Multi-Field Information Extraction and Cross-Document Fusion

Gideon S. Mann and **David Yarowsky**
Department of Computer Science
The Johns Hopkins University
Baltimore, MD 21218 USA
{gsm,yarowsky}@cs.jhu.edu

Abstract

In this paper, we examine the task of extracting a set of biographic facts about target individuals from a collection of Web pages. We automatically annotate training text with positive and negative examples of fact extractions and train Rote, Naïve Bayes, and Conditional Random Field extraction models for fact extraction from individual Web pages. We then propose and evaluate methods for fusing the extracted information across documents to return a consensus answer. A novel cross-field bootstrapping method leverages data interdependencies to yield improved performance.

1 Introduction

Much recent statistical information extraction research has applied graphical models to extract information from one particular document after training on a large corpus of annotated data (Leek, 1997; Freitag and McCallum, 1999).[1] Such systems are widely applicable, yet there remain many information extraction tasks that are not readily amenable to these methods. Annotated data required for training statistical extraction systems is sometimes unavailable, while there are examples of the desired information. Further, the goal may be to find a few interrelated pieces of information that are stated multiple times in a set of documents.

Here, we investigate one task that meets the above criteria. Given the name of a celebrity such as "Frank Zappa", our goal is to extract a set of biographic facts (e.g., birthdate, birth place and occupation) about that person from documents on the Web.

First, we describe a general method of automatic annotation for training from positive and negative examples and use the method to train Rote, Naïve Bayes, and Conditional Random Field models (Section 2). We then examine how multiple extractions can be combined to form one consensus answer (Section 3). We compare fusion methods and show that frequency voting outperforms the single highest confidence answer by an average of 11% across the various extractors. Increasing the number of retrieved documents boosts the overall system accuracy as additional documents which mention the individual in question lead to higher recall. This improved recall more than compensates for a loss in per-extraction precision from these additional documents. Next, we present a method for cross-field bootstrapping (Section 4) which improves per-field accuracy by 7%. We demonstrate that a small training set with only the most relevant documents can be as effective as a larger training set with additional, less relevant documents (Section 5).

2 Training by Automatic Annotation

Typically, statistical extraction systems (such as HMMs and CRFs) are trained using hand-annotated data. Annotating the necessary data by hand is time-consuming and brittle, since it may require large-scale re-annotation when the annotation scheme changes. For the special case of Rote extractors, a more attractive alternative has been proposed by Brin (1998), Agichtein and Gravano (2000), and Ravichandran and Hovy (2002).

[1] Alternatively, Riloff (1996) trains on in-domain and out-of-domain texts and then has a human filtering step. Huffman (1995) proposes a method to train a different type of extraction system by example.

Proceedings of the 43rd Annual Meeting of the ACL, pages 483–490,
Ann Arbor, June 2005.

Essentially, for any text snippet of the form $A_1 p A_2 q A_3$, these systems estimate the probability that a relationship $r(p,q)$ holds between entities p and q, given the interstitial context, as[2]

$$P(r(p,q) \mid pA_2q) = P(r(p,q) \mid pA_2q) = \frac{\sum_{x,y \in T} c(xA_2y)}{\sum_x c(xA_2)}$$

That is, the probability of a relationship $r(p,q)$ is the number of times that pattern xA_2y predicts any relationship $r(x,y)$ in the training set T. $c(.)$ is the count. We will refer to x as the **hook**[3] and y as the **target**. In this paper, the hook is always an individual. Training a Rote extractor is straightforward given a set T of example relationships $r(x,y)$. For each hook, download a separate set of relevant documents (a **hook corpus**, D_x) from the Web.[4] Then for any particular pattern A_2 and an element x, count how often the pattern xA_2 predicts y and how often it retrieves a spurious $\bar{y}$.[5]

This annotation method extends to training other statistical models with positive examples, for example a Naïve Bayes (**NB**) unigram model. In this model, instead of looking for an exact A_2 pattern as above, each individual word in the pattern A_2 is used to predict the presence of a relationship.

$$\begin{aligned} P(r(p,q) \mid pA_2q) & \\ & \propto P(pA_2q \mid r(p,q))P(r(p,q)) \\ & = P(A_2 \mid r(p,q)) \\ & = \prod_{a \in A_2} P(a \mid r(p,q)) \end{aligned}$$

We perform add-lambda smoothing for out-of-vocabulary words and thus assign a positive probability to any sequence. As before, a set of relevant documents is downloaded for each particular hook. Then every hook and target is annotated. From that markup, we can pick out the interstitial A_2 patterns and calculate the necessary probabilities.

