	Thomas Stemberg	2002	0.056
Textron	James Hardymon	1992	0.018
Textron	Lewis Campbell	2004	0.009
Staples	Ronald Sargent	2002	0.006

Table 6.6: Extracted Birthyears for CEOs

Heinz	William Johnson	Pittsburgh	45.099
Heinz	Anthony O'Reilly	Pittsburgh	21.337
Textron	Lewis Campbell	Capital	1.991
Staples	Thomas Stemberg	Dover	1.486
Staples	Ronald Sargent	Boston Massachusetts	0.744
Boeing	Philip Condit	Stuttgart	0.532
Textron	James Hardymon	Providence	0.005

Table 6.7: Extracted CEO Birthplaces

Boeing	<b>Harry Stonecipher</b>	executive	7.514
Boeing	<b>Philip Condit</b>	chairman	6.07
Textron	<b>Lewis Campbell</b>	chairman	5.936
Heinz	<b>William Johnson</b>	chairman	5.375
Heinz	<b>Anthony O'Reilly</b>	businessman	5.205
Staples	<b>Thomas Stemberg</b>	chairman	4.622
Textron	<b>James Hardymon</b>	chairman	0.851
Staples	<b>Ronald Sargent</b>	retailer	0.657

Table 6.8: Extracted CEO Occupations

## 6.6 Extracting Temporal Information

The prior section presented a method for extracting managers by training an extraction system with a set of examples following the methodology first described in Chapter 4. This section examines automatic ways to provide a CEO timeline. Ideally, the timeline would give the ordering of CEOs, marked up with start and end dates for each of the positions, or in particular the dates at which the managers holding each post changed. Clearly, this task is extremely difficult, and often the requisite information does not exist in the data. As a back-off to this information, the system also presents a relative ordering of the CEOs.

### 6.6.1 Co-sentential Year Baseline

As discussed in Section 6.8, prior to the work presented in this dissertation there has been very little work in temporal ordering of facts across multiple documents. This section presents baseline methods which find temporal information for CEOs.

One potential method for choosing start and end years might be to find all years which appear co-sententially with a given CEO, and then pick the earliest year as start year and the last year as the end year. On the development set, this method gets none of the start or end years correct.

Another method might be to choose the most frequently appearing year as either a potential start or end year. On the development set, the most commonly occurring co-sentential year for one person is a end-year, and for another is a start-year, giving the “most frequent co-sentential year” system a  $1/8$  accuracy for both start and end tenure years.

Finally, for relative ordering the mean of the co-sentential years could be used. In this case, the means of the co-sentential years proves to be correct in all four ordering of the development set.

### **6.6.2 Succession Facts for Ordering**

The methods presented in Section 6.5 presented a way to reliably extract one CEO for a given company, with the majority of other CEOs being ranked in the top 10 list. Typically, the most recent CEO was first ranked, though this was not always the case.

This section presents a method for using a known CEO candidate to pick out a CEO which either is before or after the chosen CEO and suggest a relative ordering choice. The next section will discuss another method, suggesting a relative order ranking given a pair of candidates with higher accuracy than the method presented in this chapter. The essence of the method discussed here is to build a model of succession and given a list of candidates rank them by likelihood of succession and precession with respect to a known CEO.

In more detail, the basic model is a succession model (`succeed(A,B)`), where for a given pair A and B the model evaluates how likely it is that the relationship is asserted in a span of text. It is trained from all of the succession facts available from the annotated companies. These facts consist of information such as `succeed(Daniel P. Burnham, William H. Swanson)`, where Swanson succeeds Burnham. Following the framework proposed in Section 2.2, a corpus of sentences with both names is generated, the sentences are annotated, and then a CRF model is trained from those examples. The finite state transducer which performs the annotation is shown in Figure 6.5.

```

IGN =  $\Sigma/1$ 
IGN2 =  $\Sigma/2$ 
CUR = <cur> <cur>
PREV = <prev> <prev>
MARK-SUCCESSION =
  (  $\epsilon$ :<0> IGN*  $\epsilon$ :</0>
     $\epsilon$ :<_cur> CUR  $\epsilon$ :</_cur>
     $\epsilon$ :<1> IGN2*  $\epsilon$ :</1>
     $\epsilon$ :<_prev> PREV  $\epsilon$ :</_prev>
     $\epsilon$ :<0> IGN*  $\epsilon$ :</0> )
| (  $\epsilon$ :<0> IGN*  $\epsilon$ :</0>
     $\epsilon$ :<_prev> PREV  $\epsilon$ :</_prev>
     $\epsilon$ :<2> IGN2*  $\epsilon$ :</2>
     $\epsilon$ :<_cur> CUR  $\epsilon$ :</_cur>
     $\epsilon$ :<0> IGN*  $\epsilon$ :</0> )

```

Figure 6.5: FST which annotates for succession information

In extraction, the system starts with a known CEO and a list of candidate successors or predecessors. The system then retrieves documents, and for each of the pairs (A,B) marks up the sentences in two ways, the first for the fact succeed(A,B) and then for the fact succeed(B,A), and evaluates the CRF for these two facts. This evaluation of the CRF is done by taking the confidence estimate of the sentence having labels which correspond to the proposed ordering.

Tables 6.3 and 6.4 show the top 10 CEOs extracted for the four development set companies. For the CEO extracted from Boeing as shown in Table 6.3, Philip Condit is the known CEO, and the other candidates are Harry Stonecipher, Alan Mulally, James Albaugh, Mike Marino, USA Today, James Bell, William Gates, Frank Shrontz, and Phillip Condit. The orderings checked are:

- successor(Philip Condit,Harry Stonecipher)
- successor(Harry Stonecipher, Philip Condit)

Sentence	Philip Condit first	Harry Stonecipher first
<b>Harry Stonecipher</b> , who <i>took over</i> as chief executive last week <i>after the resignation of Philip Condit</i> , told The Telegraph that the group was still looking for a transatlantic deal and described BAE as “the favorite partner”.	.67	.02
Last week Boeing’s new chief executive <b>Harry Stonecipher</b> voiced confidence the deal would go through despite ethics problems that <i>pushed his predecessor Philip Condit</i> to resign in December	.89	.28

Table 6.9: CRF Confidence estimates for management orderings. The correct ordering (Philip Condit then Harry Stonecipher) get .67 and .89 probability, more than the incorrect orderings which get .02 and .27 probability. Crucial lexical context for the two examples might be “took over”, and “after the resignation of” in the first and “pushed his predecessor” in the second. Because the two orderings have different features marked in the sentences, the probabilities for the different orderings do not sum to 1.

- successor(Philip Condit, Alan Mulally)

and so on.

For the same sentence, two separate orderings were evaluated. Since each sentence has different features, the label estimates for each CRF are different and thus do not sum to one. The results of each of these orderings are shown in Table 6.10.

Table 6.11 gives the orderings returned by the system, where 3/4 are correct. For the instance the system got wrong, Textron, the correct preceding manager is extracted but the order is reversed. In that case, the score for the correct ordering is slightly worse than the correct score. Textron is also the company for which there is the least amount of information available.

Proposed Order	Score
<b>Philip Condit followed by Harry Stonecipher</b>	6.744
Harry Stonecipher followed by Philip Condit	5.037
Phillip Condit followed by Philip Condit	3.488
Philip Condit followed by Phillip Condit	3.253
William Gates followed by Philip Condit	2.848
Philip Condit followed by Alan Mulally	1.290
<b>Frank Shrontz followed by Philip Condit</b>	1.173
Philip Condit followed by William Gates	0.991
Philip Condit followed by Frank Shrontz	0.857
Alan Mulally followed by Philip Condit	0.176
Philip Condit followed by USA Today	0
USA Today followed by Philip Condit	0
Philip Condit followed by Mike Marino	0
Philip Condit followed by James Bell	0
Philip Condit followed by James Albaugh	0
Mike Marino followed by Philip Condit	0
James Bell followed by Philip Condit	0
James Albaugh followed by Philip Condit	0

Table 6.10: Succession ordering results for Boeing. Each ordering is shown along with the weighted confidence measure associated with that ordering.

Company	Proposed Order (A then B)	Weighted Confidence Estimate
Boeing	<b>Philip-Condit, Harry-Stonecipher</b>	6.74
Heinz	<b>Anthony-O'Reilly, William-Johnson</b>	3.67
Staples	<b>Thomas-Stemberg, Ronald-Sargent</b>	1.61
Textron	Lewis Campbell, James Hardymon	1.43

Table 6.11: Estimates of manager order, correct order in **bold**. All are correct except for Textron, where James Hardymon precedes Lewis Campbell, not the reverse.

### 6.6.3 Span Estimation

This section discusses methods for finding the tenure of a particular manager, defined as the manager's years in office. This information is then used to give additional support or to correct the ordering proposed in the prior section.

The method for span estimation again follows the framework proposed in Section 2.2. A corpus is downloaded with a CEO's name as the hook. Using the training information, for each CEO a collection of sentences which mention the CEO is extracted and each mention of a year that is within the CEO's tenure is marked. A positive and negative example model is use for annotation. In extraction, weighted-confidence estimates are computed for each candidate year.

The first product from this extraction is a probability density over extracted years calculated for each candidate, which gives a visual aid for estimating a manager's tenure. The density is computed by taking all valid years returned by the extraction stage and normalizing. Figure 6.6 gives an illustration of the densities estimated for two Heinz CEOs, Anthony O'Reilly and William Johnson. While the densities are noisy, it can be observed by inspection that the significant peaks for O'Reilly come before the peaks for Johnson. This ordering of peaks reflects the true ordering where O'Reilly precedes Johnson.

