Sampling Techniques for Streaming Cross Document Coreference Resolution

Luke Shrimpton* **Victor Lavrenko†** **Miles Osborne§**
* School of Informatics, University of Edinburgh: luke.shrimpton@ed.ac.uk
† School of Informatics, University of Edinburgh: vlavrenk@inf.ed.ac.uk
§ Bloomberg, London: mosborne29@bloomberg.net

Abstract

We present the first truly streaming cross document coreference resolution (CDC) system. Processing infinite streams of mentions forces us to use a constant amount of memory and so we maintain a representative, fixed sized sample at all times. For the sample to be representative it should represent a large number of entities whilst taking into account both temporal recency and distant references. We introduce new sampling techniques that take into account a notion of streaming discourse (current mentions depend on previous mentions). Using the proposed sampling techniques we are able to get a CEAFe score within 5% of a non-streaming system while using only 30% of the memory.

1 Introduction

Cross document coreference resolution (CDC) - identifying mentions that refer to the same entity across documents - is a prerequisite when combining entity specific information from multiple documents. Typically large scale CDC involves applying a scalable clustering algorithm to all the mentions. We consider *streaming* CDC, hence our system must conform to the streaming computational resource model (Muthukrishnan, 2005). Each mention is processed in bounded time and only a constant amount of memory is used. Honoring these constraints ensures our system can be applied to infinite streams such as newswire or social media.

Storing all the mentions in memory is clearly infeasible, hence we need to either compress mentions or store a *sample*. Compression is more computationally expensive as it involves merging/forgetting mention components (for example: components of a vector) whereas sampling decides to store or forget whole mentions. We investigate sampling techniques due to their computational efficiency. We explore which mentions should be stored while performing streaming CDC. A sample should represent a diverse set of entities while taking into account both temporal recency and distant mentions. We show that using a notion of streaming discourse, where what is currently being mentioned depends on what was previously mentioned significantly improves performance on a new CDC annotated Twitter corpus.

2 Related Work

There are many existing approaches to CDC (Bagga and Baldwin, 1998; Lee et al., 2012; Andrews et al., 2014). Few of them scale to large datasets. Singh et al. (2011) proposed a distributed hierarchical factor graph approach. While it can process large datasets, the scalability comes from distributing the problem. Wick et al. (2012) proposed a similar approach based on compressing mentions, while scalable it does not conform to the streaming resource model. The only prior work that addressed online/streaming CDC (Rao et al., 2010) was also not constrained to the streaming model. None of these approaches operate over an unbounded stream processing mentions in constant time/memory.

Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL, pages 1391–1396,
Denver, Colorado, May 31 – June 5, 2015.

3 Entities in Streams

Streams like Twitter are well known as being real-time and highly bursty. Some entities are continually mentioned throughout the stream (eg: President Obama) whereas others burst, suddenly peak in popularity then decay (eg: Phillip Hughes, a cricketer who died following a bowling injury).

Capturing the information required to perform streaming CDC in constant space requires us to *sample* from the stream. For example we may consider only the most recent information (eg: the previous 24 hours worth of mentions). This may not be ideal as it would result in a sample biased towards bursting entities, neglecting the continually mentioned entities. We propose three properties that are important when sampling mentions from a stream of tweets:

- **Recency:** There should be some bias towards recent mentions to take into account the real-time nature of the stream. The set of entities mentioned one day is likely to be similar to the set of entities mentioned on the following day.
- **Distant Reference:** The temporal gap between mentions of the same entity can vary drastically, recency captures the small gaps though to capture the larger gaps older mentions should be stored. A mention should be correctly resolved if the last mention of the entity was either a day or a week ago.
- **Entity Diversity:** The sample should contain mentions of many entities instead of storing lots of mentions about a few entities. If the sample only contains mentions of the most tweeted about entity (the one that is bursting) it is impossible to resolve references to other entities.

These properties suggest we should take into account a notion of streaming discourse when sampling: mentions sampled should depend on the previous mentions (informed sampling).

4 Approach

We implemented a representative pairwise streaming CDC system using single link clustering. Mention similarity is a linear combination of mention text and contextual similarity (weighted 0.8 and 0.2 respectively) similar to Rao et al. (2010). Mention text similarity is measured using cosine similarity of character skip bigram indicator vectors and contextual similarity is measured using tf-idf weighted cosine similarity of tweet terms. The stream is processed sequentially: we resolve each mention by finding its nearest neighbor in the sample, linking the two mentions if the similarity is above the linking threshold.