Since the NB model assigns a positive probability to every sequence, we need to pick out likely targets from those proposed by the NB extractor. We construct a **background model** which is a basic unigram language model, $P(A_2) = \prod_{a \in A_2} P(a)$. We then pick targets chosen by the confidence estimate

$$C^{\mathbf{NB}}(q) = log \frac{P(A_2 \mid r(p,q))}{P(A_2)}$$

However, this confidence estimate does not work-well in our dataset.

We propose to use negative examples to estimate $P(A_2 \mid \bar{r}(p,q))$[6] as well as $P(A_2 \mid r(p,q))$. For each relationship, we define the **target set** E_r to be all potential targets and model it using regular expressions.[7] In training, for each relationship $r(p,q)$, we markup the hook p, the target q, and all **spurious targets** ($\bar{q} \in \{E_r - q\}$) which provide negative examples. Targets can then be chosen with the following confidence estimate

$$C^{\mathbf{NB+E}}(q) = log \frac{P(A_2 \mid r(p,q))}{P(A_2 \mid \bar{r}(p,q))}$$

We call this **NB+E** in the following experiments.

The above process describes a general method for automatically annotating a corpus with positive and negative examples, and this corpus can be used to train statistical models that rely on annotated data.[8] In this paper, we test automatic annotation using Conditional Random Fields (**CRFs**) (Lafferty et al., 2001) which have achieved high performance for information extraction. CRFs are undirected graphical models that estimate the conditional probability of a state sequence given an output sequence

$$P(\mathbf{s} \mid \mathbf{o}) = \frac{1}{Z} \exp\left(\sum_{t=1}^{T} \sum_k \lambda_k f_k(s_{t-1}, s_t, \mathbf{o}, t) \right)$$

[2] The above Rote models also condition on the preceding and trailing words, for simplicity we only model interstitial words A_2.

[3] Following (Ravichandran and Hovy, 2002).

[4] In the following experiments we assume that there is one main object of interest p, for whom we want to find certain pieces of information $r(p,q)$, where r denotes the type of relationship (e.g., birthday) and q is a value (e.g., May 20th). We require one hook corpus for each hook, not a separate one for each relationship.

[5] Having a **functional constraint** $\forall \bar{q} \neq q, \bar{r}(p,\bar{q})$ makes this estimate much more reliable, but it is possible to use this method of estimation even when this constraint does not hold.

[6] $\bar{r}$ stands in for all other possible relationships (including no relationship) between p and q. $P(A_2 \mid \bar{r}(p,q))$ is estimated as $P(A_2 \mid r(p,q))$ is, except with spurious targets.

[7] e.g., $E_{birthyear} = \{\backslash d \backslash d \backslash d \backslash d\}$. This is the only source of human knowledge put into the system and required only around 4 hours of effort, less effort than annotating an entire corpus or writing information extraction rules.

[8] This corpus markup gives automatic annotation that yields noisier training data than manual annotation would.

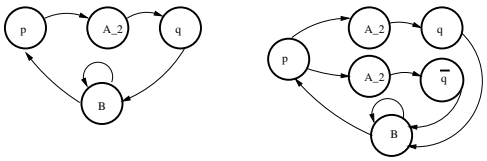

Figure 1: CRF state-transition graphs for extracting a relationship $r(p, q)$ from a sentence pA_2q. Left: **CRF** Extraction with a background model (B). Right: **CRF+E** As before but with spurious target prediction ($pA_2\bar{q}$).

	Aaron Neville	Frank Zappa
Birthday	January 24	December 21
Birth year	1941	1940
Occupation	Singer	Musician
Birthplace	New Orleans	Baltimore,Maryland
Year of Death	-	1993

Table 1: Two of 152 entries in the Biographic Database. Each entry contains incomplete information about various celebrities. Here, Aaron Neville's birth state is missing, and Frank Zappa could be equally well described as a guitarist or rock-star.