The tenure midpoint, estimated by the mean of the densities can corroborate the ordering. The mean for the O'Reilly density is 1983, while for Johnson it is 1999. The estimated midpoints for each of the CEO pairs are shown in Table 6.12. While the estimated tenure midpoint is often quite far off from the true tenure midpoint, the relative order of the estimates can provide an additional source of evidence for the relative ordering of CEO

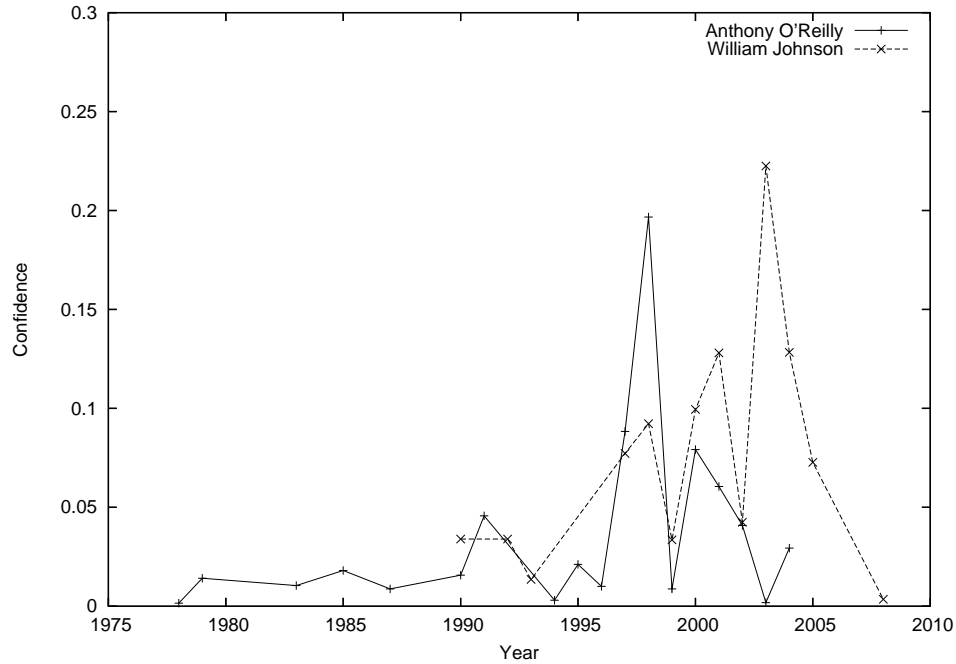


Figure 6.6: Tenure Span Estimates: Normalized confidence estimated for CEO to be in office for that year, with outliers removed.

tenures.

#### 6.6.4 Start/End Year Estimation

Thus far in the cascade, a reliable ordering for some of the CEOs for a particular company has been found. This section looks at a direct extraction of the start and end dates for a given CEO's tenure to exactly define the span of the CEO. These methods can be trained just as the other methods were, following again the framework in Section 2.2.

Table 6.13 shows the rank and score for the start year for all of the individuals and Table 6.14 shows the rank and score for individuals who retired prior to publication of this dissertation. As can be seen, the precision of extraction of start year and end years is poor. The top 10 proposed start dates for Anthony O'Reilly can be found in Table 6.15.

Company	CEO	Tenure	Estimated Tenure Midpoint
Boeing	Philip Condit	1996-2003	1993
Boeing	Harry Stonecipher	2003-2005	2000
Heinz	Anthony O'Reilly	1979-1998	1983
Heinz	William Johnson	1998-	1999
Staples	Thomas Stemberg	1985-2002	1995
Staples	Ronald Sargent	2002-	2001
Textron	James Hardymon	1992-1998	1987
Textron	Lewis Campbell	1998-	2002

Table 6.12: Estimated Tenure Midpoints from weighted sum of densities. The tenure midpoints can be used to provide more reliable relative ordering.

Name	Company	Rank of Correct Answer	Answer	Confidence
Philip Condit	Boeing	1	1996	2.26
James Hardymon	Textron	1	1992	0.75
Harry Stonecipher	Boeing	3	2003	0.74
Anthony O'Reilly	Heinz	5	1979	0.46
Lewis Campbell	Textron	1	1998	0.37
Thomas Stemberg	Staples	4	1985	0.17
Ronald Sargent	Staples	4	2002	0.08
William Johnson	Heinz	3	1998	0.07

Table 6.13: Start Year Estimation. The correct answers are ranked by the confidence of the system in that answer. For all of the candidates, the correct year is in the top 5.

Name	Company	Rank of Correct Answer	Answer	Confidence
Anthony O'Reilly	Heinz	1	1998	0.94
James Hardymon	Textron	2	1998	0.40
Philip Condit	Boeing	2	2003	0.38
Thomas Stemberg	Staples	3	2002	0.12

Table 6.14: End Year Estimation (for CEOs who have actually retired). The correct year is in the top 3 for all of the companies.

Year	Confidence
1973	0.74
2001	0.58
1998	0.56
1969	0.51
<b>1979</b>	0.46
1936	0.36
1995	0.29
1987	0.16
1997	0.12
1990	0.08

Table 6.15: Top 10 system candidates for Anthony O'Reilly's start year. 1973 is the year he became president of the company (not CEO), and 1979 is the year he became CEO. In 2001 he was knighted, in 1998 he retired, and in 1969 he joined the company. 1936 is his birthday.

### 6.6.5 Using Reasoning to Sharpen Year Estimates

Given the start and end years proposed in the previous section, and the immediate ordering information, it is possible to arrive at transition year estimates using the knowledge that someone's end year is the same as his successor's start year. The confidence estimate for a given transition year  $C_{AB}(X)$  from person A to B is estimated from a linear combination of the weighted confidence score assigned of the predecessor's end date,  $P_A^E(X)$  and the successor's start date,  $P_B^S(X)$ , under the models described in (Section 6.6.4):

$$C_{AB}(X) = P_A^E(X) \times P_B^S(X)$$

Table 6.16 shows the performance of using a linear interpolation of start years and end years for transition year. The estimated transition year provides a slightly more reliable estimate than either of the extractors individually, with a performance of 2/4 correct as opposed to 1/4 correct for either of the separate taggers.

Company	Transition X to Y	Est. End X	Est. Start Y	Est. Transition
Boeing	Philip Condit to Harry Stonecipher	1996	2005	<b>2003</b>
Heinz	Anthony O'Reilly to William Johnson	<b>1998</b>	2003	<b>1998</b>
Staples	Thomas Stemberg to Ronald Sargent	2000	2001	2001
Textron	James Hardymon to Lewis Campbell	1992	<b>1998</b>	1992

Table 6.16: A linear interpolation of the end date of one CEO and the start of his successor can lead to improved performance at estimating the transition year. Column 3 is the estimated end year of the preceding CEO, Column 4 is the estimated start year of the succeeding CEO. Column 5 is the estimated year of transition. In each row, all of the values should be the same (succession(x,y)  $\rightarrow$  end(x,D), start(y,D), transition(x,y,D)). Years in **bold** are correct.

Company	CEO	Est. Start Rank	Est. Transition	Re-Est. Start Rank
Boeing	Philip Condit	1	<b>2003</b>	1
Heinz	Anthony O'Reilly	5	<b>1998</b>	3
Staples	Thomas Stemberg	4	2001	3
Textron	James Hardymon	1	1992	-

Table 6.17: Using transition year estimates to re-estimate the start year of the preceding CEO. Columns 3 and 5 give the rank of the correct answer in the ranked start year list before and after using the transition year information. A “-” indicates that the correct year was not found in the ranked list.

Once the transition year has been estimated, it can then be used to filter the performance on start year estimation for the preceding CEOs. Table 6.17 shows the results of using this method. There is a slight improvement in the cases where the transition year was estimated correctly. However, when the transition year was not estimated correctly, there was a dramatic decrease in performance.

### 6.6.6 Timeline Evaluation and Visualization

The previous sections have proposed a method for finding the management succession history for a particular company using cascaded fact extraction and fusion, and an important issue is the quality of the extracted histories. While a system with complete

knowledge would return the entire succession history, with the CEO history from the start of the company, for reasons discussed above (Section 6.2), data on CEOs from all but the past few years are unavailable in the collected data set, and thus evaluation on those people will not be attempted. Instead, the system task will be evaluated on the CEO management history starting from 1990. For CEOs who served during the 1990s, the system should be evaluated on precision and recall of the extracted people, and the accuracy of the recovered dates. As this information is built up incrementally, in cases where incorrect information is accumulated at initial steps, this information then has a contaminating effect on information recovered further down the pipeline. An alternative method of evaluation is to grade the top-1 choice for each fact extractor/fuser in cases when the extractor was starting from the correct information. This isolates the performance of each distinct system, and may provide insight as to where exactly the system is failing.

As an alternative output, a visualization of the CEO succession information is present, displaying confidence estimates over the tenure for each manager in addition to other relevant information. Figure 6.7 gives the extracted confidence graph for the development set example of the Heinz company for all CEOs since 1990 (Anthony O'Reilly and William Johnson). A normalized confidence measure is displayed for recent years, with outlying years removed. In addition, the graph displays the estimated tenure midpoints for each of the CEOs, which are used to estimate succession ordering. The transition year estimates are then calculated using all the succession information, and are noted on the graph as well.

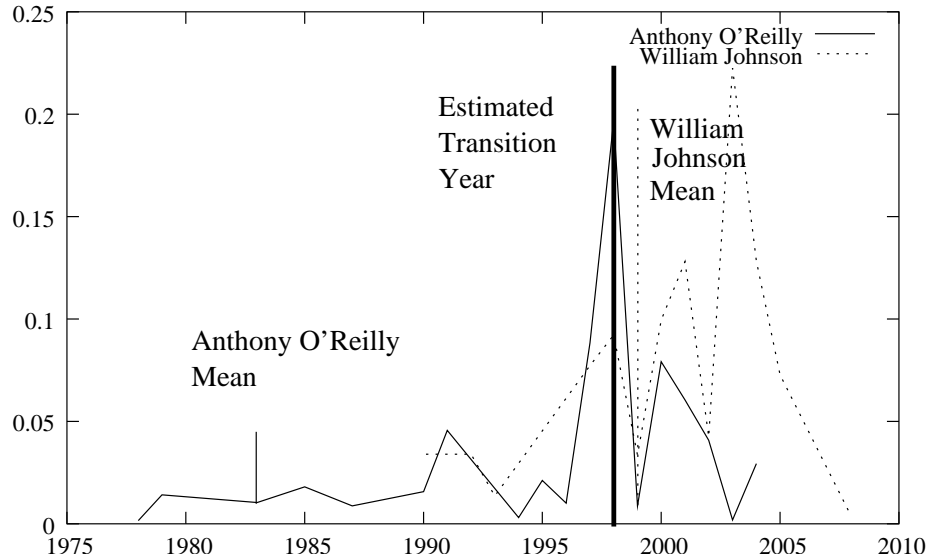


Figure 6.7: Tenure Span Estimates: Normalized confidence estimated for CEO to be in office for that year, with outliers removed. Marked years indicate estimates of tenure midpoint for each person, and an estimate of the date of transition between the two people.

