# Real-time Population of Knowledge Bases: Opportunities and Challenges

**Ndapandula Nakashole, Gerhard Weikum**
Max Planck Institute for Informatics
Saarbrücken, Germany
{nnakasho,weikum}@mpi-inf.mpg.de

## Abstract

Dynamic content is a frequently accessed part of the Web. However, most information extraction approaches are batch-oriented, thus not effective for gathering rapidly changing data. This paper proposes a model for fact extraction in real-time. Our model addresses the difficult challenges that timely fact extraction on frequently updated data entails. We point out a naive solution to the main research question and justify the choices we make in the model we propose.

## 1 Introduction

**Motivation.** Dynamic content is an important part of the Web, it accounts for a substantial amount of Web traffic. For example, much time is spent reading news, blogs and user comments in social media. To extract meaningful relational facts from text, several approaches have emerged (Dalvi 2009; Etzioni 2008; Weikum 2009). These approaches aim to construct large knowledge bases of facts. However, most knowledge bases are built in a batch-oriented manner. Facts are extracted from a snapshot of a corpus and some weeks or months later, another round of facts are extracted from a new snapshot. This process is repeated at irregular and long intervals resulting in incomplete and partly stale knowledge bases. If knowledge bases were updated in-sync with Web changes, time spent examining long Web articles for facts would be reduced significantly.

Current Web information extraction systems rely on snapshots such as the ClueWeb09 crawl[1] (Fader 2011; Nakashole 2011) which is now three years old. The NELL system (Carlson 2010) follows a "never-ending" extraction model with the extraction process going on 24 hours a day. However NELL's focus is on language learning by iterating mostly on the same ClueWeb09 corpus. Our focus is on capturing the latest information enriching it into the form of relational facts. Web-based news aggregators such as Google News and Yahoo! News present up-to-date information from various news sources. However, news aggregators present headlines and short text snippets. Our focus is on presenting this information as relational facts that can facilitate relational queries spanning new and historical data.

**Challenges.** Timely knowledge extraction from frequently updated sources entails a number of challenges:

1. **Relation Discovery:** We need to discover and maintain a *dynamically evolving open set of relations*. This needs to go beyond common relations such as "bornIn" or "headquateredIn". For example, major knowledge bases lack potentially interesting relations like "firedFrom" or "hasGoddaughter". For completeness, we need to automatically discover such relations. Furthermore, we may occasionally pick up completely new relations, such as the new notion of a person ”unfriending” another person in an online community. The TextRunner/Reverb project has addressed this challenge to some extent(Banko 2007; Fader 2011) , but the output has the form of verbal phrases, rather than typed relations and it is computed in a batch-oriented manner.

[1] lemurproject.org/clueweb09.php/

*Proc. of the Joint Workshop on Automatic Knowledge Base Construction & Web-scale Knowledge Extraction (AKBC-WEKEX)*, pages 41–45,
NAACL-HLT, Montréal, Canada, June 7-8, 2012.

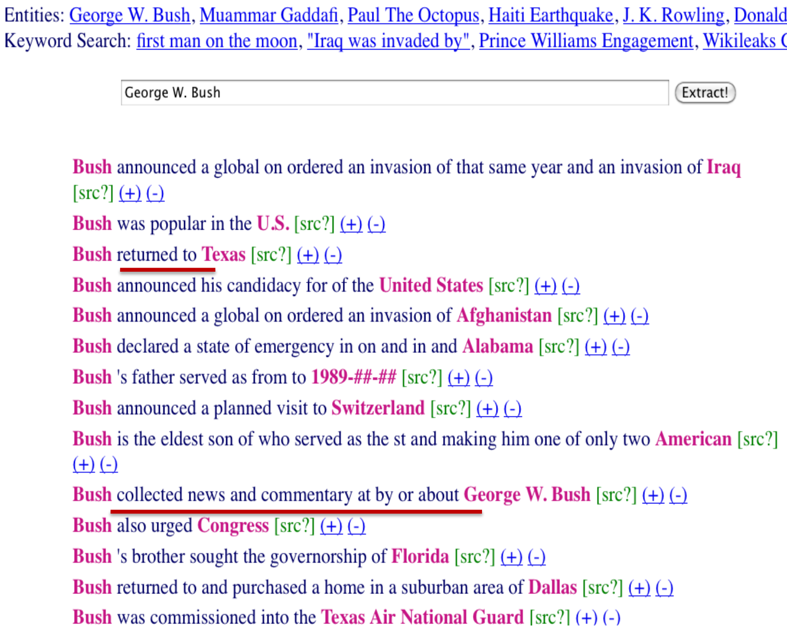
2. **Dynamic Entity Recognition:** We need to map noun phrases in text to entities in a dictionary of entities provided by knowledge bases. For example, when we encounter the noun phrase “Jeff Dean”, we need to map it to the correct entity, which can either be the Google engineer or the rock musician. However, knowledge bases are incomplete in the entities they contain, due to newly emerging entities and entities in the long tail. For example, Jeff Dean the Google engineer, does not have a Wikipedia page, and thus is missing in Wikipedia-derived knowledge bases. We need to recognize and handle out-of-knowledg-base entities as they emerge.