We use the Mallet system (McCallum, 2002) for training and evaluation of the CRFs. In order to examine the improvement by using negative examples, we train CRFs with two topologies (Figure 1). The first, **CRF**, models the target relationship and background sequences and is trained on a corpus where targets (positive examples) are annotated. The second, **CRF+E**, models the target relationship, spurious targets and background sequences, and it is trained on a corpus where targets (positive examples) as well as spurious targets (negative examples) are annotated.

Experimental Results

To test the performance of the different extractors, we collected a set of 152 semi-structured mini-biographies from an online site (www.infoplease.com), and used simple rules to extract a biographic fact database of birthday and month (henceforth birthday), birth year, occupation, birth place, and year of death (when applicable). An example of the data can be found in Table 1. In our system, we normalized birthdays, and performed capitalization normalization for the remaining fields. We did no further normalization, such as normalizing state names to their two letter acronyms (e.g., California → CA). Fifteen names were set aside as training data, and the rest were used for testing. For each name, 150 documents were downloaded from Google to serve as the hook corpus for either training or testing.[9]

In training, we automatically annotated documents using people in the training set as hooks, and in testing, tried to get targets that exactly matched what was present in the database. This is a very strict method of evaluation for three reasons. First, since the facts were automatically collected, they contain errors and thus the system is tested against wrong answers.[10] Second, 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 writer and the retrieved occupation might be novelist). Third, since the retrieved targets were not normalized, there system may have retrieved targets that were correct but were not recognized (e.g., the database birthplace is New York, and the system retrieves NY).

In testing, we rejected candidate targets that were not present in our target set models E_r. 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.

Before fusion (Section 3), we gathered all the facts extracted by the system and graded them in isolation. We present the per-extraction **precision**

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

We also present the **pseudo-recall**, which is the average number of times per person a correct target was extracted. It is difficult to calculate true recall without manual annotation of the entire corpus, since it cannot be known for certain how many times the document set contains the desired information.[11]

$$\textit{Pre-Fusion Pseudo-Recall} = \frac{\textit{\# Correct Extracted Targets}}{\#People}$$

The precision of each of the various extraction methods is listed in Table 2. The data show that on average the Rote method has the best precision,

[9]Name polyreference, along with ranking errors, result in the retrieval of undesired documents.

[10]These deficiencies in testing also have implications for training, since the models will be trained on annotated data that has errors. The phenomenon of missing and inaccurate data was most prevalent for occupation and birthplace relationships, though it was observed for other relationships as well.

[11]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.
Rote	.789	.355	.305	.510	.527	.497
NB+E	.423	.361	.255	.217	.088	.269
CRF	.509	.342	.219	.139	.267	.295
CRF+E	.680	.654	.246	.357	.314	.450

Table 2: Pre-Fusion Precision of extracted facts for various extraction systems, trained on 15 people each with 150 documents, and tested on 137 people each with 150 documents.

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

Table 3: Pre-Fusion Pseudo-Recall of extract facts with the identical training/testing set-up as above.

while the NB+E extractor has the worst. Training the CRF with negative examples (CRF+E) gave better precision in extracted information then training it without negative examples. Table 3 lists the pseudo-recall or average number of correctly extracted targets per person. The results illustrate that the Rote has the worst pseudo-recall, and the plain CRF, trained without negative examples, has the best pseudo-recall.

To test how the extraction precision changes as more documents are retrieved from the ranked results from Google, we created retrieval sets of 1, 5, 15, 30, 75, and 150 documents per person and repeated the above experiments with the CRF+E extractor. The data in Figure 2 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.

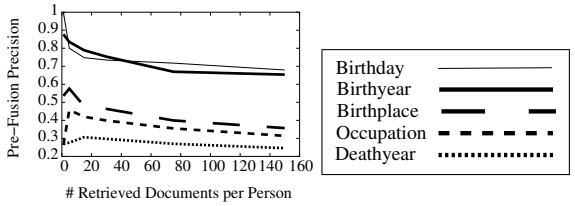

Figure 2: As more documents are retrieved per person, pre-fusion precision drops.