## 6.7 Experimental Results

Prior reported results have been from the development set. The results presented in this section are on a set of randomly selected held-out companies: General Electric, General Motors, Gannett, The Home Depot, IBM, Kroger, Sears and UPS.

### 6.7.1 CEO Extraction

The CEO extraction results are very promising, with higher performance than was achieved in the development set. For all of the companies, the top-ranked extracted CEO is always a CEO, and for 93% of the cases, the first two top ranked extracted CEOs are in fact CEOs of the company. The absolute recall<sup>5</sup> is 68%, with the recall errors coming predominantly from CEOs before 1980.

<sup>5</sup>This was computed using the entire candidate list for each company, which ranged from 70 to 624.

CEO	Rank
GE	
John F. Welch Jr.	1
Jeffrey R. Immelt	2
Ralph Cordiner	55
Reg Jones	248
Fred J. Borch	-
GM	
Richard Wagoner	1
Roger B. Smith	2
John F. Smith	5
Alfred Pritchard Sloan Jr.	6
Robert Stempel	13
James M. Roche	15
Charles Wilson	23
Thomas Murphy	27
Richard Gerstenberg	95
Harlow H. Curtice	-
Frederic G. Donner	-
Gannett	
Douglas H. McCorkindale	1
Craig A. Dubow	2
Allen H. Neuharth	3
John J. Curley	5
Frank Gannett	6
Paul Miller	11
The Home Depot	
Robert L. Nardelli	1
Arthur M. Blank	3

CEO	Rank
IBM	
Samuel J. Palmisano	1
Louis V. Gerstner Jr.	2
Thomas J. Watson Sr.	3
John F. Akers	4
Thomas J. Watson Jr.	20
John Opel	23
Frank Cary	49
Vincent Learson	-
Kroger	
Joseph A. Pichler	1
David B. Dillon	2
James P. Herring	23
Barney Kroger	-
Albert Morrill	-
Charles M. Robertson	-
Joseph B. Hall	-
Jacob E. Davis	-
Lyle Everingham	-
Sears	
Alan J. Lacy	1
Arthur C. Martinez	2
Arthur Wood	7
Charles Kellstadt	12
Edward R. Telling	37
Theodore Houser	-
Fowler McConnell	-
Austin Cushman	-
Gordon Metcalf	-
Edward Brennen	-
UPS	
Michael L. Eskew	1
James P. Kelly	2
Kent Nelson	3
James E. Casey	-
George D. Smith	-
Harold Oberkotter	-
George Lamb	-
John Rogers	-

Table 6.18: Rank of Correctly Found CEOs for the test set. Typically, the most recent CEOs are correctly found, while some of the older CEOs are missed.

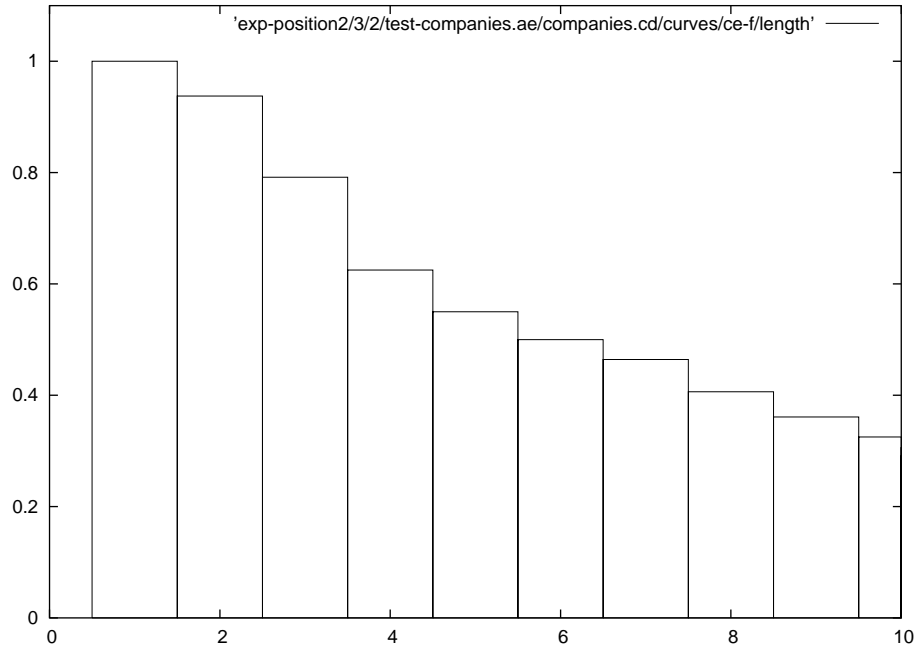


Figure 6.8: Precision at different points in the ranked list. CEOs just ranked first are 100% correct, CEOs ranked first or second are 93% correct.

### 6.7.2 Succession Information

From the extracted CEO candidate lists from the prior section, the top-1 proposed candidate for each company was taken as a known starting point, and then using the succession models proposed in Section 6.6.2 an adjacent CEO was picked from the list, and an ordering was chosen by the system. Table 6.19 shows the results of succession extraction and ordering. In 7/8 cases (87%), the system picked out the correct adjacent CEO, and out of those got 6/7 (85%) correct orderings. While the orderings will be corrected by the next stage, the incorrectly found adjacent CEO will be propagated through the rest of the pipeline.

The direct succession model yields a list of direct succession orderings and proposes candidates both before and after a given CEO. Once the highest ranked proposed candidate

Company	Proposed Order (A then B)	Confidence
IBM	<b>Louis Gerstner, Samuel Palmisano</b>	24.1
Sears	<b>Alan Lacy, Aylwin Lewis</b>	22.6
GM	Robert Lutz, Richard Wagoner	6.6
Gannett	<b>Douglas McCorkindale, Craig Dubow</b>	5.2
Kroger	<b>Joseph Pichler, David Dillon</b>	4.3
UPS	<i>Michael Eskew, James Kelly</i>	2.4
Home Depot	<b>Arthur Blank, Robert Nardelli</b>	2.0
GE	<b>John Welch, Jeffrey Immelt</b>	1.9

Table 6.19: Successor Extraction. For 7/8 companies, an adjacent CEO to the top-1 was picked out correctly. For 6/8 companies, the succession ordering was also correctly chosen (in **bold**). For 1 company, an adjacent CEO was picked, but the order wasn't correct (in *italics*).

Company	Proposed Order (A then B)	Confidence
Sears	Edward Lampert, Alan-Lacy	13.4
Gannett	<b>John Curley, Douglas McCorkindale</b>	3.8
GM	Richard Wagoner, John Smith	1.4

Table 6.20: Additional Successor Extraction. Out of 3 cases, where an adjacent CEO is proposed after the top-1 choice, only one is correct.

is chosen, the model can be used to pick out another neighboring candidate. For example, for Gannett the selected CEO was Dubow, and top direct succession candidate was that McCorkindale follows Dubow. Given that McCorkindale follows Dubow, the top candidate proposed to precede Dubow was Curley. Table 6.20 shows the additional succession candidates.

### 6.7.3 Tenure Midpoint Estimates For Relative Order Re-Estimation

Using the methods proposed in Section 6.6.3, the system estimated the tenure midpoint for each of the officials found in earlier stages, and then used these estimated tenure midpoints to reorder the individuals.

On the development set, the tenure midpoint estimate gave increased accuracy to

Company	CEO	Tenure	Estimated Tenure Midpoint
GE	John Welch	1981 - 2001	1991
GE	Jeffrey Immelt	2001 -	2001
Gannett	John Curley	1986 - 2000	1993
Gannett	Douglas McCorkindale	2000 - 2005	1998
Gannett	Craig Dubow	2005 -	2001
Home Depot	Arthur Blank	1997 - 2000	1992
Home Depot	Robert Nardelli	2000 -	2001
IBM	Louis Gerstner	1993 - 2002	1995
IBM	Samuel Palmisano	2002 -	2001
Kroger	Joseph Pichler	1990 - 2003	2001
Kroger	David Dillon	2003 -	1991
Sears	Alan Lacy	2000 - 2005	2000
Sears	Aylwin Lewis	2005 -	2003
UPS	James Kelly	1997 - 2002	1994
UPS	Michael Eskew	2002 -	1994

Table 6.21: Estimated Tenure Midpoints from weighted sum of densities. The table shows the tenure for the CEOs. All relative tenures found are correct except for Kroger, where Dillon’s tenure midpoint estimate comes before Pichler’s, and in the case of UPS where the difference between the dates is negligible.

the orderings over the direct succession modeling, but on the testing set direct succession modeling performed slightly better. For 5/7 pairs, the tenure midpoint system got the ordering correct, given one earlier mistake from incorrect CEO extraction, as opposed to 6/7 pairs for the succession modeling. For one pair, the system got very similar midpoints for the two people, and for one additional pair, the system returned the wrong answer. On the pairs which the tenure midpoint and the successor returned the same answer, the precision was 5/5, with recall of 5/7 (70%). A baseline method of taking the means of co-sentential years (Section 6.6.1), gives correct orderings to 6/7 of the pairs, which suggests that the duration model doesn’t out-perform the baseline.

