Cross-Document Co-Reference Resolution using Sample-Based Clustering with Knowledge Enrichment

Sourav Dutta
Max Planck Institute for Informatics
Saarbrücken, Germany
sdutta@mpi-inf.mpg.de

Gerhard Weikum
Max Planck Institute for Informatics
Saarbrücken, Germany
weikum@mpi-inf.mpg.de

Abstract

Identifying and linking named entities across information sources is the basis of knowledge acquisition and at the heart of Web search, recommendations, and analytics. An important problem in this context is cross-document co-reference resolution (CCR): computing equivalence classes of textual mentions denoting the same entity, within and across documents. Prior methods employ ranking, clustering, or probabilistic graphical models using syntactic features and distant features from knowledge bases. However, these methods exhibit limitations regarding run-time and robustness.

This paper presents the $CROCS$ framework for unsupervised CCR, improving the state of the art in two ways. First, we extend the way knowledge bases are harnessed, by constructing a notion of *semantic summaries* for intra-document co-reference chains using co-occurring entity mentions belonging to different chains. Second, we reduce the computational cost by a new algorithm that embeds sample-based bisection, using spectral clustering or graph partitioning, in a hierarchical clustering process. This allows scaling up CCR to large corpora. Experiments with three datasets show significant gains in output quality, compared to the best prior methods, and the run-time efficiency of CROCS.

1 Introduction

1.1 Motivation and Problem Statement

We are witnessing another revolution in Web search, user recommendations, and data analytics: transitioning from documents and keywords to data, knowledge, and entities. Examples of this megatrend are the Google Knowledge Graph and its applications, and the IBM Watson technology for deep question answering. To a large extent, these advances have been enabled by the construction of huge knowledge bases (KB's) such as DBpedia, Yago, or Freebase; the latter forming the core of the Knowledge Graph. Such semantic resources provide huge collections of *entities*: people, places, companies, celebrities, movies, etc., along with rich knowledge about their properties and relationships.

Perhaps the most important value-adding component in this setting is the recognition and disambiguation of named entities in Web and user contents. *Named Entity Disambiguation* (NED) (see, e.g., (Cucerzan, 2007; Milne & Witten, 2008; Cornolti et al., 2013)) maps a mention string (e.g., a person name like "Bolt" or a noun phrase like "lightning bolt") onto its proper entity if present in a KB (e.g., the sprinter Usain Bolt).

A related but different task of *co-reference resolution (CR)* (see, e.g., (Haghighi & Klein, 2009; Ng, 2010; Lee et al., 2013)) identifies all mentions in a given text that refer to the same entity, including anaphoras such as "the president's wife", "the first lady", or "she". This task when extended to process an entire corpus is then known as *cross-document co-reference resolution (CCR)* (Singh et al., 2011). It takes as input a set of documents with entity mentions, and computes as output a set of equivalence classes over the entity mentions. This does not involve mapping mentions to the entities of a KB. Unlike NED, CCR can deal with long-tail or emerging entities that are not captured in the KB or are merely in very sparse form.

State of the Art and its Limitations. CR methods, for co-references within a document, are generally based on rules or supervised learning using differ-

Transactions of the Association for Computational Linguistics, vol. 3, pp. 15–28, 2015. Action Editor: Hwee Tou Ng.
Submission batch: 9/2014; Revision batch 11/2014; Published 1/2015.

ent kinds of linguistic features like syntactic paths between mentions, the distances between them, and their semantic compatibility as derived from co-occurrences in news and Web corpora (Haghighi & Klein, 2009; Lee et al., 2013). Some methods additionally use distant labels from knowledge bases (KB's). Cluster-ranking and multi-sieve methods incrementally expand groups of mentions and exploit relatedness features derived from semantic types, alias names, and Wikipedia categories (Rahman & Ng, 2011a; Ratinov & Roth, 2012).

The CCR task - computing equivalence classes across documents - is essentially a clustering problem using a similarity metric between mentions with features like those discussed above. However, standard clustering (e.g., k-means or EM variants, CLUTO, etc.) lacks awareness of the transitivity of co-reference equivalence classes and suffers from knowledge requirement of model dimensions. Probabilistic graphical models like Markov Logic networks (Richardson & Domingos, 2006; Domingos et al., 2007; Domingos & Lowd, 2009) or factor graphs (Loeliger, 2008; Koller & Friedman, 2009) take into consideration constraints such as transitivity, while spectral clustering methods (Luxburg, 2007) implicitly consider transitivity in the underlying eigenspace decomposition, but suffer from high computational complexity. In particular, all methods need to precompute features for the data points and similarity values between all pairs of data points. The latter may be alleviated by pruning heuristics, but only at the risk of degrading output quality.

Note that CCR cannot be addressed by simply applying local CR to a "super-document" that concatenates all documents in the corpus. Within a document, identical mentions typically refer to the same entity, while in different documents, identical mentions can have different meanings. Although a cross-document view gives the opportunity to spot joint cues from different contexts for an entity, documents vary in their styles of referring to entities and merely combining the local co-reference chains into a super-group might lead to substantial noise introduction. In addition, CR methods are not designed for scaling to huge "super-documents" corresponding to millions of web pages or news articles.