However, even though the extractor's precision drops, the data in Figure 3 indicate that there continue to be instances of the relevant biographic data.

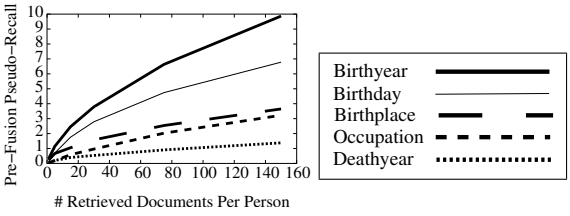

Figure 3: Pre-fusion pseudo-recall increases as more documents are added.

3 Cross-Document Information Fusion

The per-extraction performance was presented in Section 2, but the final task is to find the single correct target for each person.[12] In this section, we examine two basic methodologies for combining candidate targets. Masterson and Kushmerick (2003) propose **Best** which gives each candidate a score equal to its highest confidence extraction: $\textbf{Best}(x) = \operatorname{argmax}_x C(x)$.[13] We further consider **Voting**, which counts the number of times each candidate x was extracted: $\textbf{Vote}(x) = |C(x) > 0|$. Each of these methods ranks the candidate targets by score and chooses the top-ranked one.

The experimental setup used in the fusion experiments was the same as before: training on 15 people, and testing on 137 people. However, the post-fusion evaluation differs from the pre-fusion evaluation. After fusion, the system returns one consensus target for each person and thus the evaluation is on the **accuracy** of those targets. That is, missing tar-

[12] 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.

[13] $C(x)$ is either the confidence estimate (NB+E) or the probability score (Rote,CRF,CRF+E).

	Best	Vote
Rote	.364	.450
NB+E	.385	.588
CRF	.513	.624
CRF+E	.650	.678

Table 4: Average Accuracy of the Highest Confidence (Best) and Most Frequent (Vote) across five extraction fields.

gets are graded as wrong.[14]

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

Additionally, since the targets are ranked, we also calculated the mean reciprocal rank (MRR).[15] The data in Table 4 show the average system performance with the different fusion methods. Frequency voting gave anywhere from a 2% to a 20% improvement over picking the highest confidence candidate. CRF+E (the CRF trained with negative examples) was the highest performing system overall.

Birth Day		
	Fusion Accuracy	Fusion MRR
Rote Vote	.854	.877
NB+E Vote	.854	.889
CRF Vote	.650	.703
CRF+E Vote	**.883**	**.911**
Birth year		
Rote Vote	.387	.497
NB+E Vote	.778	.838
CRF Vote	.796	.860
CRF+E Vote	**.869**	**.876**
Occupation		
Rote Vote	.299	.405
NB+E Vote	**.642**	**.751**
CRF Vote	.606	.740
CRF+E Vote	.423	.553
Birthplace		
Rote Vote	.321	.338
NB+E Vote	**.474**	**.586**
CRF Vote	.321	.476
CRF+E Vote	.467	.560
Year of Death		
Rote Vote	.389	.389
NB+E Vote	.194	.383
CRF	**.750**	**.840**
CRF+E Vote	**.750**	.827

Table 5: Voting for information fusion, evaluated per person. CRF+E has best average performance (67.8%).

Table 5 shows the results of using each of these extractors to extract correct relationships from the top 150 ranked documents downloaded from the Web. CRF+E was a top performer in 3/5 of the cases. In the other 2 cases, the NB+E was the most successful, perhaps because NB+E's increased recall was more useful than CRF+E's improved precision.

[14]For year of death, we only graded cases where the person had died.

[15]The reciprocal rank = 1 / the rank of the correct target.

Retrieval Set Size and Performance

As with pre-fusion, we performed a set of experiments with different retrieval set sizes and used the CRF+E extraction system trained on 150 documents per person. The data in Figure 4 show that performance improves as the retrieval set size increases. Most of the gains come in the first 30 documents, where average performance increased from 14% (1 document) to 63% (30 documents). Increasing the retrieval set size to 150 documents per person yielded an additional 5% absolute improvement.