Company	Transition X , Y (X,Y)	Est. End X	Est. Start	Combined	Real
GE	John Welch to Jeffrey Immelt	<b>2001</b>	2000	2000	2001
Gannett	Douglas McCorkindale to Craig Dubow	2003	<b>2005</b>	2000	2005
Home Depot	Arthur Blank to Robert Nardello	2002	<b>2000</b>	<b>2000</b>	2000
IBM	Louis Gerstner to Samuel Palmisano	1993	2000	2000	2002
Sears	Alan Lacy to Aylwin Lewis	2000	<b>2005</b>	<b>2005</b>	2005

Table 6.22: A linear interpolation of the end date of one CEO and the start of his successor. Column 3 is the estimated end year of the preceding CEO, Column 4 is the estimated start year of the succeeding CEO. Column 5 is the estimated year of transition. In each row, all of the values should be the same (succession(x,y)  $\rightarrow$  end(x,D), start(y,D), transition(x,y,D)). Years in **bold** are correct.

#### 6.7.4 Transition Estimation

Using a method of linearly combining the proposed start and end dates proposed in Section 6.6.5, proposed transition times are given in Table 6.22. Unlike with the development set, the combined values aren't better than the estimated start or end dates individually. In all, 2/5 (40%) of the transition dates are extracted correctly, so for 2/5 companies a pair with correct ordering and with the correct estimated transition date was found.

#### 6.7.5 Timeline Construction

Figures 6.10 through 6.12 show some of the partial timelines constructed by means of the cascade of extraction and fusion. The figures display estimated tenures for each of the CEOs, and the estimated tenure midpoints. Transition years are marked for all cases, except those in which the relative ordering is incorrectly identified.

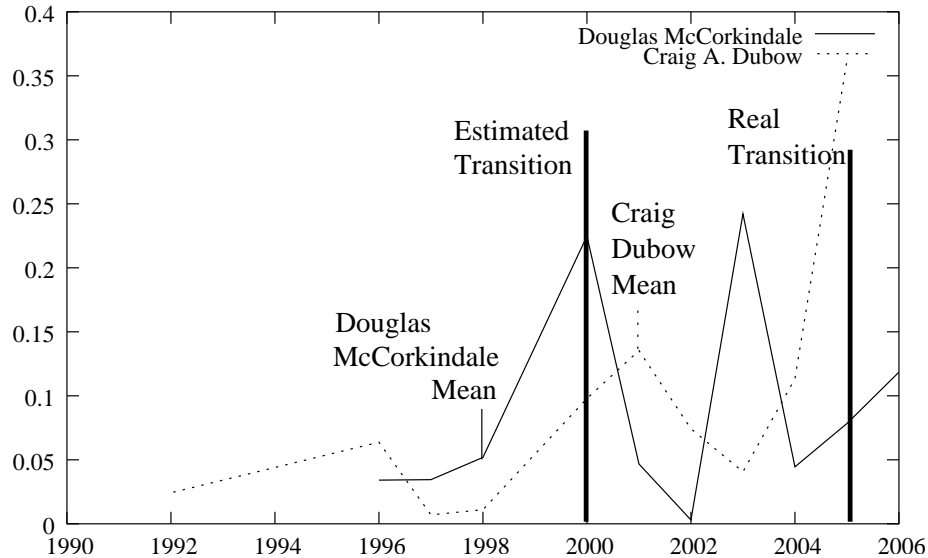


Figure 6.9: Graph of duration of CEO tenures for Gannett. As the CEO manager list has many candidates, the past three candidates have been correctly picked out and identified (John Curley, Douglas McCorkindale, Craig A. Dubow), though only the transition between McCorkindale and Dubow is estimated, and incorrectly at that. Outlying years are removed.

Overall, even in cases where the relative ordering by tenure midpoints is incorrectly identified (Figures 6.9 and 6.12), the distributions used for the midpoint calculations provide useful information. In the case of Gannett, the past three CEOs were identified and correctly ordered.

## 6.8 Related Work

Management succession has been studied in the context of the MUC-6 evaluation (1995), which included, among other tasks, an evaluation of extraction of management succession events from a single document. Two notable machine learning MUC systems were [Soderland, 1999], which learns a set of regular expressions, and [Chieu and Ng, 2002], which uses a log-linear classifier, both of which look at a non-temporal succession event

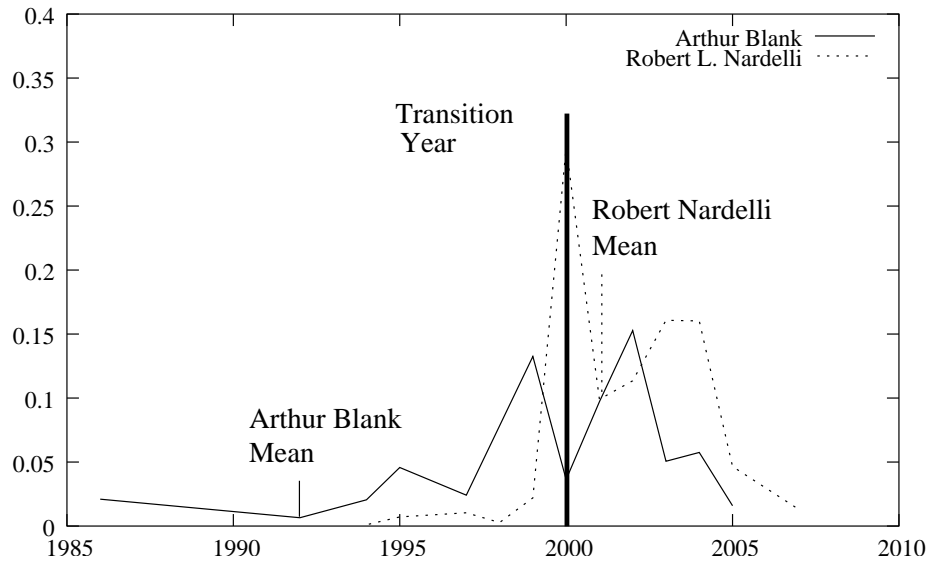


Figure 6.10: Graph of duration of CEO tenures for Home-Depot. The relative ordering by tenure midpoint estimation is correct (Arthur Blank is followed by Robert Nardelli), and the transition year is correct. Outlying years are removed.

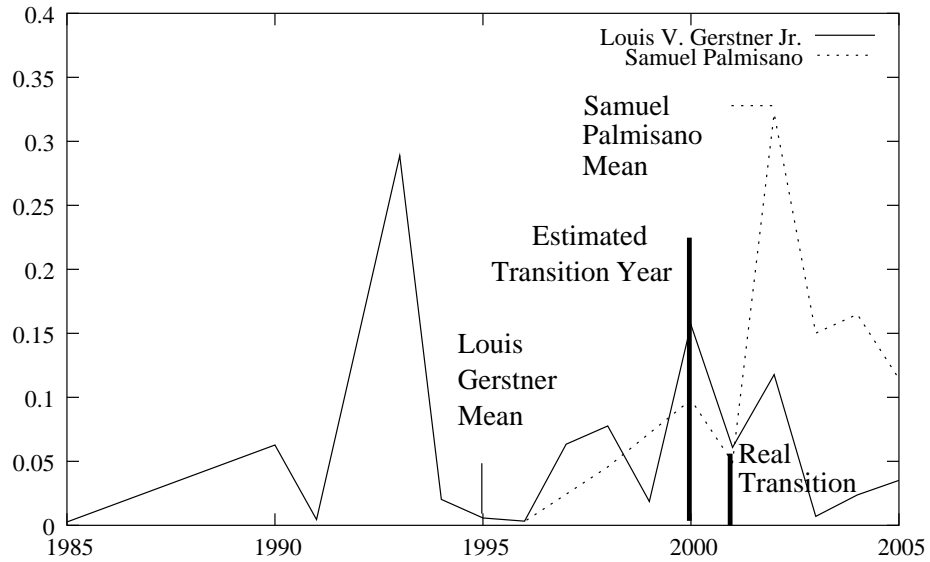


Figure 6.11: Graph of duration of CEO tenures for IBM. The relative ordering by tenure midpoint estimation is correct (Louis Gerstner followed by Samuel Palmisano), but the transition year is incorrect. Outlying years are removed.

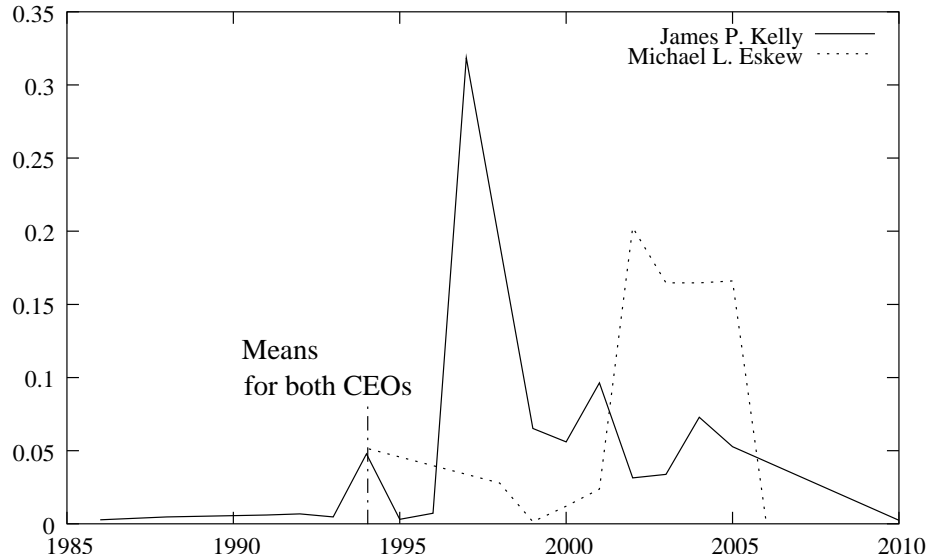


Figure 6.12: Graph of duration of CEO tenures for UPS. The relative ordering by tenure midpoint estimation is inconclusive, and the midpoint estimates are similar due to skews in the outlying years, though the graphs suggest the correct ordering.

and find the company, the position, the previous position holder, and the successor. For that task, [Chieu and Ng, 2002] reports results of 60% F-measure for the multi-slot management succession extraction from a single document. While it is difficult to provide a direct comparison to the work presented here, those results suggest higher performance on extraction. The improved results can be attributed to the presence of labeled training data (6915 annotated instances), a more homogeneous news text corpus, matched training and test data, and possibly better feature selection (e.g. parse features).