3. **Extraction under Time Constraints:** Due to the need for timely extraction, our extraction methods need to produce results under *time constraints*. We discuss ideas for optimizing execution time.

Our goal is to design a model for fact extraction that adequately addresses these three main challenges.
**Overview.** Section 2 presents a naive baseline and points out its shortcomings. Section 3 gives an overview of our approach. Sections 4 and 5 describe our solutions for generating open sets of relations and entities. Section 6 describes our proposal for dealing with time constraints. Finally, Section 7 concludes.

## 2 Naive Approach

One straightforward approach that embodies the concepts of open sets of relations and entities, is to greedily extract all pairs of noun phrases co-occurring within a sentence. Each such extracted co-occurrence would then be considered to be a relational facts. However, this results meaningless facts. More importantly, even for the facts that are supposed to be meaningful, they do not exhibit real semantics. Figure 1 is a screenshot of a prototype where we applied this approach to the ClueWeb corpus. From Figure 1, we can spot meaningless patterns which should not be extracted (see marked lines). We can also spot patterns that can benefit from having semantics associated with them. Such semantics would indicate what the pattern actually means. For example, synonymy semantics could indicate that ”return to” is the same as ”traveled to, trip to, ...”, and typing semantics could reveal that it is a pattern that applies to a person and a location. Such lexical semantics would be useful for both a user looking at the data, and also for an application using the data.



Entities: George W. Bush, Muammar Gaddafi, Paul The Octopus, Haiti Earthquake, J. K. Rowling, Donald
Keyword Search: first man on the moon, "Iraq was invaded by", Prince Williams Engagement, Wikileaks C

George W. Bush

Bush announced a global on ordered an invasion of that same year and an invasion of Iraq [src?] (+) (-)
Bush was popular in the U.S. [src?] (+) (-)
Bush returned to Texas [src?] (+) (-)
Bush announced his candidacy for of the United States [src?] (+) (-)
Bush announced a global on ordered an invasion of Afghanistan [src?] (+) (-)
Bush declared a state of emergency in on and in and Alabama [src?] (+) (-)
Bush 's father served as from to 1989-##-## [src?] (+) (-)
Bush announced a planned visit to Switzerland [src?] (+) (-)
Bush is the eldest son of who served as the st and making him one of only two American [src?] (+) (-)
Bush collected news and commentary at by or about George W. Bush [src?] (+) (-)
Bush also urged Congress [src?] (+) (-)
Bush 's brother sought the governorship of Florida [src?] (+) (-)
Bush returned to and purchased a home in a suburban area of Dallas [src?] (+) (-)
Bush was commissioned into the Texas Air National Guard [src?] (+) (-)

Figure 1: Noisy triples obtained from naive approach

TextRunner/Reverb has developed a method for reducing noise by aggregating and cleaning the resulting triples, in a linguistic and statistical manner. However, they do not address the issue of semantics, for example, there are no synonyms or type constraints provided. We aim for an approch which provides semantics.

## 3 Overview of our Approach

Our main idea is that *semantic types* are crucial for discovering an open set of relations of high quality and for recognizing out-of-knowledge-base entities. Semantic types or classes are available in many knowledge bases. For example, Wikipedia assigns entities to categories (e.g., Jeff Dean is in the categories “living people, rock musicians, . . . ”), YAGO

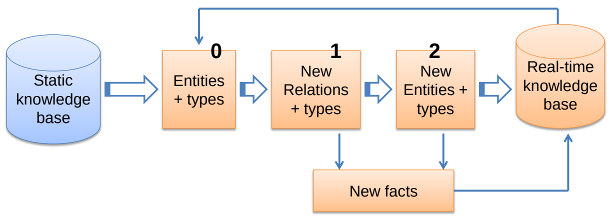


Figure 2: Architectural overview of our approach

(Suchanek 2007) assigns entities to Wordnet classes, Freebase (Bollacker 2008) assigns entities to its own set of categories.