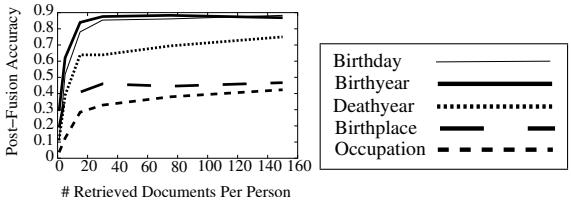

Figure 4: Fusion accuracy increases with more documents per person

Post-fusion errors come 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, which we measure as

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

The data in Figure 5 suggest that the decrease in missing targets is a significant contributing factor to the improvement in performance with increased document size. Missing targets were a major problem for Birthplace, constituting more than half the errors (32% at 150 documents).

4 Cross-Field Bootstrapping

Sections 2 and 3 presented methods for training separate extractors for particular relationships and for doing fusion across multiple documents. In this section, we leverage data interdependencies to improve performance.

The method we propose is to bootstrap across fields and use knowledge of one relationship to improve performance on the extraction of another. For

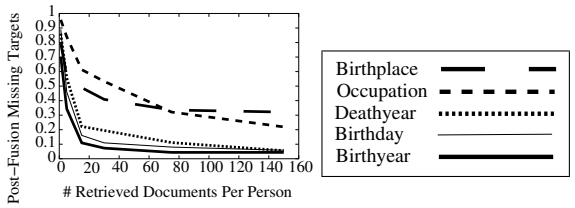

Figure 5: Additional documents decrease the number of post-fusion missing targets, targets which are never extracted in any document.

Birth year		
	Extraction Precision	Fusion Accuracy
CRF	.342	.797
+ birthday	.472	.861
CRF+E	.654	.869
+ birthday	**.809**	**.891**
Occupation		
	Extraction Precision	Fusion Accuracy
CRF	.219	.606
+ birthday	.217	.569
+ birth year(f)	21.9	.599
+ all	.214	.591
CRF+E	.246	.423
+ birthday	.325	.577
+ birth year(f)	**.387**	**.672**
+ all	.382	.642
Birthplace		
	Extraction Precision	Fusion Accuracy
CRF	.139	.321
+ birthday	.158	.372
+ birth year(f)	.156	.350
CRF+E	.357	.467
+ birthday	.350	.474
+ birth year(f)	.294	.350
+ occupation(f)	.314	.354
+ all	**.362**	**.532**

Table 6: Performance of Cross-Field Bootstrapping 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.

example, to extract birth year given knowledge of the birthday, in training we mark up each hook corpus D_x with the known birthday $b : birthday(x, b)$ and the target birth year $y : birthyear(x, y)$ and add an additional feature to the CRF that indicates whether the birthday has been seen in the sentence.[16] In testing, for each hook, we first find the birthday using the methods presented in the previous sections, annotate the corpus with the extracted birthday, and then apply the birth year CRF (see Figure 6 next page).

[16]The CRF state model doesn't change. When bootstrapping from multiple fields, we add the conjunctions of the fields as features.

Table 6 shows the effect of using this bootstrapped data to estimate other fields. Based on the relative performance of each of the individual extraction systems, we chose the following schedule for performing the bootstrapping: 1) Birthday, 2) Birth year, 3) Occupation, 4) Birthplace. We tried adding in all knowledge available to the system at each point in the schedule.[17] There are gains in accuracy for birth year, occupation and birthplace by using cross-field bootstrapping. The performance of the plain CRF+E averaged across all five fields is 67.4%, while for the best bootstrapped system it is 74.6%, a gain of 7%.

Doing bootstrapping in this way improves for people whose information is already partially correct. As a result, the percentage of people who have completely correct information improves to 37% from 13.8%, a gain of 24% over the non-bootstrapped CRF+E system. Additionally, erroneous extractions do not hurt accuracy on extraction of other fields. 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.

5 Training Set Size Reduction

One of the results from Section 2 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 improves 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. In a final set of experiments we looked at system performance when the extractor is trained on fewer than 150 documents per person.

The data in Figure 7 show that training on 30 documents per person yields around the same performance as training on 150 documents per person. Average performance when the system was trained on 30 documents per person is 70%, while average performance when trained on 150 documents per person is 68%. Most of this loss in performance comes from losses in occupation, but the other relationships

[17]This system has the extra knowledge of which fused method is the best for each relationship. This was assessed by inspection.