For the problem of temporal information, the main focus of research has been in various forms of summarization [Allan et al., 2001, Chieu and Lee, 2004] or in time extraction and resolution [Mani and Wilson, 2000]. There is no other work on the general topic of time-bounded fact extraction, or in timeline construction for a set of relationship facts.

## 6.9 Conclusion

This chapter has demonstrated that a cascade of fact extraction and fusion methods trained by example can be used to build up structured information, by combining information from many documents. The pipeline is described in Section 6.4 and lays out the steps (manager extraction, succession finding, tenure duration midpoints, and information combination) which build up the entire timeline. This process of knowledge refinement and acquisition is made possible by the training of multiple extractors by example using the framework proposed in Section 2.2.

## Chapter 7

# Conclusion

This dissertation has demonstrated the benefits of using statistical fact extractors trained by examples for two information analysis tasks, question answering and cross-document coreference. Methods for fusion of extracted facts were explored on two different tasks, biographic fact extraction and CEO management succession timelines, and the method of cascaded information retrieval, extraction, and fusion was successfully applied to each of these tasks. The next sections detail the methodological advances, new problem domains explored, and the tools developed.

## 7.1 Contributions

### Methodological Advances

#### Application of Relationship Extraction for Information Retrieval

Prior to this work, fact extraction had been used primarily as an end in itself, with the sole exception being as input to summarization. In this thesis, in two domains (question answering and cross-document coreference/personal name disambiguation), facts were used as additional sources of information, beyond the word level, and led to increased performance. For both of those problem domains, other researchers have adopted the techniques described here, and then replicated and validated the results ([Niu et al., 2004] and [Fleischman and Hovy, 2003]). The work on biographic fact extraction and management succession bolsters this argument by demonstrating that facts extracted with high precision can be synthesized to yield increased performance on a larger task.

For ad-hoc information retrieval, bag-of-words methods have been predominant. The successful application of extracted facts on information retrieval related tasks is significant because it supports the argument that for applications of interest, information extraction can also serve as a useful building block upon which more complicated models of language can be built.

#### Methods for Training Fact Extraction by Example

There has been a great deal of research effort in training fact extractors from annotated data, with considerably less effort invested in training them from examples. This dissertation explored the training of sequence models (CRFs) for fact extraction, and

demonstrated that along with positive examples, negative examples can be used to build better extractors. The speed-up due to this form of training allows the application of fact extractors to domains where manually annotating training data would be prohibitive.

### **Multi-Document Information Fusion Methods**

Multi-document information fusion has had relatively little attention in the research community. This dissertation compares the performance of a number of different methods for performing cross-document information fusion, including best score, Viterbi frequency and weighted confidence estimation. On a variety of tasks, weighted confidence fusion was demonstrated to be the most effective method. Crucially, given corpora with redundant information, fusion improves the precision of the extracted facts enough for them to become useful building blocks.

### **Cascaded Information Retrieval, Extraction and Fusion**

The use of extracted facts to help improve performance on related facts is another interesting result on the general theme of extracted factual information as a building block. Two sets of experiments demonstrated the use of cascaded information retrieval, extraction and fusion. For biographic facts, the system first found one fact (such as a person's birthday) through multi-document extraction and fusion, annotated this information in text, and finally extracted a separate biographic fact (such as birthyear) from this newly annotated text. The high precision of the extracted facts post-fusion made this approach possible, and the high textual correlation between facts in the domain of biographic facts allow this to lead to improved performance. For corporate management succession, one set of

extractors create a set of candidate managers, and later extractors filtered the list, ordered the candidates, and then provided estimates of the tenure of each manager to build a timeline. The extraction of an entire semantic network in a timeline, as opposed to isolated facts, is a novel use of information extraction.

## **New Problem Domains Explored**

### **Multi-Document Information Extraction**

The problem space of multi-document fact extraction has been little explored previously. Biographic fact extraction and CEO succession are two places in this dissertation in which multi-document information extraction has been successfully applied.

The key insight to be gained is that facts are frequently repeated across documents, and by taking advantage of redundancies, much higher precision performance can be achieved than would be possible from any single source.

### **Cross-Document Coreference on Web Pages**

Prior to this dissertation, the predominant published evaluations of cross-document coreference were done in the context of news corpora, and primarily focused on the “John Smith” corpus.

This dissertation presents results of cross-document coreference on the Web for both pseudonyms and a statistically-motivated sample of real polyreferent names. Based on those results, this dissertation suggests caution in the use of pseudonyms, since it demonstrates they may have a tendency to provide atypical example sets. Furthermore, experiments provided evidence that TF-IDF may not be the best weighted measure for Web

pages. The dissertation also presents state-of-the-art results of cross-document coreference on the “John Smith” Corpus using agglomerative-hierarchical centroid clustering.

### **Time-Bounded Fact Extraction and Timeline Construction**

The problem of management succession has been examined in the single-document case, but the problem of automatic management succession, span finding, and time-bounded fact extraction has been almost unexplored before this dissertation.

### **Resources and Tools Developed**

#### **Celebrity Facts**

While some of the facts used as reference came from one online source (info-please.com), a set of familial relationships and education backgrounds were manually determined for a set of 21 people. This set of facts took several days to develop. The celebrities in this set are listed in Appendix A.2.

#### **Referent-Disambiguated Web Pages**

A set of Web pages disambiguated with respect to particular referents were created. The set consists of 32 distinct people, 1,658 pages and almost 5 million words. It took 2-3 weeks to create. The personal names used are listed in Appendix B.

#### **CEO Succession Data Sources**

In order to run the management succession experiments, it was necessary to create a training and test set of corporate succession histories. Eighteen companies were selected,

and their complete history of chief executives were found. In total, there were 136 chief executives with starting dates that were tabulated. This information took several days to compile. The data is displayed in full in Appendix C.

## 7.2 Potential Applications

The scope of fact extraction, as defined in Section 1.1, is broad and can be applied to a surprisingly large number of information needs commonly underserved or poorly served by Web search engines. The methods explored in this dissertation are applicable to vast portions of relevant information, including factoid queries, such as the largest mountain in the world, or the president of Bolivia.

Cascaded fact extraction and fusion allows for the induction of large bodies of information like semantic networks. This dissertation examined time-bounded facts in the context of organizational hierarchy and membership information, which can be useful for a variety of tasks. The methods introduced extend also to corporate relationships, and could track, for example, the sequence of corporate acquisitions or membership on multiple corporate boards.

The methods examined in the context of management succession are easily extended to other domains. These methods would be applicable to spatial relationships between locations, and would allow one to build a map of the world from online sources. Causal ordering might also be handled in this framework. In the field of bio-informatics, cascaded fact extraction and fusion could be applied to build models of relationships larger than protein-to-protein interaction, and perhaps build the entire chain of gene to protein

to substance to macro-level effect through pieces of evidence gathered and fused.

Currently, the information extraction and fusion system requires a moderate amount of supervision, in particular a set of examples. Techniques which lower the requirements on supervision would enable large-scale deployment of these methods to arbitrary relationships, and might allow non-experts to create new fact extraction systems.

A slightly more speculative vision would be the use of extracted facts to allow a system to learn larger general ontological facts about the world. For example, by generalizing over a person's birth years and the year in which they become CEO of a particular company, a model of typical CEO age could be built, which could then be used to extract more subtle information. In this fashion, the bottleneck of knowledge acquisition may be surmounted by a gradual aggregation of a massive collection of facts about the world, extracted from natural language text, which allow for machine learning of semantic structures of increasing complexity and ultimately lead to language understanding.

# Bibliography

- [Agichtein, 2005] Agichtein, E. (2005). *Extracting Relations from Large Text Collections*. PhD thesis, Columbia University.
- [Agichtein and Gravano, 2000] Agichtein, E. and Gravano, L. (2000). Snowball: Extracting relations from large plain-text collections. In *Proceedings of the International Conference on Digital Libraries*, pages 85–94.
- [Allan et al., 2001] Allan, J., Gupta, R., and Khandelwal, V. (2001). Temporal summaries of news topics. In *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 10–18.
- [Appelt et al., 1993] Appelt, D., Hobbs, J., Bear, J., Israel, D., and Tyson, M. (1993). FASTUS: a finite-state processor for information extraction from real-world text. In *Proceedings of the International Joint Conference on Artificial Intelligence*.
- [Bagga, 1998] Bagga, A. (1998). Evaluation of coreferences and coreference resolution systems. In *Proceedings of the Language Resources and Evaluation Conference*, pages 563–566.
- [Bagga and Baldwin, 1998] Bagga, A. and Baldwin, B. (1998). Entity-based cross-

- document coreferencing using the vector space model. In *Proceedings of the Meeting of the Association for Computational Linguistics*, pages 79–85.
- [Barzilay et al., 1999] Barzilay, R., McKeown, K. R., and Elhadad, M. (1999). Information fusion in the context of multi-document summarization. In *Proceedings of the Meeting of the Association for Computational Linguistics*, pages 550–557.
- [Bekkerman and McCallum, 2005] Bekkerman, R. and McCallum, A. (2005). Disambiguating web appearances of people in a social network. In *Proceedings of the World Wide Web Conference*.
- [Berland and Charniak, 1999] Berland, M. and Charniak, E. (1999). Finding parts in very large corpora. In *Proceedings of the Meeting of the Association for Computational Linguistics*, pages 57–64.
- [Bikel et al., 1999] Bikel, D. M., Schwartz, R. L., and Weischedel, R. M. (1999). An algorithm that learns what’s in a name. *Machine Learning*, 34(1-3):211–231.
- [Brill, 1992] Brill, E. (1992). A Simple Rule-based Part of Speech Tagger. In *Proceedings of the Conference on Applied Natural Language Processing*.
- [Brill et al., 2001] Brill, E., Lin, J., Banko, M., Dumais, S., and Ng, A. (2001). Data-intensive question answering. In *Proceedings of the Text Retrieval Conference*, pages 183–189.
- [Brill and Resnik, 1994] Brill, E. and Resnik, P. (1994). A rule-based approach to pp attachment. In *Proceedings of the Meeting of the International Committee on Computational Linguistics*.