Starting with a static knowledge base consisting of entities and their semantic types, we can generate relations in the form of phrases associated with type signatures. For example, we could generate a relation ⟨*actor*⟩ *'s character in* ⟨*movie*⟩. The types indicate the kinds of entities that can stand in the relation expressed by the phrase. Having typed phrases helps to remove many noisy facts by automatically disqualifying entities whose types do not agree with phrase types. Additionally, for out-of-knowledge-base entities we can leverage the phrases they cooccur with to infer types for new entities. In the simplest case, we can assume that for all pairs of entities X and Y occurring with the phrase "'s character in", X is an actor and Y is a movie. We elaborate later on the limitations of this assumption. Figure 2 illustrates the overall approach.

## 4 Open Set of Relations

In order to generate a large comprehensive set of relations, we developed methods for automatically mining relations from text corpora. We define a relation as a pattern which frequently occurs with entities of the same type, this results in *semantically-typed patterns*, example relations are: ⟨*actor*⟩ *'s character in* ⟨*movie*⟩ and ⟨*comedian*⟩ *parodied* ⟨*person*⟩. We say that each pattern has a type signature; for example for the latter, the type signature is: *comedian* × *person*.

Requiring that the patterns have type signatures provides a number of benefits: First we add semantics to our patterns; second we prune noisy patterns; third, we can infer types for previously unseen entities. For further semantics of a our relations, we arrange them into groups of synonymous patterns and into a hierarchy of subsumptions where general patterns subsume more specific ones. For example, the relation ⟨*person*⟩ *met* ⟨*person*⟩ subsumes the relation ⟨*person*⟩ *married* ⟨*person*⟩. A full description of the relation mining algorithm is beyond the scope of this paper.

## 5 Open Set of Entities

Having mined a large collection of semantically-typed patterns, we now discuss the open set of entities. Patterns require that entities have types which satisfy the type signatures. However, if an entity is new, its types are not known at the time of extraction. To prevent missing out on the facts pertaining to new entities, we need to deduce types for new entities. We propose to align new entities along the type signatures of patterns by inferring entity types from the type signatures. One approach would be based on the following hypothesis: For a given pattern such as ⟨*actor*⟩*'s character in* ⟨*movie*⟩, we can conclude that an entity pair $(X, Y)$, occurring with the pattern in text, implies that X and Y are of the types actor and movie, respectively. However, directly inferring the types from the semantically-typed patterns would lead to many false positives due to the following problems:

- **Polysemy of Syntax.** The same lexico-syntactic pattern can have different type signatures. For example, the following are three different patterns: ⟨*singer*⟩ *released* ⟨*album*⟩, ⟨*music_band*⟩ *released* ⟨*album*⟩, ⟨*country*⟩ *released* ⟨*prisoner*⟩. For an entity pair $(X, Y)$ occurring with this pattern, X can be one of three different types (singer, music_band, country) and Y can be one of two different types (album, prisoner).

- **Incorrect Paths between Entities.** Path tracing between entities is a larger limitation which emerges when a pair of entities occurring in the same sentence do not stand in a relation. This arises more prominently in long sentences. Deep linguistic processing such as dependency parsing facilitates correct path finding between entity pairs. However, due to time limitations in a real-time setting, deep linguistic parsing would be a throughput bottleneck. For example, consider the sentence: *Liskov graduated from Stanford and obtained her PhD degree from MIT.*, In this sentence, there is no relationship between Stanford and MIT, however we may erroneously extract: *[Stanford] obtained her PhD degree from [MIT].* If Stanford were an unknown entity and we had a semantically-typed pattern which says people obtain PhDs from institutions, then we would wrongly infer that Stanford is a person.

We propose to jointly tackle the polysemy and incorrect-path limitations. Our approach aims to solve the optimization problem: which types are most likely valid for a new entity X, given that X occurs with patterns associated with different type signatures. The features over which to optimize include: relative frequencies that X occurs in placeholders of different types, specificity of patterns to different types, and type disjointness constraints which state, for example, that a university cannot be a person. This optimization problem can be formulated as an integer linear program.