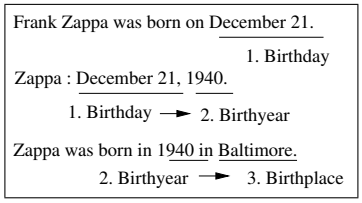

Figure 6: Cross-Field Bootstrapping: In step (1) The birthday, December 21, is extracted and the text marked. In step 2, cooccurrences with the discovered birthday make 1940 a better candidate for birthyear. In step (3), the discovered birthyear appears in contexts where the discovered birthday does not and improves extraction of birth place.

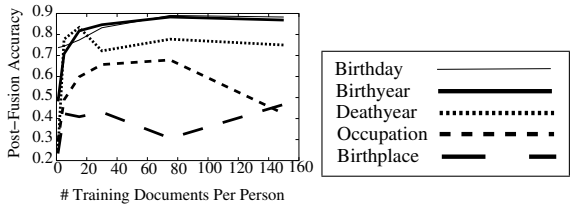

Figure 7: Fusion accuracy doesn't improve with more than 30 training documents per person.

have either little or no gain from training on additional documents. There are two possible 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 8) support this hypothesis since it appears that later instances are often contaminating earlier models.

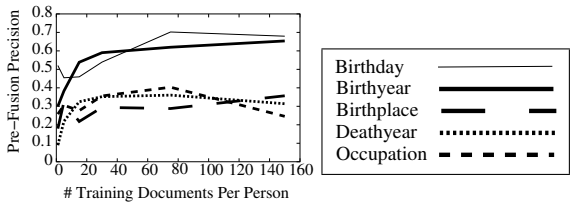

Figure 8: Pre-Fusion precision shows slight drops with increased training documents.

The data in Figure 9 suggest an alternate possibility that later documents also shift the prior toward a model where it is less likely that a relationship is observed as fewer targets are extracted.

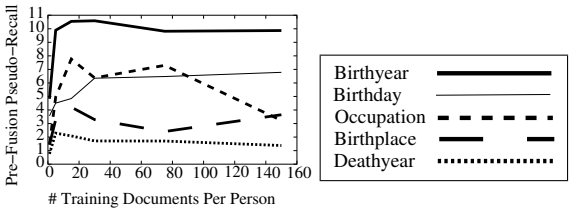

Figure 9: Pre-Fusion Pseudo-Recall also drops with increased training documents.

6 Related Work

The closest related work to the task of biographic fact extraction was done by Cowie et al. (2000) and Schiffman et al. (2001), who explore the problem of biographic summarization.

There has been rather limited published work in multi-document information extraction. The closest work to what we present here is Masterson and Kushmerick (2003), who perform multi-document information extraction trained on manually annotated training data and use Best Confidence to resolve each particular template slot. In summarizarion, many systems have examined the multi-document case. Notable systems are SUMMONS (Radev and McKeown, 1998) and RIPTIDE (White et al., 2001), which assume perfect extracted information and then perform closed domain summarization. Barzilay et al. (1999) does not explicitly extract facts, but instead picks out relevant repeated elements and combines them to obtain a summary which retains the semantics of the original.

In recent question answering research, 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.

There has been a considerable amount of work in training information extraction systems from annotated data since the mid-90s. The initial work in the field used lexico-syntactic template patterns learned using a variety of different empirical approaches (Riloff and Schmelzenbach, 1998; Huffman, 1995;

Soderland et al., 1995). Seymore et al. (1999) use HMMs for information extraction and explore ways to improve the learning process.

Nahm and Mooney (2002) suggest a method to learn word-to-word relationships across fields by doing data mining on information extraction results. Prager et al. (2004) uses knowledge of birth year to weed out candidate years of death that are impossible. Using the CRF extractors in our data set, this heuristic did not yield any improvement. More distantly related work for multi-field extraction suggests methods for combining information in graphical models across multiple extraction instances (Sutton et al., 2004; Bunescu and Mooney, 2004) .