- [Brin, 1998] Brin, S. (1998). Extracting patterns and relations from the world wide web. In *WebDB Workshop at 6th International Conference on Extending Database Technology, EDBT'98*, pages 172–183.
- [Brown et al., 1992] Brown, P. F., Pietra, V. J. D., DeSouza, P. V., Lai, J. C., and Mercer, R. L. (1992). Class-based n-gram models of natural language. *Computational Linguistics*, pages 467–479.
- [Bunescu and Mooney, 2004] Bunescu, R. and Mooney, R. (2004). Collective information extraction with relational markov networks. In *Proceedings of the Meeting of the Association for Computational Linguistics*, pages 438–445.
- [Califf and Mooney, 1998] Califf, M. E. and Mooney, R. J. (1998). Relational learning of pattern-match rules for information extraction. In *Working Notes of AAAI Spring Symposium on Applying Machine Learning to Discourse Processing*, pages 6–11, Menlo Park, CA. AAAI Press.
- [Caraballo, 1999] Caraballo, S. (1999). Automatic acquisition of a hypernym-labeled noun hierarchy from text. In *Proceedings of the Meeting of the Association for Computational Linguistics*.
- [Caraballo and Charniak, 1999] Caraballo, S. and Charniak, E. (1999). Determining the specificity of nouns from text. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing*.
- [Cardie and Wagstaff, 1999] Cardie, C. and Wagstaff, K. (1999). Noun phrase coreference

- as clustering. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing*, pages 82–89.
- [Census, 1990] Census (1990). Published by the US Census Bureau, available at: <http://www.census.gov/main/www/cen1990.html>.
- [Chen and Bian, 1998] Chen, H.-H. and Bian, G.-W. (1998). White pages construction from web pages for finding people on the internet. *Computational Linguistics and Chinese Language Processing*, 3(1):75–100.
- [Chieu and Lee, 2004] Chieu, H. L. and Lee, Y. K. (2004). Query based event extraction along a timeline. In *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 425–432.
- [Chieu and Ng, 2002] Chieu, H. L. and Ng, H. T. (2002). A maximum entropy approach to information extraction from semi-structured and free text. In *Proceedings of the National Conference on Artificial Intelligence*, pages 786–791.
- [Ciravenga, 2001] Ciravenga, F. (2001). Adaptive information extraction from text by rule induction and generalization. In *Proceedings of the International Joint Conference on Artificial Intelligence*.
- [Clarke et al., 2001] Clarke, C. L. A., Cormack, G. V., and Lynam, T. R. (2001). Exploiting redundancy in question answering. In *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 358–365.
- [Collins, 1999] Collins, M. (1999). *Head-Driven Statistical Models for Natural Language Parsing*. PhD thesis, University of Pennsylvania.

- [Collins and Singer, 1999] Collins, M. and Singer, Y. (1999). Unsupervised models for named entity classification. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing*.
- [Copestake, 1990] Copestake, A. (1990). An approach to building the hierarchical element of a lexical knowledge base from a machine readable dictionary. In *First International Workshop on Inheritance in NLP*.
- [Cowie et al., 2000] Cowie, J., Nirenburg, S., and Molina-Salgado, H. (2000). Generating personal profiles. In *The International Conference On MT And Multilingual NLP*.
- [Csomai, 2005] Csomai, A. (2005). Wordnet bibliography. Available at: <http://mira.csci.unt.edu/~wordnet/>.
- [Culotta and McCallum, 2004] Culotta, A. and McCallum, A. (2004). Confidence estimation for information extraction. In *Proceedings of the Conference on Human Language Technologies and North American Association for Computational Linguistics*.
- [Culotta and Sorensen, 2004] Culotta, A. and Sorensen, J. (2004). Dependency tree kernels for relation extraction. In *Proceedings of the Meeting of the Association for Computational Linguistics*.
- [Dalmas and Webber, 2004] Dalmas, T. and Webber, B. (2004). Information fusion for answering factoid questions. In *Proceedings of 2nd CoLogNET-ElsNET Symposium. Questions and Answers: Theoretical Perspectives*.
- [Downey et al., 2005] Downey, D., Etzioni, O., and Soderland, S. (2005). A probabilistic

- model of redundancy in information extraction. In *Proceedings of the International Joint Conference on Artificial Intelligence*.
- [Etzioni et al., 2004] Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D., and Yates, A. (2004). Web-scale information extraction in knowitall. In *Proceedings of the World Wide Web Conference*.
- [Etzioni et al., 2005] Etzioni, O., Cafarella, M., Downey, D., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D., and Yates, A. (2005). Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165(1):91–134.
- [Feng and Hovy, 2005] Feng, D. and Hovy, E. (2005). Handling biographical questions with implicature. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing*.
- [Fleischman and Hovy, 2003] Fleischman, M. and Hovy, E. (2003). Offline strategies for online question answering: Answering questions before they are asked. In *Proceedings of the Meeting of the Association for Computational Linguistics*.
- [Fleischman and Hovy, 2004] Fleischman, M. B. and Hovy, E. (2004). Multi-document person name resolution. In *Proceedings of ACL Reference Resolution Workshop*.
- [Florian and Ngai, 2001] Florian, R. and Ngai, G. (2001). Multidimensional transformation-based learning. In *Proceedings of the Conference on Natural Language Learning*.
- [Francis and Kucera, 1964] Francis, W. and Kucera, H. (1964). Brown corpus manual:

Manual of information to accompany a standard corpus of present day edited American English. Brown University, Providence, Rhode Island.

- [Freitag, 1998] Freitag, D. (1998). Toward General-Purpose Learning for Information Extraction. In *Proceedings of the Meeting of the Association for Computational Linguistics*.
- [Freitag and Kushmerick, 2000] Freitag, D. and Kushmerick, N. (2000). Boosted wrapper induction. In *Proceedings of the Joint National Conference on Artificial Intelligence and the Innovative Applications of Artificial Intelligence Conference*, pages 577–583.
- [Freitag and McCallum, 1999] Freitag, D. and McCallum, A. (1999). Information extraction with hmms and shrinkage. In *Proceedings of the AAAI-99 Workshop on Machine Learning for Information Extraction*, pages 31–36.
- [Gale et al., 1992] Gale, B., Church, K., and Yarowsky, D. (1992). Work on statistical methods for word sense disambiguation. In *AAAI Fall Symposium on Probabilistic Approaches to Natural Language Processing*, pages 54–60, Cambridge, MA.
- [Gaustad, 2001] Gaustad, T. (2001). Statistical corpus-based word sense disambiguation: Pseudowords vs. real ambiguous words. In *Proceedings of the Student Research Workshop at the Joint Meeting of the Association for Computation Linguistics and the European Association for Computational Linguistics*.
- [Girju, 2001] Girju, R. (2001). Answer fusion with on-line ontology development. In *Student Research Workshop Proceedings at The Meeting of the North American Chapter of the Association for Computational Linguistics*.

- [Gooi and Allan, 2004] Gooi, C. H. and Allan, J. (2004). Cross-document coreference on a large scale corpus. In *Proceedings of the Conference on Human Language Technologies and North American Association for Computational Linguistics*.
- [Harabagiu et al., 2000] Harabagiu, S., Moldovan, D., Pasca, M., Mihalcea, R., nu, M. S., Bunescu, R., Girju, R., Rus, V., and Mor, P. (2000). Falcon : Boosting knowledge for answer engines. *Proceedings of the Text Retrieval Conference*.
- [Hearst, 1992] Hearst, M. (1992). Automatic acquisition of hyponyms from large text corpora. *Proceedings of the Fourteenth International Conference on Computational Linguistics (COLING-92)*.
- [Hildebrandt et al., 2004] Hildebrandt, W., Katz, B., and Lin, J. (2004). Answering definition questions using multiple knowledge sources. In *Proceedings of the Conference on Human Language Technologies and North American Association for Computational Linguistics*.
- [Hobbs, 1986] Hobbs, J. R. (1986). Overview of the tacitus project. *Computational Linguistics*, 12(3).
- [Jones et al., 1999] Jones, R., McCallum, A., Nigam, K., and Riloff, E. (1999). Bootstrapping for Text Learning Tasks. In *IJCAI-99 Workshop on Text Mining: Foundations, Techniques, and Applications*.
- [Kupiec, 1993] Kupiec, J. (1993). Murax: A robust linguistic approach for question answering using an on-line encyclopedia. In *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*.

- [Lafferty et al., 2001] Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning*, pages 282–289.
- [Leek, 1997] Leek, T. R. (1997). Information extraction using hidden markov models. Master’s Thesis, UC San Diego.
- [Lehnert et al., 1991] Lehnert, W., Cardie, C., Fisher, D., Riloff, E., and Williams, R. (1991). University of Massachusetts: Description of the CIRCUS System as Used for MUC-3. In *Proceedings of the Message Understanding Conference*, pages 223–233, San Mateo, CA. Morgan Kaufmann.
- [Li et al., 2005] Li, X., Morie, P., and Roth, D. (2005). Semantic integration in text: From ambiguous names to identifiable entities. *AI Magazine. Special Issue on Semantic Integration*, pages 45–68.
- [Lin and Pantel, 2001] Lin, D. and Pantel, P. (2001). Induction of semantic classes from natural language text. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 317–322.
- [Lyman and Varian, 2003] Lyman, P. and Varian, H. (2003). How much information. P. Lyman and H. Varian. ”How Much Information”. Retrieved from <http://www.sims.berkeley.edu/how-much-info-2003> on 9/2/2005.
- [MacIntyre, 1995] MacIntyre, R. (1995). Penn tokenizer. <http://www.cis.upenn.edu/~treebank/tokenizer.sed>.