Problem Statement. We aim to overcome the above limitations by proposing a CCR method that makes rich use of distant KB features, considers transitivity, and is computationally efficient.

1.2 Approach and Contribution

In this paper, we efficiently tackle the CCR problem by considering co-occurring mentions and rich features from external knowledge bases, and using a transitivity-aware sampling-based hierarchical clustering approach. We developed the $CROCS$ (*CRO*ss-document *C*o-reference re*S*olution) framework with *unsupervised hierarchical clustering* by repeated bisection using spectral clustering or graph partitioning. $CROCS$ harnesses semantic features derived from KB's by constructing a notion of *semantic summaries* (*semsum's*) for the intra-document co-reference chains. In addition to incorporating KB labels as features for the co-referring mentions, we also consider co-occurring mentions belonging to other entities and utilize their features. Consider the text: `Hillary lived in the White House and backed Bill despite his affairs.` containing 3 mention groups: {"Hillary"}, {"Bill"}, and {"White House"}. Merely obtaining distant KB features for the first mention group, the sparse information leads to high ambiguity, e.g., may refer to the mountaineer Sir Edmund Hillary. But by also obtaining features from KB for "White House" (co-occurring mention), we obtain much stronger cues towards the correct solution.

CROCS adopts a bisection based clustering method and invokes it repeatedly in a top-down hierarchical procedure with an information-theoretic stopping criterion for cluster splitting. We escape the quadratic run-time complexity for pair-wise similarity computations by using a *sampling technique* for the spectral eigenspace decomposition or for graph partitioning. This is inspired by the recent work of (Krishnamurty et al., 2012; Wauthier et al., 2012) on active clustering techniques. Similarity computations between mention groups are performed lazily on-demand for the dynamically selected samples.

In a nutshell, the novel contributions are:

- $CROCS$, a framework for cross-document co-reference resolution using sample-based spectral clustering or graph partitioning embedded in a hierarchical bisection process;
- *semsum's*, a method for incorporating distant features from KB's also considering the coupling between co-occurring mentions in different co-reference chains;

- experimental evaluation with benchmark corpora demonstrating substantial gains over prior methods in accuracy and run-time.

2 Computational Framework

The CROCS model assumes an input set of text documents $D = \{d_1, d_2, \dots\}$, with markup of entity mentions $M = \{m_{11}, m_{12}, \dots, m_{21}, m_{22}, \dots\}$, $m_{ij} \in d_j$, present in the documents. CROCS computes an equivalence relation over M with equivalence classes C_j, where $C_j \cap C_k = \emptyset$ for $j \neq k$ and $\cup_j C_j = M$. The number of desired classes is apriori unknown; it needs to be determined by the algorithm. Detecting the mentions and marking their boundaries within the text is a problem by itself, referred to as NER (Named Entity Recognition). This paper does not address this issue and relies on established methods.

The CROCS framework consists of 4 stages:

1. **Intra-document CR**: Given an input corpus, D with mentions M, we initially perform *intra*-document co-reference resolution.
2. **Knowledge enrichment:** For each of the local mention groups ($\{m_{ij}\}$) obtained in the previous step, we combine the sentences of the mentions to determine the best matching entity in a KB and retrieve its features. Analogous steps are performed for co-occurring mentions (of $\{m_{ij}\}$) and their features included. We term this feature set of $\{m_{ij}\}$ as *semantic summary* (*semsum's*).
3. **Similarity computation:** We compute similarity scores between mention groups based on the features extracted above. These are computed on-demand, and only for a sampled subset of mentions (avoiding quadratic computation cost).
4. **Sampling-based clustering:** We perform spectral clustering or balanced graph partitioning (using the similarity metric) in a hierarchical fashion to compute the *cross*-document co-reference equivalence classes of mentions.

3 Intra-Document CR

$CROCS$ initially pre-processes input documents to cast them into plain text (using standard tools like (`https://code.google.com/p/boilerpipe/`), (`www.jsoup.org`), etc.). It then uses the Stanford CoreNLP tool suite to detect mentions and anaphors (`http://nlp.stanford.edu/software/`). The detected mentions are also tagged with coarse-grained lexical types (person, organization, location, etc.) by the Stanford NER Tagger (Finkel et al., 2005). This forms the input to the intra-document CR step, where we use the state-of-the-art open-source CR tool (based on multi-pass sieve algorithm) from Stanford to compute the local mention co-reference chains (Raghunathan et al., 2010; Lee et al., 2011; Lee et al., 2013). The tagged texts and the local co-reference chains are then passed to the second stage.

This local CR step may produce errors (e.g., incorrect chaining of mentions or omissions) which propagate to the later stages. However, improving intra-document CR is orthogonal to our problem and thus out of the scope of this paper. Our experiments later show that CROCS is robust and produces high-quality output even with moderate errors encountered during the local-CR stage.