7 Conclusion

This paper has presented new experimental methodologies and results for cross-document information fusion, focusing on the task of biographic fact extraction and has proposed a new method for cross-field bootstrapping. In particular, we have shown that automatic annotation can be used effectively to train statistical information extractors such Naïve Bayes and CRFs, and that CRF extraction accuracy can be improved by 5% with a negative example model. We looked at cross-document fusion and demonstrated that voting outperforms choosing the highest confidence extracted information by 2% to 20%. Finally, we introduced a cross-field bootstrapping method that improved average accuracy by 7%.

References

E. Agichtein and L. Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of ICDL*, pages 85–94.

R. Barzilay, K. R. McKeown, and M. Elhadad. 1999. Information fusion in the context of multi-document summarization. In *Proceedings of ACL*, pages 550–557.

E. Brill, J. Lin, M. Banko, S. Dumais, and A. Ng. 2001. Data-intensive question answering. In *Proceedings of TREC*, pages 183–189.

S. Brin. 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.

R. Bunescu and R. Mooney. 2004. Collective information extraction with relational markov networks. In *Proceedings of ACL*, pages 438–445.

C. L. A. Clarke, G. V. Cormack, and T. R. Lynam. 2001. Exploiting redundancy in question answering. In *Proceedings of SIGIR*, pages 358–365.

J. Cowie, S. Nirenburg, and H. Molina-Salgado. 2000. Generating personal profiles. In *The International Conference On MT And Multilingual NLP*.

T. Dalmas and B. Webber. 2004. Information fusion for answering factoid questions. In *Proceedings of 2nd CoLogNET-ElsNET Symposium. Questions and Answers: Theoretical Perspectives*.

D. Freitag and A. McCallum. 1999. Information extraction with hmms and shrinkage. In *Proceedings of the AAAI-99 Workshop on Machine Learning for Information Extraction*, pages 31–36.

S. B. Huffman. 1995. Learning information extraction patterns from examples. In *Working Notes of the IJCAI-95 Workshop on New Approaches to Learning for Natural Language Processing*, pages 127–134.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML*, pages 282–289.

T. R. Leek. 1997. Information extraction using hidden markov models. Master's Thesis, UC San Diego.

D. Masterson and N. Kushmerick. 2003. Information extraction from multi-document threads. In *Proceedings of ECML-2003: Workshop on Adaptive Text Extraction and Mining*, pages 34–41.

A. McCallum. 2002. Mallet: A machine learning for language toolkit.

U. Nahm and R. Mooney. 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.

J. Prager, J. Chu-Carroll, and K. Czuba. 2004. Question answering by constraint satisfaction: Qa-by-dossier with constraints. In *Proceedings of ACL*, pages 574–581.

D. R. Radev and K. R. McKeown. 1998. Generating natural language summaries from multiple on-line sources. *Computational Linguistics*, 24(3):469–500.

D. Ravichandran and E. Hovy. 2002. Learning surface text patterns for a question answering system. In *Proceedings of ACL*, pages 41–47.

E. Riloff and M. Schmelzenbach. 1998. An empirical approach to conceptual case frame acquisition. In *Proceedings of WVLC*, pages 49–56.

E. Riloff. 1996. Automatically Generating Extraction Patterns from Untagged Text. In *Proceedings of AAAI*, pages 1044–1049.

B. Schiffman, I. Mani, and K. J. Concepcion. 2001. Producing biographical summaries: Combining linguistic knowledge with corpus statistics. In *Proceedings of ACL*, pages 450–457.

K. Seymore, A. McCallum, and R. Rosenfeld. 1999. Learning hidden markov model structure for information extraction. In *AAAI'99 Workshop on Machine Learning for Information Extraction*, pages 37–42.

S. Soderland, D. Fisher, J. Aseltine, and W. Lehnert. 1995. CRYSTAL: Inducing a conceptual dictionary. In *Proceedings of IJCAI*, pages 1314–1319.

C. Sutton, K. Rohanimanesh, and A. McCallum. 2004. Dynamic conditional random fields: factorize probabilistic models for labeling and segmenting sequence data. In *Proceedings of ICML*.

M. White, T. Korelsky, C. Cardie, V. Ng, D. Pierce, and K. Wagstaff. 2001. Multi-document summarization via information extraction. In *Proceedings of HLT*.