Optimizing Sampling-based Entity Resolution over Streaming Documents

Christian Grant and Daisy Zhe Wang
University of Florida

SIAM BSA Workshop 2015
Knowledge Bases are important structure for organizing and categorizing information.
Knowledge Bases are important structure for organizing and categorizing information.
Knowledge Bases are important structure for organizing and categorizing information.
• Many of these knowledge bases and new knowledge bases are bootstrapped using Wikipedia/Freebase.
• Many of these knowledge bases and new knowledge bases are bootstrapped using Wikipedia/Freebase.

• All Wikipedia information is based on facts from (reputable?) web sources.
Many of these knowledge bases and new knowledge bases are bootstrapped using Wikipedia/Freebase.

All Wikipedia information is based on facts from (reputable?) web sources.
Many of these knowledge bases and new knowledge bases are bootstrapped using Wikipedia/Freebase. All Wikipedia information is based on facts from (reputable?) web sources.
Many of these knowledge bases and new knowledge bases are bootstrapped using Wikipedia/Frebase. All Wikipedia information is based on facts from (reputable?) web sources.
The average time between an event and its appearance on Wikipedia is 356 days.
Knowledge Base Acceleration
Knowledge Base Acceleration

NIST TREC created a track that reads in streaming documents and a set of entities and suggests citations for wikipedia entities.
Knowledge Base Acceleration

NIST TREC created a track that reads in streaming documents and a set of entities and suggests citations for wikipedia entities.
Knowledge Base Acceleration

NIST TREC created a track that reads in streaming documents and a set of entities and suggests citations for wikipedia entities.

Challenges:
Knowledge Base Acceleration

NIST TREC created a track that reads in streaming documents and a set of entities and suggests citations for wikipedia entities.

Challenges:

1) A large amount of documents
Knowledge Base Acceleration

NIST TREC created a track that reads in streaming documents and a set of entities and suggests citations for Wikipedia entities.

Challenges:

1) A large amount of documents

2) Ambiguous text
Knowledge Base Acceleration

NIST TREC created a track that reads in streaming documents and a set of entities and suggests citations for wikipedia entities.

Challenges:

1) A large amount of documents
2) Ambiguous text
3) Ambiguous Entities
Knowledge Base Acceleration

NIST TREC created a track that reads in streaming documents and a set of entities and suggests citations for wikipedia entities.

Challenges:

1) A large amount of documents
2) Ambiguous text
3) Ambiguous Entities
4) Finding relevant facts
Example System

- Wikipedia
- Alias Extraction (Wiki API, Wiki text)
- Manual Aliases Extraction (Twitter)
- Name Order Generator
- Training Data
- Chunk Files Index Generator
- StreamItems Index Generator
- Streaming Slot Value Extraction
- High Accuracy Filter
- Streaming Slot Values

Manual Aliases Extraction (TwiEer)
Example System

- **Wikipedia**
  - Manual Aliases Extraction (TwEer)
  - Wiki API, Wiki text
- **Web Corpus**
  - Chunk Files Index Generator
  - StreamItems Index Generator
- **Training Data**
  - Streaming Slot Value Extraction
  - High Accuracy Filter
  - Streaming Slot Values
Entity Resolution

- Entity resolution is the process of identifying and clustering different manifestations (e.g., mentions, noun phrases, named entities) of the same real world object.
Entity Resolution

• Entity resolution is the process of identifying and clustering different manifestations (e.g., mentions, noun phrases, named entities) of the same real world object.

• Difficult because of ambiguity
Entity Resolution

• Entity resolution is the process of identifying and clustering different manifestations (e.g., mentions, noun phrases, named entities) of the same real world object.

• Difficult because of ambiguity

Same Name, Different Person
Entity Resolution

• Entity resolution is the process of identifying and clustering different manifestations (e.g., mentions, noun phrases, named entities) of the same real world object.

• Difficult because of ambiguity

  Same Name, Different Person

  Different Name, Same Person
Entity Resolution

• Entity resolution is the process of identifying and clustering different manifestations (e.g., mentions, noun phrases, named entities) of the same real world object.

• Difficult because of ambiguity

  * Same Name, Different Person
  * Different Name, Same Person
Entity Resolution

• Entity resolution is the process of identifying and clustering different manifestations (e.g., mentions, noun phrases, named entities) of the same real world object.