5 Sampling Techniques

The sampling techniques we investigate are summarized below. Each technique has an insertion and removal policy that are followed each time a mention is processed. New sampling techniques are indicated by a star (*). The new sampling techniques require the nearest neighbor to be identified. As this is already computed during resolution hence the overhead of these new techniques is very low. Parameters particular to each sampling technique are noted in square brackets and are set using a standard grid search on a training dataset.

- **Exact:** To provide an upper bound on performance we forgo the constraints of the streaming resource model and store all previously seen mentions:

  Insertion: Add current mention to sample. **Removal:** Do nothing.

- **Window:** We sample a moving window of the most recent mentions (first in, first out). For example this technique assumes that if we are processing mentions on Monday with a window of approximately 24 hours all relevant entities were mentioned since Sunday.

  Insertion: Add current mention to sample. **Removal:** Remove oldest mention.

- **Uniform Reservoir Sampling (Uniform-R):** A uniform sample of previously seen mentions will capture a diverse set of entities from the entire stream - taking into account diversity and distant references. This can be achieved using a reservoir sample (Vitter, 1985). We assume each previously seen mention is equally likely to help resolve the current mention.

Insertion: Add current mention with probability p_i. **Removal:** If a mention was inserted choose a mention uniformly at random to remove.

Setting $p_i = k/N$ where k is the sample size and N is the number of items seen ensures the sample is uniform (Vitter, 1985).

- **Biased Reservoir Sampling (Biased-R):** To resolve both distant and recent references we should store recent mentions and some older mentions. An uninformed approach from randomized algorithms is to use an exponentially biased reservoir sample (Aggarwal, 2006; Osborne et al., 2014). It will store mostly recent mentions but will probabilistically allow older mentions to stay in the sample. For example this technique will sample lots of mentions from yesterday and less from the day before yesterday.

  Insertion: Add current mention with probability p_i **Removal:** If a mention was inserted choose a mention uniformly at random to remove.

  Unlike uniform reservoir sampling p_i is constant. A higher value puts more emphasis on the recent past. [p_i]

- **Cache*:** We should keep past mentions critical to resolving current references (an informed implementation of recency) and allow mentions to stay in the sample for an arbitrary period of time to help resolve distance references. For example if the same mention is used to resolve a reference on Saturday and Sunday it should be in the sample on Monday.

  Insertion: Add current mention to sample. **Removal:** Choose a mention that was not recently used to resolve a reference uniformly at random and remove it.

  When a mention is resolved we find its most similar mention in the sample, recording its use in a first in, first out cache of size n. The mention to be removed is chosen from the set of mentions not in the cache. We set n equal to a proportion of the sample size. [Proportion of mentions to keep]

- **Diversity*:** If we store fewer mentions about each distinct entity we can represent more entities in the sample. For example if news breaks that a famous person died yesterday the sample should not be full of mentions about that entity at the expense of other entities mentioned today.

  Insertion: Add current mention to sample. **Removal:** If there is a sufficiently similar mention in the sample remove it else choose uniformly at random to be removed.

  We remove the past mention most similar to the current mention, but only if the similarity exceeds a threshold. [Replacement Threshold]

- **Diversity-Cache (D-C)*:** We combine Diversity and Cache sampling.

  Insertion: Add current mention to sample. **Removal:** If there is a sufficiently similar mention in the sample remove it else remove a mention that has not recently been used to resolve a reference chosen uniformly at random.

  For this technique we first choose the replacement threshold then the proportion of mentions to keep. [Replacement threshold and proportion of mentions to keep]

6 Dataset

We collected 52 million English tweets from the 1% sample of all tweets sent over a 77 day period. We performed named entity recognition using Ritter et al. (2011). It is clearly infeasible for us to annotate all the mentions in the dataset. Hence we annotated a sample of the entities. As with most prior work we focused on person named entity mentions (of which there is approximately 6 million in the dataset).

