Names and Similarities on the Web: Fact Extraction in the Fast Lane

Marius Paşca
Google Inc.
Mountain View, CA 94043
mars@google.com

Dekang Lin
Google Inc.
Mountain View, CA 94043
lindek@google.com

Jeffrey Bigham*
University of Washington
Seattle, WA 98195
jbigham@cs.washington.edu

Andrei Lifchits*
University of British Columbia
Vancouver, BC V6T 1Z4
alifchit@cs.ubc.ca

Alpa Jain*
Columbia University
New York, NY 10027
alpa@cs.columbia.edu

Abstract

In a new approach to large-scale extraction of facts from unstructured text, distributional similarities become an integral part of both the iterative acquisition of high-coverage contextual extraction patterns, and the validation and ranking of candidate facts. The evaluation measures the quality and coverage of facts extracted from one hundred million Web documents, starting from ten seed facts and using no additional knowledge, lexicons or complex tools.

1 Introduction

1.1 Background

The potential impact of structured fact repositories containing billions of relations among named entities on Web search is enormous. They enable the pursuit of new search paradigms, the processing of database-like queries, and alternative methods of presenting search results. The preparation of exhaustive lists of hand-written extraction rules is impractical given the need for domain-independent extraction of many types of facts from unstructured text. In contrast, the idea of bootstrapping for relation and information extraction was first proposed in (Riloff and Jones, 1999), and successfully applied to the construction of semantic lexicons (Thelen and Riloff, 2002), named entity recognition (Collins and Singer, 1999), extraction of binary relations (Agichtein and Gravano, 2000), and acquisition of structured data for tasks such as Question Answering (Lita and Carbonell, 2004; Fleischman et al., 2003). In the context of fact extraction, the resulting iterative acquisition framework starts from a small set of seed facts, finds contextual patterns that extract the seed facts from the underlying text collection, identifies a larger set of candidate facts that are extracted by the patterns, and adds the best candidate facts to the previous seed set.

1.2 Contributions

Figure 1 describes an architecture geared towards large-scale fact extraction. The architecture is similar to other instances of bootstrapping for information extraction. The main processing stages are the acquisition of contextual extraction patterns given the seed facts, acquisition of candidate facts given the extraction patterns, scoring and ranking of the patterns, and scoring and ranking of the candidate facts, a subset of which is added to the seed set of the next round.

Within the existing iterative acquisition framework, our first contribution is a method for automatically generating generalized contextual extraction patterns, based on dynamically-computed classes of similar words. Traditionally, the acquisition of contextual extraction patterns requires hundreds or thousands of consecutive iterations over the entire text collection (Lita and Carbonell, 2004), often using relatively expensive or restrictive tools such as shallow syntactic parsers (Riloff and Jones, 1999; Thelen and Riloff, 2002) or named entity recognizers (Agichtein and Gravano, 2000). Comparatively, generalized extraction patterns achieve exponentially higher coverage in early iterations. The extraction of large sets of candidate facts opens the possibility of fast-growth iterative extraction, as opposed to the de-facto strategy of conservatively growing the seed set by as few as five items (Thelen and Riloff, 2002) after each iteration.

*Work done during internships at Google Inc.

Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, pages 809–816,
Sydney, July 2006.