• Difficult because of ambiguity

  Same Name, Different Person

  Different Name, Same Person
Entity Resolution Model
Entity Resolution Model
\[ \hat{e} = \arg \max_e p(e) \]
\[ = \arg \max_e \sum_{e \in e} \left\{ \sum_{m,n \in e} \psi_{amn} + \sum_{m \in e, n \notin e} \psi_{rmn} \right\} \]
Find the best arrangement.

\[ \hat{e} = \arg \max_e p(e) \]

\[ = \arg \max_e \sum_{e \in e} \left\{ \sum_{m, n \in e} \psi_{a}^{mn} + \sum_{m \in e, n \notin e} \psi_{r}^{mn} \right\} \]
Entity Resolution Model

\[ \hat{e} = \arg \max_e p(e) \]
\[ = \arg \max_e \sum_{e \in e} \left\{ \sum_{m, n \in e} \psi_{mn} + \sum_{m \in e, n \notin e} \psi_{mn} \right\} \]

Find the best arrangement.
\[ \hat{e} = \underset{e}{\arg\max} \ p(e) \]
\[ = \underset{e}{\arg\max} \ \sum_{e} \left\{ \sum_{m,n \in e} \psi^{mn}_{a} + \sum_{m \in e, n \notin e} \psi^{mn}_{r} \right\} \]
Entity Resolution Algorithm

The Baseline ER metropolis hastings takes a random mention and adds it to a random entity.
The Baseline ER metropolis hastings takes a random mention and adds it to a random entity.
Entity Resolution Algorithm
Entity Resolution Algorithm

1. Select a source mention at \textit{random}.
1. Select a source mention at *random*.
Entity Resolution Algorithm

1. Select a source mention at random.
2. Select a destination mention at random.
Entity Resolution Algorithm

1. Select a source mention at *random*.
2. Select a destination mention at *random*. 

Diagram of entity resolution algorithm.
Entity Resolution Algorithm

1. Select a source mention at random.
2. Select a destination mention at random.
3. Propose a merge.
Entity Resolution Algorithm

1. Select a source mention at random.
2. Select a destination mention at random.
3. Propose a merge.

Reject!
1. Select a source mention at random.
2. Select a destination mention at random.
3. Propose a merge.

Entity Resolution Algorithm

\[ \alpha(e, e') = \min \left( 1, \frac{p(e')}{p(e)} \right) \]
1. Select a source mention at random.
2. Select a destination mention at random.
3. Propose a merge.
1. Select a source mention at *random*.
2. Select a destination mention at *random*.
3. Propose a merge.
Entity Resolution Algorithm

1. Select a source mention at *random.*
2. Select a destination mention at *random.*
3. Propose a merge.
Entity Resolution Algorithm

1. Select a source mention at *random*.
2. Select a destination mention at *random*.
3. Propose a merge.
1. Select a source mention at random.
2. Select a destination mention at random.
3. Propose a merge.
Entity Resolution Algorithm

1. Select a source mention at *random*.
2. Select a destination mention at *random*.
3. Propose a merge.
1. Select a source mention at *random*.
2. Select a destination mention at *random*.
3. Propose a merge.
Entity Resolution Algorithm

1. Select a source mention at *random*.
2. Select a destination mention at *random*.
3. Propose a merge.
1. Select a source mention at random.
2. Select a destination mention at random.
3. Propose a merge.
1. Select a source mention at *random*.
2. Select a destination mention at *random*.
3. Propose a merge.
Entity Resolution Algorithm

1. Select a source mention at *random*.
2. Select a destination mention at *random*.
3. Propose a merge.
1. Select a source mention at *random*.
2. Select a destination mention at *random*.
3. Propose a merge.
1. Select a source mention at *random*.
2. Select a destination mention at *random*.
3. Propose a merge.
4. Accept when it improves the state.
Entity Resolution Algorithm

1. Select a source mention at *random*.
2. Select a destination mention at *random*.
3. Propose a merge.
4. Accept when it improves the state.

Accept!
Entity Resolution Algorithm

Eventually **converges**. (State does not oscillate or vary)
Entity Resolution Algorithm

Eventually **converges**. (State does not oscillate or vary)
Sampling Optimizations

Distributed Computations (Singh et al. 2011)
Sampling Optimizations

Distributed Computations (Singh et al. 2011)
Query-Driven Computation (Grant et al. 2015)
Sampling Inefficiencies
Sampling Inefficiencies

1. Large clusters are the slowest.
Sampling Inefficiencies

1. Large clusters are the slowest.
   - Pairwise comparisons are expensive.
Sampling Inefficiencies

1. Large clusters are the slowest.
   • Pairwise comparisons are expensive. \( \Theta(n^2) \)
Sampling Inefficiencies

1. Large clusters are the slowest.
   
   Pairwise comparisons are expensive. $\Theta(n^2)$

2. Excessive computation on unambiguous entities
Sampling Inefficiencies

1. Large clusters are the slowest.
   Pairwise comparisons are expensive. $\Theta(n^2)$

2. Excessive computation on unambiguous entities
   Entities such as Carnegie Mellon are relatively unambiguous.
1. Large clusters are the slowest.
   Pairwise comparisons are expensive. $\Theta(n^2)$

2. Excessive computation on unambiguous entities
   Entities such as *Carnegie Mellon* are relatively unambiguous.

Streaming documents exacerbates these problems.
Optimizer for MCMC Sampling

Database style optimizer for streaming MCMC.
Optimizer for MCMC Sampling

Database style optimizer for streaming MCMC.