4 Knowledge Enrichment

The *knowledge enrichment* phase starts with the local co-reference chains per document. Assume that we have obtained mention groups (chains) {Michelle, she, first lady} and {the president's wife, first lady} from two documents. To assess whether these two chains should be combined, i.e., they both refer to the same entity, we compute semantic features by tapping into knowledge bases (KB's). Specifically, we harness labels and properties from `freebase.com` entries, for possibly matching entities, to enrich the features of a mention group. The KB features form a part of the *semantic summary* or *semsum's* for each local mention group. Features derived from the constructed semsum's are later used to compare different mention groups via a similarity measure (described in Section 5).

Formally, a mention m is a text string at a particular position in a document. m belongs to a mention group $M(m)$ consisting of all equivalent mentions, with the same string (at different positions) or different strings. For a given m, the *basic semsum* of m, $S_{basic}(m)$, is defined as

$$S_{basic}(m) = \{t \in sentence(m') | m' \in M(m)\} \cup \{t \in label(m') | m' \in M(m)\}$$

where t are text tokens (words or phrases), $sentence(m')$ is the sentence in which mention m' occurs, and $label(m')$ is the semantic label for m' obtained from the KB. Note that $S_{basic}(m)$ is a bag

of tokens, as different mentions in $M(m)$ can obtain the same tokens or labels and there could be multiple occurrences of the same mention string in $M(m)$ anyway.

Prior works on CR (e.g., (Rahman & Ng, 2011a; Ratinov & Roth, 2012; Hajishirzi et al., 2013; Zheng et al., 2013)) and NED (e.g., (Cucerzan, 2007; Milne & Witten, 2008; Ratinov et al., 2011; Hoffart et al., 2011; Hajishirzi et al., 2013)) have considered such form of distant features. CROCS extends these previous methods by also considering distant features for *co-occurring* mention groups, and not just the group at hand. We now introduce a general *framework for knowledge enrichment* in our CCR setting.

Strategies for knowledge enrichment involve decision making along the following dimensions:

- **Target:** items (single mentions, local mention groups, or global mention groups across documents) for which semantic features are obtained.
- **Source:** the resource from where semantic features are extracted. Existing methods consider a variety of choices: i) input corpora, ii) external text corpus, e.g., Wikipedia, and iii) knowledge bases such as Freebase, DBpedia, or Yago.
- **Scope:** the neighborhood of the target considered for enrichment. It can either be restricted to the target itself or can consider co-occurring items (other mention groups connected to the target).
- **Match:** involves mapping the target to one or more relevant items in the source, and can involve simple name queries to full-fledged NED based on relevance or score confidence.

Existing methods generally consider individual mentions or local mention groups as target. Extended scopes like co-occurring entities based on automatic NER and IE techniques have been proposed (Mann & Yarowsky, 2003; Niu et al., 2004; Chen & Martin, 2007; Baron & Freedman, 2008), but use only the input corpus as the enrichment source. Recent methods (Rahman & Ng, 2011a; Ratinov & Roth, 2012; Hajishirzi et al., 2013; Zheng et al., 2013) harness KB's, but consider only local mention groups. Also, these methods rely on high-quality NED for mapping mentions to KB entries. In contrast, $CROCS$ considers extended scopes that include mention groups along with co-occurring mention groups when tapping into KB's. We make only weak assumptions on matching mentions against KB entities, by filtering on confidence and merely treating semsum's as features rather than relying on perfectly mapped entities. Specifically, our $CROCS$ method handles the four dimensions of knowledge enrichment as follows:

Enrichment Target: We use per-document mention groups, after the local CR step, as target. In principle, we could repeat the enrichment during the iterations of the CCR algorithm. However, as CROCS performs top-down splitting of groups rather than bottom-up merging, there is no added value.

Enrichment Source: We include all the sentences of a mention group in its semsum's, thus drawing on the input document itself. The main enrichment harnesses entity-structured KB's like Freebase or Yago by querying them with phrases derived from the mention groups' summaries. The features that are extracted from the best-matching entity include semantic types or categories (e.g., "politician", "award nominee"), alias names (e.g., "Michelle Robinson"), titles (e.g., "First Lady of the United States") and gender of people. These features are appended to the semsum's and form the core of a mention group's semantic summary.

Enrichment Scope: $CROCS$ includes co-occurring mention groups as additional targets for semantic features. Consider the 4 example sentences in Figure 1. Suppose the local CR finds 4 mention groups as shown. The mentions and the sentences in which they occur are represented as a bipartite graph depicting their connections (right side of Fig. 1). Consider the mention group of "president's wife" (m_{11}) and "first lady" (m_{21}). Together with their immediate sentence neighbors in the bipartite graph, these mentions form what we call the *basic scope* for knowledge enrichment, i.e., $\{m_{11}, s_1, m_{21}, s_2\}$.

The sentences of this mention group contain other mentions which can be in mention groups spanning further sentences. We utilize this co-occurrence as additional cues for characterizing the mention group at hand. The union of the current scope with that of all the two-hop neighbors in the bipartite graph form the *extended scope*. For the group $\{m_{11}, s_1, m_{21}, s_2\}$, the two-hop mention neighbors are $\{m_{12}, m_{22}, m_{23}, m_{31}\}$. Hence, we include the scopes of these groups, the mentions and sentences, yielding the extended scope $\{m_{11}, s_1, m_{21}, s_2, m_{22}, m_{23}, m_{31}, s_3\}$.