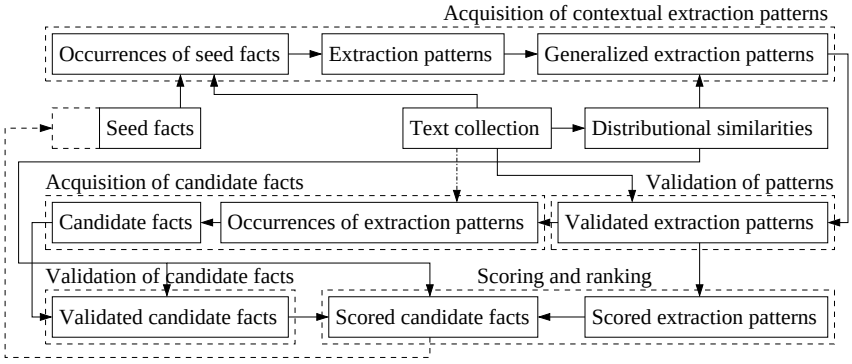

Figure 1: Large-scale fact extraction architecture

The second contribution of the paper is a method for domain-independent validation and ranking of candidate facts, based on a similarity measure of each candidate fact relative to the set of seed facts. Whereas previous studies assume clean text collections such as news corpora (Thelen and Riloff, 2002; Agichtein and Gravano, 2000; Hasegawa et al., 2004), the validation is essential for low-quality sets of candidate facts collected from noisy Web documents. Without it, the addition of spurious candidate facts to the seed set would result in a quick divergence of the iterative acquisition towards irrelevant information (Agichtein and Gravano, 2000). Furthermore, the finer-grained ranking induced by similarities is necessary in fast-growth iterative acquisition, whereas previously proposed ranking criteria (Thelen and Riloff, 2002; Lita and Carbonell, 2004) are implicitly designed for slow growth of the seed set.

2 Similarities for Pattern Acquisition

2.1 Generalization via Word Similarities

The extraction patterns are acquired by matching the pairs of phrases from the seed set into document sentences. The patterns consist of contiguous sequences of sentence terms, but otherwise differ from the types of patterns proposed in earlier work in two respects. First, the terms of a pattern are either regular words or, for higher generality, any word from a class of similar words. Second, the amount of textual context encoded in a pattern is limited to the sequence of terms between (i.e., infix) the pair of phrases from a seed fact that could be matched in a document sentence, thus excluding any context to the left (i.e., prefix) and to the right (i.e., postfix) of the seed.

The pattern shown at the top of Figure 2, which

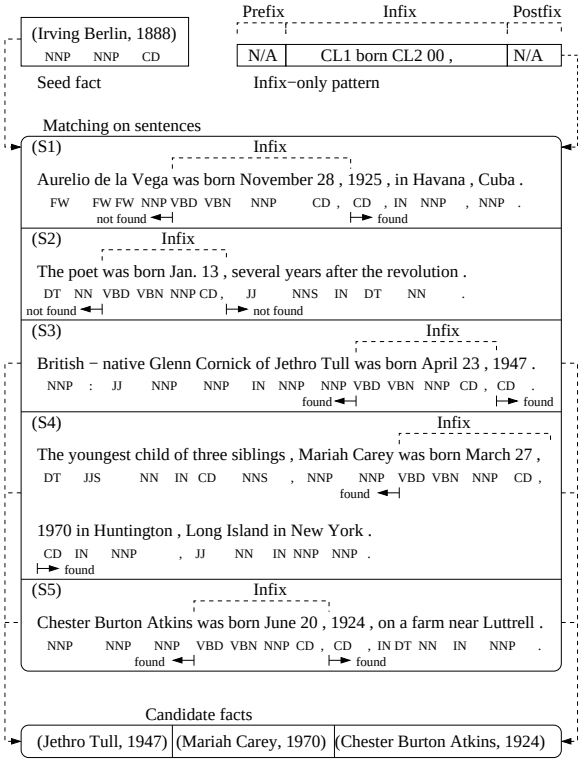

Figure 2: Extraction via infix-only patterns

contains the sequence [CL1 born CL2 00 .], illustrates the use of classes of distributionally similar words within extraction patterns. The first word class in the sequence, CL1, consists of words such as {*was*, *is*, *could*}, whereas the second class includes {*February*, *April*, *June*, *Aug.*, *November*} and other similar words. The classes of words are computed on the fly over all sequences of terms in the extracted patterns, on top of a large set of pairwise similarities among words (Lin, 1998) extracted in advance from around 50 million news articles indexed by the Google search engine over three years. All digits in both patterns and sentences are replaced with a common marker, such

that any two numerical values with the same number of digits will overlap during matching.

Many methods have been proposed to compute distributional similarity between words, e.g., (Hindle, 1990), (Pereira et al., 1993), (Grefenstette, 1994) and (Lin, 1998). Almost all of the methods represent a word by a feature vector, where each feature corresponds to a type of context in which the word appeared. They differ in how the feature vectors are constructed and how the similarity between two feature vectors is computed.

In our approach, we define the features of a word w to be the set of words that occurred within a small window of w in a large corpus. The context window of an instance of w consists of the closest non-stopword on each side of w and the stopwords in between. The value of a feature w' is defined as the pointwise mutual information between w' and w: $\text{PMI}(w', w) = -\log(\frac{P(w,w')}{P(w)P(w')})$. The similarity between two different words w_1 and w_2, $S(w_1, w_2)$, is then computed as the cosine of the angle between their feature vectors.