To select the entities we first sampled two names based on how frequently they occur in the dataset: 'Roger' was chosen randomly from the low frequency names (between 1,000 and 10,000 occurrences) and 'Jessica' was chosen similarly from medium frequency names (10,000 to 100,0000 occurrences). We first annotated all mentions of the names 'Roger' and 'Jessica' discarding entities mentioned once. For the remaining entities we annotated all their mentions (not restricting to mentions that contained the words 'Roger' or 'Jessica').

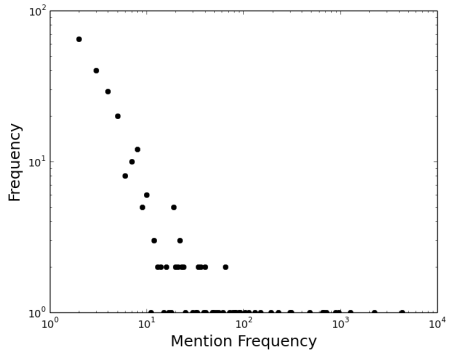

Figure 1: Mention Frequency Distribution.

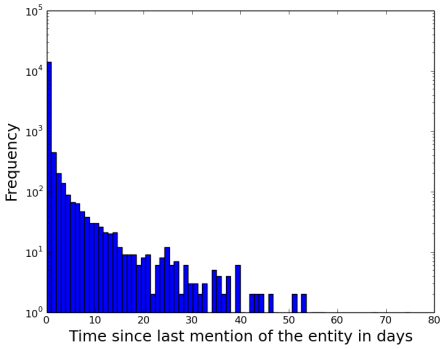

Figure 2: Distribution of time since previous mention of the same entity (gap). Each bar represents 24 hours.

This covers a diverse selection of people including: 'Roger Federer' (tennis player) and 'Jessie J' (the singer whose real name is 'Jessica Cornish') as well as less popular entities such as porn stars and journalists[1].

Some statistics of the dataset are summarized in table 1. We also plot the mention frequency (how often each entity was mentioned) distribution in figure 1 which shows a clear power law distribution similar to what Rao et al. (2010) reported on the New York Times annotated corpus. We show that recency and distant reference are important aspects by plotting the time since previous mention of the same entity (gap) for each mention in figure 2. The gap is often less than 24 hours demonstrating the importance of recency. There are also plenty of mentions with much larger gaps, demonstrating the need to be able to resolve distant references.

Source	Mentions	Entities	Wiki Page Exists
Roger	5,794	137	69%
Jessica	10,543	129	46%
All	16,337	266	58%

Table 1: Mention/entity counts and percentage of entities that have a Wikipedia page.

7 Experiments

As we are processing a stream we use a rolling evaluation protocol. Our corpus is split up into 11 constant sized temporally adjacent blocks each lasting approximately one week. Parameters are set using a standard grid search on one block then we progress to the next block to evaluate. The first block is reserved for setting the linking threshold prior to our rolling evaluation. We report the average over the remaining blocks. For all sampling techniques that have a randomized component we report an average over 10 runs.

As the sample size will have a large effect on performance we evaluate using various sample sizes. We base our sample size on the average amount of mentions per day (78,450) and evaluate our system with sample sizes of 0.25,0.5,1,2 times the average amount of mentions per day.

We evaluate using CEAFe (Luo, 2005), it is the only coreference evaluation metric that can be trivially adapted to datasets with a sample of annotated entities. With no adaption it measures how well a small amount of entities align with a system output over a large amount. To make the evaluation more representative we only use response clusters that contain an annotated mention. This scales precision and maintains interpretability. We determine if observed differences are significant by using a Wilcoxon signed-rank test with a p value of 5% over the 9 testing points. Results are shown in table 2.

- **Window:** This shows the performance that can be achieved by only considering recency.
- **Uniform Reservoir Sampling (Uniform-R):** This shows the performance achieved by using an uninformed technique to store a diverse set of older mentions.

[1]The annotations, including links to Wikipedia pages when available, can be downloaded from `https://sites.google.com/site/lukeshr/`.