Formally, for mention m in mention group

$M(m)$, its *extended semsum* $S_{extended}(m)$ is:

$$S_{extended}(m) = S_{basic}(m) \cup \left(\bigcup_{m'} (S_{basic}(m') \mid \exists s : \; m' \in s \wedge m \in s) \right)$$

where s is a sentence in which both m and m' occur.

In principle, we could consider even more aggressive expansions, like 4-hop neighbors or transitive closures. However, our experiments show that the 2-hop extension is a sweet spot that gains substantial benefits over the basic scope.

Enrichment Matching: For each local mention group, *CROCS* first inspects the *coarse-grained types* (person, organization, location) as determined by the Stanford NER Tagger. We consider pronouns to derive additional cues for person mentions. If all tags in a group agree, we mark the group by this tag; otherwise the group as a whole is not type-tagged.

To match a mention group against a KB entity, we trigger a phrase query comprising tagged phrases from the mention group to the KB interface[1]. We remove non-informative words from the phrases, dropping articles, stop-words, etc. For example, the first mention group, $\{m_{11}, m_{21}\}$ in Fig. 1 leads to the query `"president wife first lady"`. The query results are filtered by matching the result type-tag with the type tag of the mention group. For the extended scope, we construct analogous queries for the co-occurring mentions: `"White House US president residence"` and `"husband"` in the example. The results are processed as follows.

We primarily rely on the KB service itself to rank the matching entities by confidence and/or relevance/importance. We simply accept the top-ranked entity and its KB properties, and extend the semsum's on this basis. This is also done for the co-occurring mention groups, leading to the *extended scope* of the original mention group considered.

To avoid dependency on the ranking of the KB, we can alternatively obtain the top-k results for each query and also the KB's confidence for the entity matching. We then re-rank the candidates by our similarity measures and prune out candidates with low confidence. We introduce a *confidence threshold*, θ, such that all candidates having matching confidence below the threshold are ignored, i.e., the entire mention group is disregarded in the semsum construction. This makes extended scope robust to noise. For example, the mention group $\{husband\}$ having low confidence would likely degrade the semsum's quality and is thus dropped.

[1]For example, (`https://gate.d5.mpi-inf.mpg.de/webyagospotlx/WebInterface`) or (`www.freebase.com/query`)

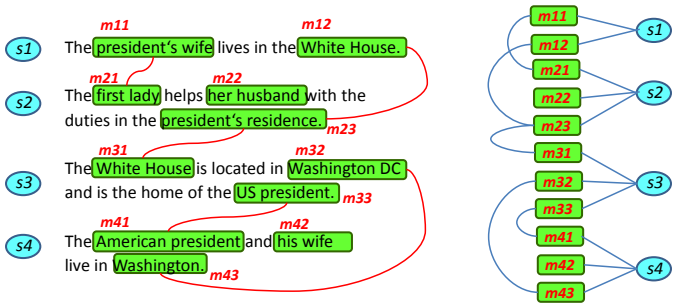

Figure 1: Example of local mention groups.

Algorithm 1: *Extended Knowledge Enrichment*

Require: Text T, Set G of mention groups (from Stanford CoreNLP), KB Match Threshold θ, Knowledge base KB

Ensure: semsum for each group in G

1: **for** each mention group, $M \in G$ **do**
2: *Basic Scope:* $semsum_M \leftarrow$ sentences from T containing mentions in M
3: Extract and add KB features for mentions and phrases in $semsum_M$ ($S_{basic}(M)$)
4: *Extended Scope:* Append context of 2-hop co-occurring mentions (from bipartite graph) to $semsum_M$
5: *Matching:* Extract phrases from $semsum_M$ for query generation to KB
6: Retrieve highest ranked KB result entity e
7: **if** match confidence of $e > \theta$ **then**
8: Extract set of features for e, L_e from KB
9: Append L_e to $semsum_M$ ($S_{extended}(M)$)
10: **end if**
11: **end for**
12: Output $semsum_M$ for all $M \in G$

Feature Vector: The semsum's of the mention groups comprise sentences and bags of phrases. For the example mention group $\{m_{11}, m_{21}\}$, we include the sentences $\{s_1, s_2, s_3\}$ during the extended-scope enrichment, and obtain phrases from the KB like: "Michelle Obama", "First Lady of United States", "capital of the United States", etc. Algorithm 1 shows the pseudo-code for constructing semsum's.

CROCS next casts each semsum into two forms, (i) a bag of words, and (ii) a bag of keyphrases, and uses both for constructing a feature vector.

5 Similarity Computation

$CROCS$ compares mention groups by a similarity measure to infer whether they denote the same entity or not. The similarity is based on the feature vectors of mention groups (constructed as in Section 4). Each feature in a mention group's vector is weighted using IR-style measures according to the bag-of-words (BoW) model or the keyphrases (KP) model for the semsum's. Empirically, the best approach is a mixture of both the words and keyphrases model, which is employed by $CROCS$. Similarity comparisons are computed on-demand and only for a small sampled set of mention groups, as required during the hierarchical clustering procedure (see Section 6).