While the previous approaches to distributional similarity have only applied to words, we applied the same technique to proper names as well as words. The following are some example similar words and phrases with their similarities, as obtained from the Google News corpus:

• Carey: Higgins 0.39, Lambert 0.39, Payne 0.38, Kelley 0.38, Hayes 0.38, Goodwin 0.38, Griffin 0.38, Cummings 0.38, Hansen 0.38, Williamson 0.38, Peters 0.38, Walsh 0.38, Burke 0.38, Boyd 0.38, Andrews 0.38, Cunningham 0.38, Freeman 0.37, Stephens 0.37, Flynn 0.37, Ellis 0.37, Bowers 0.37, Bennett 0.37, Matthews 0.37, Johnston 0.37, Richards 0.37, Hoffman 0.37, Schultz 0.37, Steele 0.37, Dunn 0.37, Rowe 0.37, Swanson 0.37, Hawkins 0.37, Wheeler 0.37, Porter 0.37, Watkins 0.37, Meyer 0.37 [..];

• Mariah Carey: Shania Twain 0.38, Christina Aguilera 0.35, Sheryl Crow 0.35, Britney Spears 0.33, Celine Dion 0.33, Whitney Houston 0.32, Justin Timberlake 0.32, Beyonce Knowles 0.32, Bruce Springsteen 0.30, Faith Hill 0.30, LeAnn Rimes 0.30, Missy Elliott 0.30, Aretha Franklin 0.29, Jennifer Lopez 0.29, Gloria Estefan 0.29, Elton John 0.29, Norah Jones 0.29, Missy Elliot 0.29, Alicia Keys 0.29, Avril Lavigne 0.29, Kid Rock 0.28, Janet Jackson 0.28, Kylie Minogue 0.28, Beyonce 0.27, Enrique Iglesias 0.27, Michelle Branch 0.27 [..];

• Jethro Tull: Motley Crue 0.28, Black Crowes 0.26, Pearl Jam 0.26, Silverchair 0.26, Black Sabbath 0.26, Doobie Brothers 0.26, Judas Priest 0.26, Van Halen 0.25, Midnight Oil 0.25, Pere Ubu 0.24, Black Flag 0.24, Godsmack 0.24, Grateful Dead 0.24, Grand Funk Railroad 0.24, Smashing Pumpkins 0.24, Led Zeppelin 0.24, Aerosmith 0.24, Limp Bizkit 0.24, Counting Crows 0.24, Echo And The Bunnymen 0.24, Cold Chisel 0.24, Thin Lizzy 0.24 [..].

To our knowledge, the only previous study that embeds similarities into the acquisition of extraction patterns is (Stevenson and Greenwood, 2005). The authors present a method for computing pairwise similarity scores among large sets of potential syntactic (subject-verb-object) patterns, to detect centroids of mutually similar patterns. By assuming the syntactic parsing of the underlying text collection to generate the potential patterns in the first place, the method is impractical on Web-scale collections. Two patterns, e.g. *chairman-resign* and *CEO-quit*, are similar to each other if their components are present in an external hand-built ontology (i.e., WordNet), and the similarity among the components is high over the ontology. Since general-purpose ontologies, and WordNet in particular, contain many classes (e.g., *chairman* and *CEO*) but very few instances such as *Osasuna*, *Crewe* etc., the patterns containing an instance rather than a class will not be found to be similar to one another. In comparison, the classes and instances are equally useful in our method for generalizing patterns for fact extraction. We merge basic patterns into generalized patterns, regardless of whether the similar words belong, as classes or instances, in any external ontology.

2.2 Generalization via Infix-Only Patterns

By giving up the contextual constraints imposed by the prefix and postfix, infix-only patterns represent the most aggressive type of extraction patterns that still use contiguous sequences of terms. In the absence of the prefix and postfix, the outer boundaries of the fact are computed separately for the beginning of the first (left) and end of the second (right) phrases of the candidate fact. For generality, the computation relies only on the part-of-speech tags of the current seed set. Starting forward from the right extremity of the infix, we collect a growing sequence of terms whose part-of-speech tags are [P_1+ P_2+ .. P_n+], where the

notation P_i+ represents one or more consecutive occurrences of the part-of-speech tag P_i. The sequence [P_1 P_2 .. P_n] must be exactly the sequence of part of speech tags from the right side of one of the seed facts. The point where the sequence cannot be grown anymore defines the boundary of the fact. A similar procedure is applied backwards, starting from the left extremity of the infix. An infix-only pattern produces a candidate fact from a sentence only if an acceptable sequence is found to the left and also to the right of the infix.