- [Mani and Wilson, 2000] Mani, I. and Wilson, G. (2000). Robust temporal processing of news. In *Proceedings of the Meeting of the Association for Computational Linguistics*, pages 69–76.
- [Mann, 2002] Mann, G. S. (2002). Learning how to answer questions using trivia games. In *Proceedings of the Meeting of the International Committee on Computational Linguistics*, pages 612–618.
- [Mann and Yarowsky, 2003] Mann, G. S. and Yarowsky, D. (2003). Unsupervised personal name disambiguation. In *Proceedings of the Conference on Natural Language Learning*, pages 33–40.
- [Masterson and Kushmerick, 2003] Masterson, D. and Kushmerick, N. (2003). Information extraction from multi-document threads. In *Proceedings of the European Conference on Machine Learning*, pages 34–41.
- [McCallum, 2002] McCallum, A. (2002). Mallet: A machine learning for language toolkit.
- [McDonald et al., 2005] McDonald, R., Pereira, F., Kulick, S., Winters, S., Jin, Y., and White, P. (2005). Simple algorithms for complex relationship extraction with applications to biomedical ie. In *Proceedings of the Meeting of the Association for Computational Linguistics*.
- [Miller, 1990] Miller, G. (1990). Wordnet: An On-line Lexical Database. *International Journal of Lexicography*, 3(4):235–312.
- [Miller et al., 1998] Miller, S., Crystal, M., Fox, H., Ramshaw, L., and z, R. S. (1998).

- Algorithms that learn to extract information bbn:descriptions of the sift system as used for muc-7. *Proceedings of the Message Understanding Conference*.
- [Mohri et al., 2003] Mohri, M., Pereira, F., and Riley, M. (2003). At&t fsm library.
- [N. Chincor, 1998] N. Chincor, P. Robinson, E. B. (1998). Hub-4 named entity task definition version 4.8. Available by ftp from [www.nist.gov/speech/hub4\\_98](http://www.nist.gov/speech/hub4_98).
- [Nahm and Mooney, 2002] Nahm, U. and Mooney, R. (2002). Text mining with information extraction. In *Proceedings of the AAAI 2220 Spring Symposium on Mining Answers from Texts and Knowledge Bases*, pages 60–67.
- [Niu et al., 2004] Niu, C., Li, W., and Srihari, R. K. (2004). Weakly supervised learning for cross-document person name disambiguation supported by information extraction. In *Proceedings of the Meeting of the Association for Computational Linguistics*.
- [Pantel, 2003] Pantel, P. A. (2003). *Clustering by Committee*. PhD thesis, University of Alberta.
- [Pasca and Harabagiu, 2001] Pasca, M. and Harabagiu, S. (2001). The informative role of wordnet in open-domain question answering. In *Proceedings of the NAACL 2001 Workshop on WordNet and Other Lexical Resources: Applications, Extensions and Customizations*, pages 138–143. Association for Computational Linguistics.
- [Pedersen et al., 2005] Pedersen, T., Purandare, A., and Kulkarni, A. (2005). Name discrimination by clustering similar contexts. In *Proceedings of the Conference on Intelligent Text Processing and Computational Linguistics*.

- [Pereira et al., 1993] Pereira, F. C. N., Tishby, N., and Lee, L. (1993). Distributional clustering of english words. In *Proceedings of the Meeting of the Association for Computational Linguistics*, pages 183–190.
- [Phillips and Riloff, 2002] Phillips, W. and Riloff, E. (2002). Exploiting strong syntactic heuristic and co-training to learn semantic lexicons. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing*.
- [Prager et al., 2004] Prager, J., Chu-Carroll, J., and Czuba, K. (2004). Question answering by constraint satisfaction: Qa-by-dossier with constraints. In *Proceedings of the Meeting of the Association for Computational Linguistics*, pages 574–581.
- [Pustejovsky et al., 2003] Pustejovsky, J., Castao, J., Ingria, R., Saur, R., Gaizauskas, R., Setzer, A., and Katz, G. (2003). Timeml: Robust specification of event and temporal expressions in text. In *Proceedings of the International Workshop on Computational Semantics*.
- [Radev and McKeown, 1998] Radev, D. R. and McKeown, K. R. (1998). Generating natural language summaries from multiple on-line sources. *Computational Linguistics*, 24(3):469–500.
- [Ravichandran and Hovy, 2002] Ravichandran, D. and Hovy, E. (2002). Learning surface text patterns for a question answering system. In *Proceedings of the Meeting of the Association for Computational Linguistics*, pages 41–47.
- [Ravin and Kazi, 1999] Ravin, Y. and Kazi, Z. (1999). Is hillary rodham clinton the presi-

- dent? disambiguating names across documents. In *Proceedings of the ACL '99 Workshop on Coreference and its Applications*.
- [Richardson et al., 1998] Richardson, S., Dolan, W., and Vanderwende, L. (1998). Mind-net:acquiring and structuring semantic information from text. In *Proceedings of the Joint Meeting of the International Committee on Computational Linguistics and the Association for Computational Linguistics*.
- [Riloff, 1993] Riloff, E. (1993). Automatically Constructing a Dictionary for Information Extraction Tasks. In *Proceedings of the National Conference on Artificial Intelligence*, pages 811–816. AAAI Press/The MIT Press.
- [Riloff and Jones, 1999a] Riloff, E. and Jones, R. (1999a). Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping. In *Proceedings of the National Conference on Artificial Intelligence*.
- [Riloff and Jones, 1999b] Riloff, E. and Jones, R. (1999b). Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the National Conference on Artificial Intelligence*, pages 1044–1049.
- [Roark and Charniak, 1998] Roark, B. and Charniak, E. (1998). Noun-phrase Co-occurrence Statistics for Semi-automatic Semantic Lexicon Construction. In *Proceedings of the Meeting of the Association for Computational Linguistics*, pages 1110–1116.
- [Schiffman et al., 2001] Schiffman, B., Mani, I., and Concepcion, K. J. (2001). Producing biographical summaries: Combining linguistic knowledge with corpus statistics. In *Proceedings of the Meeting of the Association for Computational Linguistics*, pages 450–457.

- [Schutze, 1992] Schutze, H. (1992). Context space. In *AAAI Fall Symposium on Probabilistic Approaches to Natural Language*.
- [Schutze, 1998] Schutze, H. (1998). Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123.
- [Scott and Matwin, 1998] Scott, S. and Matwin, S. (1998). Text classification using WordNet hypernyms. In Harabagiu, S., editor, *Use of WordNet in Natural Language Processing Systems: Proceedings of the Conference*, pages 38–44. Association for Computational Linguistics, Somerset, New Jersey.
- [Setzer, 2001] Setzer, A. (2001). *Temporal Information in Newswire Articles: An Annotation Scheme and Corpus Study*. PhD thesis, University of Sheffield.
- [Skounakis and Craven, 2003] Skounakis, M. and Craven, M. (2003). Evidence combination in biomedical natural language processing. In *SIGKDD Workshop on Data Mining in Bioinformatics*.
- [Soderland, 1999] Soderland, S. (1999). Learning Information Extraction Rules for Semi-structured and Free Text. *Machine Learning*.
- [Soderland et al., 1995] Soderland, S., Fisher, D., Aseltine, J., and Lehnert, W. (1995). CRYSTAL: Inducing a conceptual dictionary. In *Proceedings of IJCAI*, pages 1314–1319.
- [Srihari and Li, 1999] Srihari, R. and Li, W. (1999). Information extraction supported question answering. In *Proceedings of the Text Retrieval Conference*.
- [Stevenson, 2004] Stevenson, M. (2004). Information extraction from single and multiple

- sentences. In *Proceedings of the Meeting of the International Committee on Computational Linguistics*.
- [Sussna, 1993] Sussna, M. (1993). Word sense disambiguation for free-text indexing using a massive semantic network. In *Proceedings of the Conference on Information and Knowledge Management*.
- [Sutton et al., 2004] Sutton, C., Rohanimanesh, K., and McCallum, A. (2004). Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. In *Proceedings of the International Conference on Machine Learning*.
- [TREC-9 Proceedings, 2000] TREC-9 Proceedings (2000). *Proceedings of the Ninth Text Retrieval Conference*. National Institute of Standards and Technology, Special Publication 500-249, Gaithersburg, MD.
- [Voorhees, 1993] Voorhees, E. (1993). Using wordnet to disambiguate word sense for information retrieval. In *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*.
- [Wacholder et al., 1997] Wacholder, N., Ravin, Y., and Choi, M. (1997). Disambiguation of proper names in text. In *Proceedings of the Conference on Applied Natural Language Processing*, pages 202–208.
- [White et al., 2001] White, M., Korelsky, T., Cardie, C., Ng, V., Pierce, D., and Wagstaff, K. (2001). Multi-document summarization via information extraction. In *Proceedings of the Conference on Human Language Technologies*.

- [Yangarber et al., 2000] Yangarber, R., Grishman, R., Tapanainen, P., and Huttunen, S. (2000). Automatic acquisition of domain knowledge for information extraction. In *Proceedings of the Meeting of the International Committee on Computational Linguistics*.
- [Yarowsky, 1995] Yarowsky, D. (1995). Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*.
- [Zelenko et al., 2003] Zelenko, D., Aone, C., and Richardella, A. (2003). Kernel methods for relation extraction. *Journal of Machine Learning Research*, 3:1083–1106.