The similarity of two mentions groups G_1, G_2 is,

$$sim(G_1, G_2) = \alpha \times sim_{BoW}(G_1, G_2) + (1 - \alpha) \times sim_{KP}(G_1, G_2)$$

where α is a tunable hyper-parameter. Whenever two mention groups are to be combined (referring to the same entity), their feature vectors are combined by computing a bag union of their words and/or phrases, and then recomputing the weights. Without loss of generality, our default setting is $\alpha = 0.5$.

Bag-of-Words Model (BoW): For this model, we compute the term frequency, $tf(w)$ for each word w in the semsum's, and also the inverse document frequency, $idf(w)$, of the word across all semsum's (i.e., all mention groups from all input documents). The weight of w , $wgt(w) = tf(w) \times idf(w)$. As the semsum's are short, we use the simple product rather than dampening tf values or other variations. Alternatively, more advanced IR weighting models such as Okapi BM25 or statistical language models can be used. However, the classical $tf \times idf$ measure works quite well. $CROCS$ computes the similarity of two feature vectors by their *cosine distance*.

Keyphrases Model (KP): The keyphrases of a mention group are obtained by extracting proper names, titles, alias names, locations, organization, etc., from its semsum's. Similar to the BoW model, CROCS supports $tf \times idf$ style weights for entire keyphrases.

For computing the similarity of keyphrases between two mention groups G_1 and G_2, $CROCS$ matches the keyphrases of G_1 in the semsum's of G_2, and vice versa. However, entire phrases rarely match exactly. For example, the keyphrase *"Peace Nobel"* match only partially in the text *"Nobel prize for Peace"*. To consider such partial matches and reward both high overlap of words and short distances between matching words (locality), we adopt the scoring model of (Taneva et al., 2011). The score for a partial match of keyphrase p in text x is,

$$S(p|x) = \frac{\#\ match\ words}{len.\ of\ cov(p|x)} \left(\frac{\sum_{w \in cov(p)} wgt(w)}{\sum_{w \in p} wgt(w)} \right)^{1+\gamma}$$

where the *cover* (cov) of p in x is the shortest word span (in x) containing all the words of p present in x (with a bound of 10-20 words). For the example above, the cover of p = *"Peace Nobel"* in the text x is "Nobel prize for Peace" (all 2 words matching with cover length 4). The parameter γ, $(0 < \gamma < 1)$ serves to tune the progression of penalizing missing words. In our experiments, γ was set to 0.5 and stopwords such as "a", "the", etc. were removed with only keywords being considered.

For mention groups G_1 and G_2, we compute,

$$sim(G_1|G_2) = \sum_{p \in KP(G_1)} wgt(p) \times S(p|semsum's(G_2))$$

Finally, we resolve the asymmetry in similarity measure due to the ordering of the two groups by setting,

$$sim(G_1, G_2) = max\{sim(G_1|G_2), sim(G_2|G_1)\}$$

6 Clustering Algorithm

The final stage of $CROCS$ takes the mention groups and the semsum's as input. It performs a top-down hierarchical bisection process, based on similarity scores among entities, to cluster together co-referring mention groups at each splitting level.

Initially all mention groups are placed in a single cluster, and are then recursively split until a stopping criterion finalizes a cluster as leaf. At each level, cluster splitting is performed by using either *spectral clustering* (Luxburg, 2007) or *balanced graph partitioning* (Karypis & Kumar, 1998). Both these methods implicitly consider transitivity, which is essential as the equivalence classes of mentions should be transitively closed. The challenge of this seemingly simple procedure lies in (i) judiciously choosing and optimizing the details (model selection and stopping criterion), and (ii) reducing the computational cost. The latter is crucial as spectral clustering has cubic complexity, graph partitioning heuristics computations are expensive, and CCR (unlike CR) needs to cope with Web-scale inputs consisting of millions of documents and entities.

Clustering is invoked for each of the coarse-grained entity types separately (as obtained from Stanford NER tagger): people, places, and organizations. The benefit is twofold: gaining efficiency and improving accuracy, as two different entity types would not co-refer in reality. However, the risk is that two differently tagged mention groups might actually refer to the same entity, with at least one tag being incorrect. Our experiments show that the benefits clearly outweigh this risk. Without loss of generality, we only consider chains that are tagged into one of the above types, and other co-reference chains are ignored. Although this might lead to certain mentions being overlooked, improving the accuracy and recall of NER tagging approaches are orthogonal to our current scope of work.

Active spectral clustering: Spectral clustering (Luxburg, 2007) uses the eigenspace of the similarity graph's Laplacian matrix to compute graph partitions as clusters. $CROCS$ adopts the recently proposed *Active Spectral Clustering* technique (Krishnamurty et al., 2012; Wauthier et al., 2012), which approximates the eigenspace of a Laplacian with a small subset of *sampled data points* (mention groups in CROCS). For n data points and sample size s in the order of $O(\log n)$, this technique reduces the cost of spectral clustering from $O(n^3)$ to $O(\log^3 n)$ (with bounded error). CROCS initializes each bisection step by selecting s mention groups from a cluster and computes all pair-wise similarities among the sampled groups. Spectral clustering is then performed on this subset to obtain a split into 2 clusters. The non-sampled mention groups are assigned to the closest cluster in terms of average distance to cluster centroids. The children clusters are iteratively split further at next levels until the stopping criterion fires.