Figure 2 illustrates the process on the infix-only pattern mentioned earlier, and one seed fact. The part-of-speech tags for the seed fact are [NNP NNP] and [CD] for the left and right sides respectively. The infix occurs in all sentences. However, the matching of the part-of-speech tags of the sentence sequences to the left and right of the infix, against the part-of-speech tags of the seed fact, only succeeds for the last three sentences. It fails for the first sentence S_1 to the left of the infix, because [.. NNP] (for *Vega*) does not match [NNP NNP]. It also fails for the second sentence S_2 to both the left and the right side of the infix, since [.. NN] (for *poet*) does not match [NNP NNP], and [JJ ..] (for *several*) does not match [CD].

3 Similarities for Validation and Ranking

3.1 Revisiting Standard Ranking Criteria

Because some of the acquired extraction patterns are too generic or wrong, all approaches to iterative acquisition place a strong emphasis on the choice of criteria for ranking. Previous literature quasi-unanimously assesses the quality of each candidate fact based on the number and quality of the patterns that extract the candidate fact (more is better); and the number of seed facts extracted by the same patterns (again, more is better) (Agichtein and Gravano, 2000; Thelen and Riloff, 2002; Lita and Carbonell, 2004). However, our experiments using many variations of previously proposed scoring functions suggest that they have limited applicability in large-scale fact extraction, for two main reasons. The first is that it is impractical to perform hundreds of acquisition iterations on terabytes of text. Instead, one needs to grow the seed set aggressively in each iteration. Previous scoring functions were implicitly designed for cautious acquisition strategies (Collins and Singer, 1999), which expand the seed set very slowly across consecutive iterations. In that case, it makes sense to single out a small number of best candidates, among the other available candidates. Comparatively, when 10,000 candidate facts or more need to be added to a seed set of 10 seeds as early as after the first iteration, it is difficult to distinguish the quality of extraction patterns based, for instance, only on the percentage of the seed set that they extract. The second reason is the noisy nature of the Web. A substantial number of factors can and will concur towards the worst-case extraction scenarios on the Web. Patterns of apparently high quality turn out to produce a large quantity of erroneous "facts" such as (*A-League*, *1997*), but also the more interesting (*Jethro Tull*, *1947*) as shown earlier in Figure 2, or (*Web Site David*, *1960*) or (*New York*, *1831*). As for extraction patterns of average or lower quality, they will naturally lead to even more spurious extractions.

3.2 Ranking of Extraction Patterns

The intuition behind our criteria for ranking generalized pattern is that patterns of higher precision tend to contain words that are indicative of the relation being mined. Thus, a pattern is more likely to produce good candidate facts if its infix contains the words *language* or *spoken* if extracting Language-SpokenIn-Country facts, or the word *capital* if extracting City-CapitalOf-Country relations. In each acquisition iteration, the scoring of patterns is a two-pass procedure. The first pass computes the normalized frequencies of all words excluding stopwords, over the entire set of extraction patterns. The computation applies separately to the prefix, infix and postfix of the patterns. In the second pass, the score of an extraction pattern is determined by the words with the highest frequency score in its prefix, infix and postfix, as computed in the first pass and adjusted for the relative distance to the start and end of the infix.

3.3 Ranking of Candidate Facts

Figure 3 introduces a new scheme for assessing the quality of the candidate facts, based on the computation of similarity scores for each candidate relative to the set of seed facts. A candidate fact, e.g., (*Richard Steele*, *1672*), is similar to the seed set if both its phrases, i.e., *Richard Steele* and *1672*, are similar to the corresponding phrases (*John Lennon* or *Stephen Foster* in the case of *Richard Steele*) from the seed facts. For a phrase of a candidate fact to be assigned a non-default (non-minimum)