This optimizer makes two decisions:
Optimizer for MCMC Sampling

Database style optimizer for streaming MCMC.

This optimizer makes two decisions:

1. Can I approximate the state score calculation?
Optimizer for MCMC Sampling

Database style optimizer for streaming MCMC.

This optimizer makes two decisions:

1. Can I approximate the state score calculation?

2. Should I compress an Entity?
Experiments

- **Wikilink Data Set** *(Singh, Subramaniya, Pereira, McCallum, 2011)*
  - Largest fully-labeled data set
  - 40 Million Mentions
  - 180 GBs of data

![Figure 1: Links to Wikipedia as Entity Labels](http://en.wikipedia.org/wiki/Banksy)
Large Entity Sizes
Entity Compression
Entity Compression

• Known matches can be compressed into a representative mention.
Entity Compression

- Known matches can be compressed into a representative mention.
- Entity compression can reduce the number of mentions ($n$).
Entity Compression

• Known matches can be compressed into a representative mention.

• Entity compression can reduce the number of mentions ($n$).

• Compression of large and popular entities is costly.
Entity Compression

- Known matches can be compressed into a representative mention.
- Entity compression can reduce the number of mentions ($n$).
- Compression of large and popular entities is costly.
- Compression errors are permanent.
Compression Types

- Run-Length Encoding
- Hierarchical Compression (Wick et al.)
Early Stopping

• Can we estimate the computation of the features?
Early Stopping

• Can we estimate the computation of the features?

• Given a $p$ value, randomly select less values.
Early Stopping

- Can we estimate the computation of the features?
- Given a $p$ value, randomly select less values.

Singh et al. EMNLP’12
Optimizer

Current work

1. Classifier for deciding when to perform *early stopping*.

2. Classifier for the decision to *compress*. 
When should it compress?
When should it compress?

Power law says there are only a small number of very large clusters.
When should it compress?

Power law says there are only a small number of very large clusters.

We can treat these in a special way.
When should it compress?

Power law says there are only a small number of very large clusters.

We can treat these in a special way.

Examining the Wiki Link data set.
When should it compress?

Power law says there are only a small number of very large clusters.

We can treat these in a special way.

Examining the Wiki Link data set.
When should it compress?

Power law says there are only a small number of very large clusters.

We can treat these in a special way.

Examining the Wiki Link data set.

Exact String Match Initialization
When should it compress?

Power law says there are only a small number of very large clusters.

We can treat these in a special way.

Examining the Wiki Link data set.
When should it compress?

Power law says there are only a small number of very large clusters.

We can treat these in a special way.

Examining the Wiki Link data set.

Exact String Match Initialization

Ground Truth
When should it compress?

Power law says there are only a small number of very large clusters.

We can treat these in a special way.

Examining the Wiki Link data set.

Exact String Match Initialization

Ground Truth
When should it compress?

We could make 100,000 insertions in the time it take to compress a 300K mention cluster.
When should it compress?

We could make 100,000 insertions in the time it takes to compress a 300K mention cluster.

Compression must be worth it.
When should it compress?

We could make 100,000 insertions in the time it take to to compress a 300K mention cluster.

Compression must be worth it.
When should we approximate?
When should we approximate?

- Early stopping only makes sense for clusters of medium size.
- It is better to do full comparison for small and large cluster sizes.
When should we approximate?

• Early stopping only makes sense for clusters of medium size.

• It is better to do full comparison for small and large cluster sizes.
When should we approximate?

- Early stopping only makes sense for clusters of medium size.
- It is better to do full comparison for small and large cluster sizes.
Optimizer for Query-Driven Sampling

Optimizer needs to know:
• Current Cardinality of Items in each entity.
• Memory/CPU configuration for estimating baseline time

while samples-- > 0:
    m ~ Mentions
    e ~ Entities
    state’ = move(state, m, e)
    o = Optimize(state, state’, m, e)
    if (!score(state’, state, o)):
        state = state’
    doCompress(state, m, e, o)
Summary

• We motivated the need and discussed the open space for optimization of MCMC sampling methods.

• We plan to use the newly released labeled TREC stream corpus.

• Want to collaborate?! 

• Lets talk if you want to do a Ph.D. at the University of Oklahoma!
Thank you!