## 6 Constrained Response Times

Online algorithms have to be responsive and return results within a timeframe that users are willing to tolerate. The extraction process is time-consuming, as we have to perform a range of expensive functions, such as named-entity tagging, entity disambiguation and entity-type inference for new entities. For this reason, we avoid using an online algorithm that performs information extraction at query time. Instead we propose a *continuous background processing model*. This approach processes a stream of in-coming documents. The stream provider can be a Web crawler or RSS feeds. Extraction is performed on the stream of documents one *time-slice (e.g., an hour)* at a time. The time-slice is a time window where data is accumulated before extraction is executed.

**Document filtering.** Within a given time-slice we can process all the documents. However, not all documents contain any meaningful relational facts which express relations in our *open set of relations*. We can therefore filter out documents that are not promising. A natural filtering approach is to build an index on the documents and use the patterns as queries on the index. If none of the queries return a non-empty result for a given document, then we discard the document. Building an index as a filter can speed up overall execution time.

Another dimension for filtering is the topic focus of a given stream. We imagine a customizable stream, whereby the general topic of interest can be picked, much like in news aggregators. For example, consider a stream following music-related news. This setting does not require that we find facts about sports. Because our patterns are typed, we can filter out all documents which do not contain music-specific patterns.

**Execution time optimization.** There is a lot of redundancy on the Web. Therefore, for a given time-slice, we might not need to process all the documents to get high recall. It could be that processing 10% of the documents already gives us 80% fact recall but at a much lower execution time compared to processing all documents. So there is a trade-off between execution time and recall. We would assume that each time-slice has a target recall value $T_{recall}$, $0 < T_{recall} \leq 1$. We can then estimate the number of documents we need to process to achieve the target recall (Ipeirotis 2006).

## 7 Conclusions

In this paper, we discussed ongoing research on timely fact extraction from frequently changing data. We have described a model that uses semantically-typed patterns for relations, infers entity types for new entities from the semantically-typed patterns, and follows a continuous background processing model which removes extraction from query time. We envision a scalable system capable of discovering new entities and relations as they emerge in data under time constraints. We believe our model contains key ingredients toward this goal. We are investigating our model further and developing a system that embodies these ideas.

## References

Enrique Alfonseca, Marius Pasca, Enrique Robledo-Arnuncio: Acquisition of instance attributes via labeled and related instances. SIGIR 2010

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, Oren Etzioni: Open Information Extraction from the Web. IJCAI 2007

Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, Jamie Taylor: Freebase: a collaboratively created graph database for structuring human knowledge. SIGMOD Conference 2008, data at `http://freebase.com`

Michael J. Cafarella, Alon Y. Halevy, Nodira Khoussainova: Data Integration for the Relational Web. PVLDB 2(1): 1090-1101 (2009)

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka Jr., Tom M. Mitchell. Toward an Architecture for Never-Ending Language Learning. AAAI 2010.

Timothy Chklovski, Patrick Pantel: VerbOcean: Mining the Web for Fine-Grained Semantic Verb Relations. EMNLP 2004

N.N. Dalvi, R. Kumar, B. Pang, R. Ramakrishnan, A. Tomkins, P. Bohannon, S. Keerthi, S. Merugu: A web of concepts. PODS 2009

O. Etzioni, M. Banko, S. Soderland, D.S. Weld: Open information extraction from the web. Commun. ACM 51(12), 2008

Anthony Fader, Stephen Soderland, Oren Etzioni: Identifying Relations for Open Information Extraction. EMNLP 2011.

Panagiotis G. Ipeirotis, Eugene Agichtein, Pranay Jain, Luis Gravano To search or to crawl?: towards a query optimizer for text-centric tasks. SIGMOD 2006

Ndapandula Nakashole, Martin Theobald, Gerhard Weikum: Scalable Knowledge Harvesting with High Precision and High Recall. WSDM 2011

Fabian M. Suchanek, Gjergji Kasneci, Gerhard Weikum: YAGO: a Core of Semantic Knowledge. WWW 2007

G. Weikum, G. Kasneci, M. Ramanath, F.M. Suchanek: Database and information-retrieval methods for knowledge discovery. Commun. ACM 52(4), 2009

Limin Yao, Aria Haghighi, Sebastian Riedel, Andrew McCallum: Structured Relation Discovery using Generative Models. EMNLP 2011