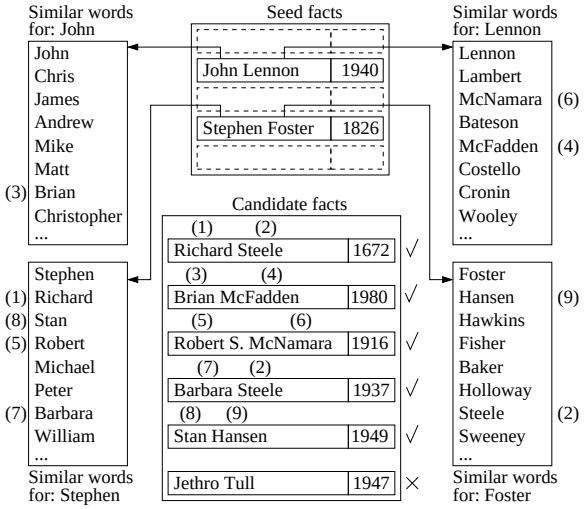

Figure 3: The role of similarities in estimating the quality of candidate facts

similarity score, the words at its extremities must be similar to one or more words situated at the same positions in the seed facts. This is the case for the first five candidate facts in Figure 3. For example, the first word *Richard* from one of the candidate facts is similar to the first word *John* from one of the seed facts. Concurrently, the last word *Steele* from the same phrase is similar to *Foster* from another seed fact. Therefore *Robert Foster* is similar to the seed facts. The score of a phrase containing N words is:

$$\begin{cases} C_1 + \sum_{i=1}^{N} log(1 + Sim_i) & \text{, if } Sim_{1,N} > 0 \\ C_2 & \text{, otherwise.} \end{cases}$$

where Sim_i is the similarity of the component word at position i in the phrase, and C_1 and C_2 are scaling constants such that $C_2 \ll C_1$. Thus, the similarity score of a candidate fact aggregates individual word-to-word similarity scores, for the left side and then for the right side of a candidate fact. In turn, the similarity score of a component word Sim_i is higher if: a) the computed word-to-word similarity scores are higher relative to words at the same position i in the seeds; and b) the component word is similar to words from more than one seed fact.

The similarity scores are one of a linear combination of features that induce a ranking over the candidate facts. Three other domain-independent features contribute to the final ranking: a) a phrase completeness score computed statistically over the entire set of candidate facts, which demotes candidate facts if any of their two sides is likely to be incomplete (e.g., *Mary Lou* vs. *Mary Lou Retton*, or *John F.* vs. *John F. Kennedy*); b) the average PageRank value over all documents from which the candidate fact is extracted; and c) the pattern-based scores of the candidate fact. The latter feature converts the scores of the patterns extracting the candidate fact into a score for the candidate fact. For this purpose, it considers a fixed-length window of words around each match of a candidate fact in some sentence from the text collection. This is equivalent to analyzing all sentence contexts from which a candidate fact can be extracted. For each window, the word with the highest frequency score, as computed in the first pass of the procedure for scoring the patterns, determines the score of the candidate fact in that context. The overall pattern-based score of a candidate fact is the sum of the scores over all its contexts of occurrence, normalized by the frequency of occurrence of the candidate over all sentences.

Besides inducing a ranking over the candidate facts, the similarity scores also serve as a validation filter over the candidate facts. Indeed, any candidates that are not similar to the seed set can be filtered out. For instance, the elimination of (*Jethro Tull, 1947*) is a side effect of verifying that *Tull* is not similar to any of the last-position words from phrases in the seed set.

4 Evaluation

4.1 Data

The source text collection consists of three chunks W_1, W_2, W_3 of approximately 100 million documents each. The documents are part of a larger snapshot of the Web taken in 2003 by the Google search engine. All documents are in English. The textual portion of the documents is cleaned of Html, tokenized, split into sentences and part-of-speech tagged using the TnT tagger (Brants, 2000).

The evaluation involves facts of type Person-BornIn-Year. The reasons behind the choice of this particular type are threefold. First, many Person-BornIn-Year facts are probably available on the Web (as opposed to, e.g., City-CapitalOf-Country facts), to allow for a good stress test for large-scale extraction. Second, either side of the facts (Person and Year) may be involved in many other types of facts, such that the extraction would easily divergence unless it performs correctly. Third, the phrases from one side (Person) have an utility in their own right, for lexicon

Table 1: Set of seed Person-BornIn-Year facts

Name	Year	Name	Year
Paul McCartney	1942	John Lennon	1940
Vincenzo Bellini	1801	Stephen Foster	1826
Hoagy Carmichael	1899	Irving Berlin	1888
Johann Sebastian Bach	1685	Bela Bartok	1881
Ludwig van Beethoven	1770	Bob Dylan	1941

construction or detection of person names.