# Appendix A

## Celebrities

*This is the list of 152 celebrities who were used for training and testing of biographic fact extraction, and in the pseudoname disambiguation experiments. The list and the facts about the individuals were extracted from from [www.infoplease.com](http://www.infoplease.com).*

### A.1 IP152

Aaron Neville, Abigail Buren, Akira Kurosawa, Al Capp, Alan Hale, Alan Jackson, Alan Ladd, Albert Brooks, Alfred Knopf, Alicia Silverstone, Alison Krauss, Alistair Cooke, Andie MacDowell, Andy Warhol, Angela Davis, Angelina Jolie, Anita Bryant, Ann Magnuson, Annie Potts, Anthony Edwards, Anthony Perkins, Anthony Quinn, Antonio Banderas, Art Carney, Arthur Fiedler, Arthur Miller, Ashley Judd, Athol Fugard, August Wilson, Ava Gardner, Avery Brooks, Barry Levinson, Ben Harper, Benicio Toro, Benjamin Bratt, Bernadette Peters, Bernardo Bertolucci, Beverly Sills, Bill Moyers, Bill Paxton, Billy Crystal, Billy Joel, Blake Edwards, Bonnie Parker, Brad Pitt, Brent Spiner, Brian Wilson,

Bridget Fonda, Bruce Dern, Bryant Gumbel, Buddy Holly, Burt Bacharach, Burt Lancaster, Buster Keaton, Calista Flockhart, Camille Paglia, Camryn Manheim, Carlo Ponti, Carol Channing, Carole Lombard, Catherine Scorsese, Charles Bronson, Charles Fuller, Charles Grodin, Charlie Parker, Charlize Theron, Chris Columbus, Chris Farley, Christina Aguilera, Christina Ricci, Darryl Zanuck, David Byrne, David Chokachi, David Hasselhoff, David Pierce, David Letterman, David Mamet, David McCord, David Selznick, Debbie Allen, Debra Winger, Demi Moore, Denise Levertov, Denver Pyle, Derek Taylor, Diane Keaton, Diane Ladd, Diane Sawyer, Dick Cavett, Dick Dyke, Dino Laurentiis, Don McLean, Donald Sutherland, Donna Karan, Doris Day, Dwight Yoakam, Eavan Boland, Eddie Vedder, Edgar Poe, Edgar Bergen, Edward Burns, Edward Mulhare, Elisabeth Shue, Elizabeth Hurley, Ella Baker, Ellen Barkin, Ellen DeGeneres, Elliott Gould, Elvis Presley, Emilio Estevez, Emily Blackwell, Emma Thompson, Ernest Hemingway, Ethel Merman, Eva Saint, Farrah Fawcett, Faye Wattleton, Faye Wong, Florence Henderson, Frances McDormand, Frank Sinatra, Frank Zappa, Frankie Muniz, Garth Brooks, George Kennedy, Gina Lollobrigida, Glenn Ford, Gloria Estefan, Goldie Hawn, Greer Garson, Greg Kinnear, Guy Lombardo, Gwyneth Paltrow, Harold Robbins, Harry Connick, Harry Saltzman, Heather Locklear, Helen Shaver, Henny Youngman, Henry Fonda, Henry Mancini, Hilary Swank, Hugh Grant, Hume Cronyn, Ian Bannen, Irene Papas, Irving Berlin, Ismail Merchant, Itzhak Perlman, Jackie Chan, Jacqueline Bisset, James Brown

## A.2 C21

*This list of celebrities was used in relationship extraction. A few additional individuals were used in development of the various methods, but no results are reported on them.*

Barbara Walters, Billie Jean King, Bob Dylan, Donald Trump, Gloria Steinem, Hermann Hese, Jackson Pollock, Joerg Haider, Ludwig van Beethoven, Mickey Mantle, Miles Davis, Mohandas Gandhi, Niels Bohr, Philip Roth, Ronald Reagan, Ruth bader Ginsburg, Sigmund Freud, Tom Cruise, William Blake, Wolfgang Amadeus Mozart, and Woody Harrelson.

## Appendix B

# Individuals in Web Polyreference Experiments

*This is the list of the 36 individuals in the W03 set who were used for training and testing of cross-document coreference on the web. The first names and last name pairs were generated uniformly at random (assuming strict independence) from U.S. census statistics.*

Abby Watkins, Alexander Markham, Alfred Schroeder, Alice Gilbreath, Armando Valencia, Cathie Ely, Celeste Paquette, Charlotte Bergeron, Cynthia Voigt, Dan Rhone, Elmo Hardy, Gillian Symons, Gregory Brennan, Guy Crider, Guy Dunbar, Hannah Bassham, Helen Cawthorne, Ione Westover, Louis Sidoti, Luke Choi, Maile Doyle, Martin Nagel, Mary Lemanski, Miranda Bollinger, Pam Tetu, Patrick Karlsson, Roy Tamashiro, Sidney Shorter, Stacey Doughty, Tim Whisler, Todd Platts, Young Dawkins

## Appendix C

# Corporate Succession Data

*The following tables list all of the chief executives for 18 companies since the founding of the company. The companies were chosen from a list of the Fortune 500, and the rank of the company on the list is given next to the company name. The list of chief executives was compiled by manual search on the Internet. One difficulty in compiling the list is that for some companies the title given to the chief executive changed over time, often starting as president and in later years changing to chief executive officer and chairman. In the below tables, chief executives who had a title other than CEO are marked with an asterisk (\*).*

Chief Executive	Starting Year
General Motors (# 3)	
Alfred P. Sloan	1923
Charles E. Wilson	1946
Harlow H. Curtice	1953
Frederic G. Donner	1958
James M. Roche	1967
Richard C. Gerstenberg	1972
Thomas A. Murphy	1974
Roger B. Smith	1981
Robert C. Stempel	1990
John F. Smith	1992
Richard Wagoner	2000
General Electric (# 5)	
*Charles A. Coffin	1913
*E. W. Rice	1922
*Gerald Swope	1922
*Charles E. Wilson	1940
Ralph J. Cordiner	1950
Fred J. Borch	1963
Reg Jones	1972
Jack Welch	1981
Jeff Immelt	2001
IBM (# 9)	
Thomas J. Watson Sr.	1914
Thomas J. Watson Jr.	1956
Vincent Learson	1971
Frank T. Cary	1973
John R. Opel	1981
John Akers	1986
Louis V. Gerstner Jr.	1993
Sam Palmisano	2002
Hewlett-Packard (# 11)	
Dave Packard	1947
Bill Hewlett	1969
John Young	1978
Lew Platt	1992
Carleton "Carly" S. Fiorina	1999
Mark Hurd	2005

Chief Executive	Starting Year
The Home Depot (# 13)	
Arthur Blank	1997
Robert L. Nardelli	2000
Kroger (# 19)	
Barney H. Kroger	1883
Albert H. Morrill	1930
Charles M. Robertson	1942
Joseph B. Hall	1946
Jacob E. Davis	1964
James P. Herring	1970
Lyle Everingham	1978
Joseph A. Pichler	1990
David B. Dillon	2003
Boeing (# 21)	
*William Boeing	1916
*Edgar Gott	1922
*Philip G. Johnson	1925
*Clairmont Egtvedt	1933
*Philip G. Johnson	1939
*Clairmont Egtvedt	1944
*William M. Allen	1945
Thornton A. Wilson	1968
Frank A. Shrontz	1985
Philip M. Condit	1996
Harry C. Stonecipher	2003
Pfizer (# 25)	
*Emile Pfizer	1906
*George A. Anderson	1941
*John L. Smith	1945
*John E. McKeen	1949
John J. Powers Jr.	1966
Edmund T. Pratt Jr.	1972
William C. Steere Jr.	1991
Henry A. McKinnell	2000

Chief Executive	Starting Year
Sears (# 32)	
*Richard W. Sears	1886
*Julius Rosenwald	1924
*Lessing Rosenwald	1932
*Robert E. Wood	1939
Theodore Houser	1954
Fowler McConnell	1958
Charles Kellstadt	1960
Austin Cushman	1962
Gordon Metcalf	1967
Arthur Wood	1973
Edward Telling	1978
Edward Brennen	1986
Arthur Martinez	1995
Alan Lacy	2000
Aylwin Lewis	2005
UPS (# 42)	
James E. Casey	1907
George D. Smith	1962
Paul Oberkotter	1972
Harold Oberkotter	1973
George Lamb	1980
John W. Rogers	1984
Kent Nelson	1989
James P. Kelly	1997
Michael L. Eskew	2002

Chief Executive	Starting Year
Raytheon (# 107)	
*Richard Aldrich	1922
*William Gammell Jr.	1924
*Laurence Marshall	1928
*Charles F. Adams	1948
*Richard E. Krafve	1960
*Charles F. Adams	1962
Thomas L. Phillips	1964
Dennis J. Picard	1991
Daniel P. Burnham	1998
William H. Swanson	2003
Anheuser-Busch (# 142)	
*Eberhard Anheuser	1875
*Adolphus Busch	1880
*August A. Busch Sr.	1913
*Adolphus Busch II	1934
*August A. Busch Jr.	1946
August A. Busch III	1975
Patrick T. Stokes	2002

CEO	Starting Year
Staples (# 152)	
Thomas G. Stemberg	1985
Ronald L. Sargent	2002
Textron (# 194)	
Royal Little	1923
Rupert Thompson	1960
William Miller	1968
Joseph Collinson	1977
Robert P. Straetz	1979
Beverly F. Dolan	1986
James F. Hardyman II	1992
Lewis B. Campbell	1998
Heinz (# 213)	
*Henry Johns Heinz	1869
*Howard Heinz	1919
*H. J. Heinz II	1941
*Burt Gookin	1966
Anthony J. F. O'Reilly	1979
William R. Johnson	1998
Lennar (# 230)	
Leonard Miller	1954
Stuart Miller	1997
Gannett (# 278)	
Frank Gannett	1906
Paul Miller	1957
Allen H. Neuharth	1973
John J. Curley	1986
Douglas McCorkindale	2000
Craig A. Dubow	2005
McGraw-Hill (# 356)	
*John Hill	1909
*James H. McGraw	1916
*Malcolm Muir	1928
*James McGraw Jr.	1937
*Curtis W. McGraw	1950
*Donald C. McGraw	1953
*Shelton Fisher	1968
Harold McGraw Jr.	1974
Joseph Dionne	1983
Harold "Terry" W. McGraw	1998

# Vita

Gideon Sheppard Mann was born on September 5th, 1977, and grew up in Newton, Massachusetts. He graduated Brown University receiving a Sc.B. with honors in Computer Science in May 1999. While at Brown, he worked in the nascent BLIP. He participated in the 2000 CLSP Summer Workshop and was awarded a one year fellowship. Summer 2001, he worked at MITRE in the Information Technology Center. He is currently a Research Fellow at the University of Massachusetts in Amherst.