Balanced graph partitioning: Balanced graph partitioning assigns the vertices of a graph into components of nearly the same size having few edges across components. The problem is NP-complete, and several approximation algorithms have been proposed (Buluc et al., 2013). $CROCS$ uses the *METIS* software (`http://glaros.dtc.umn.edu/gkhome/metis/metis/overview`) to obtain mention group partitioning at each level of the hierarchical clustering.

The underlying mention similarity graph is constructed by sampling s mention groups, and similarities among them represented as edge weights. For mention groups not selected in the sample, similarities to only the s sample points are computed and corresponding edges created. The graph is then partitioned using METIS (Karypis & Kumar, 1998) (multi-level recursive procedure) to minimize the edge-cuts thereby partitioning dissimilar mention groups.

Specifics of CROCS: Active spectral clustering (Krishnamurty et al., 2012) uses random sampling, chooses the number of final clusters, k based on eigengap, and enforces a balancing constraint for the k clusters to be of similar sizes. CROCS judiciously deviates from the design of (Krishnamurty et al., 2012) as:

- *Model selection:* We choose a fixed number of partitions k at each cluster-splitting step of the hierarchical process. We use a small k value, typically $k = 2$. This avoids selecting model dimension parameters, allowing the stopping criterion to decide the final number of clusters.
- *Form of graph cut:* CROCS uses balanced normalized cut for graph partitioning (Karypis & Kumar, 1998). However, unbalanced cluster sizes with several singleton clusters (having only one mention group) might be formed. In our CCR setting, this is actually a natural outcome as many long-tail entities occur only once in the corpus. Such mention groups significantly differ in semantic and contextual features compared to the other mention groups. Hence, singleton cluster mentions have low similarity score (based on semsum's) with other mentions groups. This translates to low edge weights in the underlying similarity graph structure (between mentions), thus forming favorable candidates in the initial phases of cluster splitting using minimum edge-cut based graph partitioning. Therefore, CROCS inherently incorporates early partition (during the clustering phase) of such possibly singleton mention clusters from the "main data", thereby helping in de-noising and efficiency.
- *Sampling:* Instead of sampling data points uniformly randomly, we use biased sampling similar to initialization used in k-means clustering. Starting with a random point, we add points to the sample set such that their average similarity to the already included points is minimized, thus maximizing the diversity among the samples.

Stopping criterion of CROCS: The sample-based

hierarchical clustering process operates without any prior knowledge of the number of clusters (entities) present in the corpus. We use the *Bayesian Information Criteria* (BIC) (Schwarz, 1978; Hourdakis et al., 2010) as the *stopping criterion* to decide whether a cluster should be further split or finalized. BIC is a Bayesian variant of the *Minimum Description Length* (MDL) principle (Grunwald, 2007), assuming the points in a cluster to be Gaussian distributed. The BIC score of a cluster C with s (sampled) data points, x_i and cluster centroid $\bar{C}$ is:

$$BIC(C) = \sum_{i=1,\cdots,s} \log_2(x_i - \bar{C})^2 \; + \; \log_2 s$$

The BIC score for a set of clusters is the micro-averaged BIC of the clusters. $CROCS$ splits a cluster C into sub-clusters $C_1, \ldots, C_k$ iff the combined BIC value of the children is greater than that of the parent, else C is marked as leaf.

7 Experimental Evaluation

Benchmark Datasets: We performed experiments with the following three publicly available benchmarking datasets, thereby comparing the performance of $CROCS$ against state-of-the-art baselines under various input characteristics.

- ***John Smith* corpus**: the classical benchmark for CCR (Bagga & Baldwin, 1998) comprising 197 articles selected from the New York Times. It includes mentions of 35 different "John Smith" person entities. All mentions pertaining to John Smith within a document refer to the same person. This provides a small-scale but demanding setting for CCR, as most John Smiths are long-tail entities unknown to Wikipedia or any KB.
- **WePS-2 collection**: a set of 4,500 Web pages used in the *Web People Search 2* competition (Artiles et al., 2009). The documents comprise the top 150 Web search results (using Yahoo! search (as of 2008)) for each of 30 different people (obtained from Wikipedia, ACL'08, and US Census), covering both prominent entities (e.g., Ivan Titov, computer science researcher) and long-tailed entities (e.g., Ivan Titov, actor).
- **New York Times (NYT) archive**: a set of around 1.8 million news article from the archives of the newspaper (Sandhaus, 2008) extracted between January 1987 and June 2007. We randomly select $220,000$ articles from the time range of January 1, 2004 through June 19, 2007, which contain about 3.71 million mentions, organized into 1.57 million local mention chains after the intra-document CR step.