The Person-BornIn-Year type is specified through an initial set of 10 seed facts shown in Table 1. Similarly to source documents, the facts are also part-of-speech tagged.

4.2 System Settings

In each iteration, the case-insensitive matching of the current set of seed facts onto the sentences produces basic patterns. The patterns are converted into generalized patterns. The length of the infix may vary between 1 and 6 words. Potential patterns are discarded if the infix contains only stopwords.

When a pattern is retained, it is used as an infix-only pattern, and allowed to generate at most 600,000 candidate facts. At the end of an iteration, approximately one third of the validated candidate facts are added to the current seed set. Consequently, the acquisition expands the initial seed set of 10 facts to 100,000 facts (after iteration 1) and then to one million facts (after iteration 2) using chunk W_1.

4.3 Precision

A separate baseline run extracts candidate facts from the text collection following the traditional iterative acquisition approach. Pattern generalization is disabled, and the ranking of patterns and facts follows strictly the criteria and scoring functions from (Thelen and Riloff, 2002), which are also used in slightly different form in (Lita and Carbonell, 2004) and (Agichtein and Gravano, 2000). The theoretical option of running thousands of iterations over the text collection is not viable, since it would imply a non-justifiable expense of our computational resources. As a more realistic compromise over overly-cautious acquisition, the baseline run retains as many of the top candidate facts as the size of the current seed, whereas (Thelen and Riloff, 2002) only add the top five candidate facts to the seed set after each iteration. The evaluation considers all 80, a sample of the 320, and another sample of the 10,240 facts retained after iterations 3, 5 and 10 respectively. The correctness assessment of each fact consists in manually finding some Web page that contains clear evidence that the fact is correct. If no such page exists, the fact is marked as incorrect. The corresponding precision values after the three iterations are 91.2%, 83.8% and 72.9%.

For the purpose of evaluating the precision of our system, we select a sample of facts from the entire list of one million facts extracted from chunk W_1, ranked in decreasing order of their computed scores. The sample is generated automatically from the top of the list to the bottom, by retaining a fact and skipping the following consecutive N facts, where N is incremented at each step. The resulting list, which preserves the relative order of the facts, contains 1414 facts. The 115 facts for which a Web search engine does not return any documents, when the name (as a phrase) and the year are submitted together in a conjunctive query, are discarded from the sample of 1414 facts. In those cases, the facts were acquired from the 2003 snapshot of the Web, but queries are submitted to a search engine with access to current Web documents, hence the difference when some of the 2003 documents are no longer available or indexable.

Based on the sample set, the average precision of the list of one million facts extracted from chunk W_1 is 98.5% over the top 1/100 of the list, 93.1% over the top half of the list, and 88.3% over the entire list of one million facts. Table 2 shows examples of erroneous facts extracted from chunk W_1. Causes of errors include incorrect approximations of the name boundaries (e.g., *Alma* in *Alma Theresa Rausch* is incorrectly tagged as an adjective), and selection of the wrong year as birth year (e.g., for *Henry Lumbar*).

In the case of famous people, the extracted facts tend to capture the correct birth year for several variations of the names, as shown in Table 3. Conversely, it is not necessary that a fact occur with high frequency in order for it to be extracted, which is an advantage over previous approaches that rely strongly on redundancy (cf. (Cafarella et al., 2005)). Table 4 illustrates a few of the correctly extracted facts that occur rarely on the Web.