Sample Size	Sampling Technique	CEAFe P	CEAFe R	CEAFe F
0.25 Days	Window	24.2	67.2	35.6
	Uniform-R	23.0	67.2	34.3
	Biased-R	24.8	67.8	36.3
19,613	Cache ∗	25.5	67.2	37.0
Mentions	Diversity ∗	27.6	69.2	39.4
	D-C ∗ †	**30.1**	**69.7**	**42.0**
0.5 Days	Window	31.3	69.4	43.1
	Uniform-R	29.9	69.1	41.7
	Biased-R	31.6	69.9	43.5
39,225	Cache ∗	32.9	69.7	44.7
Mentions	Diversity ∗	37.5	71.6	49.2
	D-C ∗ †	**40.1**	**72.3**	**51.6**
1.0 Days	Window	40.7	72.0	52.0
	Uniform-R	39.3	71.4	50.6
	Biased-R	40.9	72.3	52.3
78,450	Cache ∗	42.0	72.0	53.1
Mentions	Diversity ∗	48.5	74.3	58.7
	D-C ∗ †	**49.8**	**74.5**	**59.7**
2.0 Days	Window	50.2	74.1	59.8
	Uniform-R	49.2	73.7	58.9
	Biased-R	50.3	74.1	59.9
156,900	Cache ∗	50.9	74.1	60.4
Mentions	Diversity ∗	55.2	75.2	63.7
	D-C ∗	**55.5**	**75.3**	**63.9**
≈600,000 Mentions	Exact	59.7	75.4	66.6

Table 2: CEAFe performance for various sample sizes and sampling techniques. ∗ indicates significant improvement over Window sampling. † indicates significant improvement over Diversity sampling

- **Biased Reservoir Sampling (Biased-R):** Uninformed sampling of older mentions is not sufficient to significantly improve performance.

- **Cache:** By using an informed model of recency we keep mentions critical to resolving references currently being tweeted resulting in a significant performance improvement.

- **Diversity:** By using an informed technique to increase the amount of distinct entities represented in the sample we significantly improve performance.

- **Diversity-Cache (D-C):** By combining the new sampling techniques we significantly improve performance. Once we have increased the amount of entities represented in the sample we are still able to benefit from an informed model of recency.

Using uninformed sampling techniques (reservoir sampling) does not result in a significant performance improvement over Window sampling, only informed sampling techniques show a significant improvement. As the sample size increases the performance difference decreases. With larger samples there is space to represent more entities and it is less likely to remove a useful mention at random.

8 Conclusion

We presented the first truly streaming CDC system, showing that significantly better performance is achieved by using an informed sampling technique that takes into account a notion of streaming discourse. We are able to get to within 5% of an exact system's performance while using only 30% of the memory required. Instead of improving performance by using an uninformed sampling technique and doubling the memory available, similar performance can be achieved by using the same amount of memory and a informed sampling technique. Further work could look at improving the similarity metric used, applying these sampling techniques to other streaming problems or adding a mention compression component.

References

Charu C Aggarwal. 2006. On biased reservoir sampling in the presence of stream evolution. In *Proceedings of the 32nd international conference on Very large data bases*, pages 607–618. VLDB Endowment.

Nicholas Andrews, Jason Eisner, and Mark Dredze. 2014. Robust entity clustering via phylogenetic inference. In *Association for Computational Linguistics (ACL)*.

Amit Bagga and Breck Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 79–85. Association for Computational Linguistics.

Heeyoung Lee, Marta Recasens, Angel Chang, Mihai Surdeanu, and Dan Jurafsky. 2012. Joint entity and

event coreference resolution across documents. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 489–500. Association for Computational Linguistics.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*.

S Muthukrishnan. 2005. *Data streams: Algorithms and applications*. Now Publishers Inc.

Miles Osborne, Ashwin Lall, and Benjamin Van Durme. 2014. Exponential reservoir sampling for streaming language models. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 687–692. Association for Computational Linguistics.

Saša Petrović, Miles Osborne, and Victor Lavrenko. 2010. Streaming first story detection with application to twitter. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 181–189. Association for Computational Linguistics.

Delip Rao, Paul McNamee, and Mark Dredze. 2010. Streaming cross document entity coreference resolution. In *Coling 2010: Posters*, pages 1050–1058. Coling 2010 Organizing Committee.

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named entity recognition in tweets: An experimental study. In *EMNLP*.

Sameer Singh, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum. 2011. Large-scale cross-document coreference using distributed inference and hierarchical models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 793–803. Association for Computational Linguistics.

Jeffrey S Vitter. 1985. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 11(1):37–57.

Michael Wick, Sameer Singh, and Andrew McCallum. 2012. A discriminative hierarchical model for fast coreference at large scale. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 379–388. Association for Computational Linguistics.