In our experiments, we consider mentions of person entities as this is the most predominant and demanding class of entities in the datasets. The John Smith and WePS-2 datasets have explicit ground truth annotations, while the NYT contains editorial annotations for entities present in each article. For knowledge enrichment, we used Freebase; although sensitivity studies explore alternative setups with Yago.

Evaluation Measures: We use the established measures to assess output quality of CCR methods:

- B^3 **F1 score** (Bagga & Baldwin, 1998): measures the F1 score as a harmonic mean of precision and recall of the final equivalence classes. Precision is defined as the ratio of the number of correctly reported co-references (for each mention) to the total number; while recall computes the fraction of actual co-references correctly identified. Both the final precision and recall are computed by averaging over all mention groups.
- ϕ_3**-CEAF score** (Luo, 2005): an alternate way of computing precision, recall, and F1 scores using the best 1-to-1 mapping between the equivalence classes obtained and those in the ground truth. The best mapping of ground-truth to output classes exhibits the highest mention overlap.

All experiments were conducted on a 4 core Intel i5 2.50 GHz processor with 8 GB RAM running Ubuntu 12.04.

7.1 Parameter Tuning

The use of external features extracted from KB's (for mention groups) forms an integral part in the working of $CROCS$, and is represented by the choice of the *threshold*, θ. Given an input corpus, we now discuss the tuning of θ based on splitting the available data into *training* and *testing* subsets.

We randomly partition the input data into 3 parts (assumed to be labeled as A, B, and C). One of the data parts is the training data and the other two parts are the test data. Using the gold annotations of the training dataset, we empirically learn the value of θ, that provides the best B^3 F1 score for CCR, using a simple *line search*. Initially, θ is set to 1 (no KB usage) and is subsequently decreased using 0.01

Method	P (%)	R (%)	F1 (%)
CROCS	78.54	72.12	**75.21**
Stream (Rao, 2010)	84.7	59.2	69.7
Inference (Singh, 2011)	-	-	66.4

Table 1: B^3 F1 results on John Smith dataset.

Method	P (%)	R (%)	F1 (%)
$CROCS$	85.3	81.75	**83.48**
PolyUHK (Artiles, 2009)	87	79	82
UVA_1 (Artiles, 2009)	85	80	81

Table 2: B^3 F1 results on WePS-2 dataset.

as the step size for each of the learning phase iterations. As soon as the performance of CROCS is seen to degrade (compared to the previous iteration), the procedure is terminated and the previous value of θ is considered as the learned parameter value. The final results we report are averaged over 3 independent runs, each considering different data partitions (among A, B, and C) as the training data. Although more advanced learning algorithms might also be used, this simple approach is observed to work well.

Learning of the θ value might converge to a local maximum, or may be distorted due to presence of noise in the training data. However, we later show (in Section 7.5) that the performance of $CROCS$ is robust to small variations of θ.

7.2 John-Smith Corpus: Long-Tail Entities

Table 1 compares $CROCS$ with two state-of-the-art methods achieving the best published results for this benchmark. 66 randomly selected documents were used as the training set (while the rest formed the test set) and the subsequent θ value learned (as described in Section 7.1) was 0.96. Since the corpus contained mostly long-tail entities not present in any KB (only *5-6* of the 35 different John Smith's are in Wikipedia, eg. the explorer John Smith etc.), the KB matches were too unreliable and led to the introduction of noise. Hence, a high value of θ was obtained (i.e. KB features mostly disregarded).

$CROCS$ (using sample-based spectral clustering) achieves an *F1 score* of 75.21%, while *Stream* (Rao et al., 2010) and *Inference* (Singh et al., 2011) reach only 69.7% and 66.4% resp. $CROCS$ also has a high ϕ_3-CEAF score of 69.89% exhibiting substantial gains over prior methods[2]. Our novel notion of semsum's with extended scope (mentions and co-occurring mention groups) proved essential for outperforming the existing methods (see Section 7.6). The runtime of CROCS was only around 6 seconds.

7.3 WePS-2 Corpus: Web Contents

We compared sampled spectral clustering based $CROCS$ on the WePS-2 corpus against the best methods reported in (Artiles et al., 2009). We empirically obtained the KB match parameter $\theta = 0.68$ according to the train-test setup described earlier (with 1500 training documents).

$CROCS$ achieves a B^3 based $F1$ score of 83.48% and a ϕ_3-CEAF score of 74.02% (Table 2), providing an improvement of about 1.5 F1 score points[3]. The gain observed is not as high as that for the John Smith dataset, as in the WePS-2 corpus documents are longer, giving richer context with fewer ambiguous entity mentions. Thus, simpler methods also perform fairly well. The runtime of CROCS on WePS-2 corpus was about 90 seconds.

7.4 New York Times Corpus: Web Scale

The previous two datasets, John Smith and WePS-2 are too small to assess the robustness of CROCS for handling large data. We therefore ran $CROCS$ (with sample-based spectral clustering) on a random sample of 220,000 NYT news articles. The knowledge enrichment threshold θ was learned to be 0.45 with 73K training documents.