4.4 Recall

In contrast to the assessment of precision, recall can be evaluated automatically, based on external

Table 2: Incorrect facts extracted from the Web

Spurious Fact	Context in Source Sentence
(Theresa Rausch, 1912)	Alma Theresa Rausch was born on 9 March 1912
(Henry Lumbar, 1937)	Henry Lumbar was born 1861 and died 1937
(Concepcion Paxety, 1817)	Maria de la Concepcion Paxety b. 08 Dec. 1817 St. Aug., FL.
(Mae Yaeger, 1872)	Ella May/Mae Yaeger was born 20 May 1872 in Mt.
(Charles Whatley, 1821)	Long, Charles Whatley b. 16 FEB 1821 d. 29 AUG
(HOLT George W. Holt, 1845)	HOLT (new line) George W. Holt was born in Alabama in 1845
(David Morrish Canadian, 1953)	David Morrish (new line) Canadian, b. 1953
(Mary Ann, 1838)	had a daughter, Mary Ann, who was born in Tennessee in 1838
(Mrs. Blackmore, 1918)	Mrs. Blackmore was born April 28, 1918, in Labaddiey

Table 3: Birth years extracted for both pseudonyms and corresponding real names

Pseudonym	Real Name	Year
Gloria Estefan	Gloria Fajardo	1957
Nicolas Cage	Nicolas Kim Coppola	1964
Ozzy Osbourne	John Osbourne	1948
Ringo Starr	Richard Starkey	1940
Tina Turner	Anna Bullock	1939
Tom Cruise	Thomas Cruise Mapother IV	1962
Woody Allen	Allen Stewart Konigsberg	1935

lists of birth dates of various people. We start by collecting two gold standard sets of facts. The first set is a random set of 609 actors and their birth years from a Web compilation (Gold$_A$). The second set is derived from the set of questions used in the Question Answering track (Voorhees and Tice, 2000) of the Text REtrieval Conference from 1999 through 2002. Each question asking for the birth date of a person (e.g., *"What year was Robert Frost born?"*) results in a pair containing the person's name and the birth year specified in the answer keys. Thus, the second gold standard set contains 17 pairs of people and their birth years (Gold$_T$). Table 5 shows examples of facts in each of the gold standard sets.

Table 6 shows two types of recall scores computed against the gold standard sets. The recall scores over ∩Gold take into consideration only the set of person names from the gold standard with some extracted year(s). More precisely, given that some years were extracted for a person name, it verifies whether they include the year specified in the gold standard for that person name. Comparatively, the recall score denoted *All*Gold is computed over the entire set of names from the gold standard.

Table 4: Extracted facts that occur infrequently

Fact	Source Domain
(Irvine J Forcier, 1912)	geocities.com
(Marie Louise Azelie Chabert, 1861)	vienici.com
(Jacob Shalles, 1750)	selfhost.com
(Robert Chester Claggett, 1898)	rootsweb.com
(Charoltte Mollett, 1843)	rootsweb.com
(Nora Elizabeth Curran, 1979)	jimtravis.com

Table 5: Composition of gold standard sets

Gold Set	Composition and *Examples of Facts*	
Gold$_A$	Actors (Web compilation)	Nr. facts: 609
	(Andie MacDowell, 1958), (Doris Day, 1924), (Diahann Carroll, 1935)	
Gold$_T$	People (TREC QA track)	Nr. facts: 17
	(Davy Crockett, 1786), (Julius Caesar, 100 B.C.), (King Louis XIV, 1638)	

For the Gold$_A$ set, the size of the ∩Gold set of person names changes little when the facts are extracted from chunk W_1 vs. W_2 vs. W_3. The recall scores over ∩Gold exhibit little variation from one Web chunk to another, whereas the *All*Gold score is slightly higher on the W_3 chunk, probably due to a higher number of documents that are relevant to the extraction task. When the facts are extracted from a combination of two or three of the available Web chunks, the recall scores computed over *All*Gold are significantly higher as the size of the ∩Gold set increases. In comparison, the recall scores over the growing ∩Gold set increases slightly with larger evaluation sets. The highest value of the recall score for Gold$_A$ is 89.9% over the ∩Gold set, and 70.7% over *All*Gold. The smaller size of the second gold standard set, Gold$_T$, explains the higher variation of the values shown in the lower portion of Table 6.

4.5 Comparison to Previous Results

Another recent approach specifically addresses the problem of extracting facts from a similarly-sized collection of Web documents. In (Cafarella et al., 2005), manually-prepared extraction rules are applied to a collection of 60 million Web documents to extract entities of types Company and Country, as well as facts of type Person-CeoOf-Company and City-CapitalOf-Country. Based on manual evaluation of precision and recall, a total of 23,128 company names are extracted at precision of 80%; the number decreases to 1,116 at precision of 90%. In addition, 2,402 Person-CeoOf-Company facts

Table 6: Automatic evaluation of recall, over two gold standard sets Gold_A (609 person names) and Gold_T (17 person names)

Gold Set	Input Data (Web Chunk)	Recall (%)	
		∩Gold	*All*Gold
Gold_A	W_1	86.4	49.4
	W_2	85.0	50.5
	W_3	86.3	54.1
	W_1+W_2	88.5	64.5
	W_1+W_2+W_3	89.9	70.7
Gold_T	W_1	81.8	52.9
	W_2	90.0	52.9
	W_3	100.0	64.7
	W_1+W_2	81.8	52.9
	W_1+W_2+W_3	91.6	64.7

are extracted at precision 80%. The recall value is 80% at precision 90%. Recall is evaluated against the set of company names extracted by the system, rather than an external gold standard with pairs of a CEO and a company name. As such, the resulting metric for evaluating recall used in (Cafarella et al., 2005) is somewhat similar to, though more relaxed than, the recall score over the ∩Gold set introduced in the previous section.

5 Conclusion

The combination of generalized extraction patterns and similarity-driven ranking criteria results in a fast-growth iterative approach for large-scale fact extraction. From 10 Person-BornIn-Year facts and no additional knowledge, a set of one million facts of the same type is extracted from a collection of 100 million Web documents of arbitrary quality, with a precision around 90%. This corresponds to a growth ratio of 100,000:1 between the size of the extracted set of facts and the size of the initial set of seed facts. To our knowledge, the growth ratio and the number of extracted facts are several orders of magnitude higher than in any of the previous studies on fact extraction based on either hand-written extraction rules (Cafarella et al., 2005), or bootstrapping for relation and information extraction (Agichtein and Gravano, 2000; Lita and Carbonell, 2004). The next research steps converge towards the automatic construction of a searchable repository containing billions of facts regarding people.

References

E. Agichtein and L. Gravano. 2000. Snowball: Extracting relations from large plaintext collections. In *Proceedings of the 5th ACM International Conference on Digital Libraries (DL-00)*, pages 85–94, San Antonio, Texas.

T. Brants. 2000. TnT - a statistical part of speech tagger. In *Proceedings of the 6th Conference on Applied Natural Language Processing (ANLP-00)*, pages 224–231, Seattle, Washington.

M. Cafarella, D. Downey, S. Soderland, and O. Etzioni. 2005. KnowItNow: Fast, scalable information extraction from the web. In *Proceedings of the Human Language Technology Conference (HLT-EMNLP-05)*, pages 563–570, Vancouver, Canada.

M. Collins and Y. Singer. 1999. Unsupervised models for named entity classification. In *Proceedings of the 1999 Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-99)*, pages 189–196, College Park, Maryland.

M. Fleischman, E. Hovy, and A. Echihabi. 2003. Offline strategies for online question answering: Answering questions before they are asked. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL-03)*, pages 1–7, Sapporo, Japan.

G. Grefenstette. 1994. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publishers, Boston, Massachusetts.

T. Hasegawa, S. Sekine, and R. Grishman. 2004. Discovering relations among named entities from large corpora. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 415–422, Barcelona, Spain.

D. Hindle. 1990. Noun classification from predicate-argument structures. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics (ACL-90)*, pages 268–275, Pittsburgh, Pennsylvania.

D. Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of the 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics (COLING-ACL-98)*, pages 768–774, Montreal, Quebec.

L. Lita and J. Carbonell. 2004. Instance-based question answering: A data driven approach. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-04)*, pages 396–403, Barcelona, Spain.

F. Pereira, N. Tishby, and L. Lee. 1993. Distributional clustering of english words. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics (ACL-93)*, pages 183–190, Columbus, Ohio.

E. Riloff and R. Jones. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the 16th National Conference on Artificial Intelligence (AAAI-99)*, pages 474–479, Orlando, Florida.

M. Stevenson and M. Greenwood. 2005. A semantic approach to IE pattern induction. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL-05)*, pages 379–386, Ann Arbor, Michigan.

M. Thelen and E. Riloff. 2002. A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-02)*, pages 214–221, Philadelphia, Pennsylvania.

E.M. Voorhees and D.M. Tice. 2000. Building a question-answering test collection. In *Proceedings of the 23rd International Conference on Research and Development in Information Retrieval (SIGIR-00)*, pages 200–207, Athens, Greece.