$CROCS$ achieved a B^3 $F1$ score of 59.17% (with $P = 56.18\%$ and $R = 62.49\%$) and a ϕ_3-CEAF score of 50.0%. No prior methods report F1 performance figures for this large dataset. However, the factor graph based approach of (Singh et al., 2010) measures the mention co-reference accuracy for a sample of 1,000 documents. Accuracy is defined as the ratio of document clusters assigned to an entity to the ground truth annotations. We also sampled 1,000 documents considering only mentions with multiple entity candidates. $CROCS$ achieved an accuracy of 81.71%, as compared to 69.9% for (Singh et al., 2010).

As for run-time, $CROCS$ took 14.3 hours to process around 150,000 news articles selected as the test corpus. We also compared this result against al-

[2]Data and detailed $CROCS$ output results are available at (`www.dropbox.com/s/1grribug15yghys/John_Smith_Dataset.zip?dl=0`)

[3]Data and detailed $CROCS$ output results are available at (`www.dropbox.com/s/1i9ot4seavcfdyc/WePS-2_Dataset.zip?dl=0`)

CROCS configuration	WePS-2	NYT
Sentences only	50.35	39.52
Basic Scope	64.14	53.88
Extended Scope	**83.48**	**59.17**
NED baseline	61.25	59.62

Table 3: B^3 F1 scores for CROCS enrichment variants.

ternative algorithms within our framework (see Section 7.6). Hence, CROCS efficiently handles Web scale input data.

7.5 Sensitivity Studies

The $CROCS$ framework involves a number of tunable hyper-parameters adjusting the precise performance of the components. We now study the robustness of $CROCS$ (sample-based spectral clustering variant) for varying parameter values.

Knowledge Enrichment Scope:
$CROCS$ supports several levels of knowledge enrichment for semsum's construction: i) including only sentences of a mention group (disregarding the KB), ii) using distant KB labels for the given mention group only (basic scope), and iii) adding distant KB labels for co-occurring mention groups (extended scope). We compared these configurations among each other and also against a state-of-the-art NED method alone. The results are shown in Table 3. We used AIDA (Hoffart et al., 2011) open-source software (`https://github.com/yago-naga/aida`) for NED, and combined mentions mapped to the same KB entity. We use the trained value of θ obtained previously (for the respective datasets) for constructing the *basic and extended scope* of semsum's, and report the best B^3 F1 scores. Note that the *Sentences only* and *NED* configurations are independent of the choice of θ value.

Real-life Web articles contain a mixture of prominent entities, ambiguous names, and long-tail entities; hence sole reliance on NED for CCR fares poorly. The extended scope semsum's construction produces superior results compared to other models.

Knowledge Enrichment Matching Threshold:
To study the influence of different degrees of distant KB feature extraction, we varied the enrichment matching threshold θ from 0.0 (accept all KB matches) to 1.0 (no import from KB). The John Smith dataset largely containing long-tail entities uses $\theta \sim 1$ (trained value), and operates on semsum's containing practically no feature inclusion from external KB's. Hence, we only consider the scenario when the KB is completely disregarded (i.e. $\theta = 1.0$) and obtain a B^3 F1 score of 76.47%.

Dataset	θ						
	0.0	0.25	0.5	0.65	0.75	0.9	1.0
WePS-2	76.9	77.3	82.4	83.9	75.7	68.9	63.5
NYT	60.5	61.5	62.2	62.2	60.0	52.1	48.4

Table 4: B^3 F1 scores (%) for different choices of θ.

Dataset	θ used	P(%)	R(%)	F1(%)
WePS-2	0.45	83.46	80.21	81.9
NYT	0.68	59.42	64.2	61.8

Table 5: θ error sensitivity of CROCS

For the other two datasets, the B^3 F1 results for varying θ are shown in Table 4. We observe that KB features help the CCR process and the best results are obtained for θ between 0.6 and 0.7. We observe that the exact choice of θ is not a sensitive issue, and any choice between 0.25 and 0.75 yields fairly good F1 scores (within 10% of the empirically optimal F1 results). Hence, our approach is robust regarding parameter tuning.

We observe that the trained value of θ (obtained previously) for both the WePS-2 and the NYT datasets are close to the optimal setting as seen from Table 4 and provide nearly similar F1 score performance. Therefore, we set $\theta = 0.65$ and consider the entire input corpora as test set for the remainder of our experiments.

To reconfirm the robustness of $CROCS$ to θ value ranges, we use the KB threshold trained on WePS-2 dataset, and test it on the NYT dataset (and vice versa). From Table 5 we observe CROCS to render comparable performance in presence of errors during the θ learning phase.

Clustering Hyper-Parameters:
We study the effect of varying k, the number of subclusters for the bisection procedure invoked at each level of the hierarchical clustering. By default, this is set to 2 (i.e. bisection). Table 6 shows the B^3 F1 scores for different choices of k, for the three datasets (with $\theta = 1.0$ for John Smith and $\theta = 0.65$ for the other two datasets). We observe that $k = 2$ performs best in all cases. The output quality monotonically drops with increase in k, as this forces even similar mention groups to form separate clusters. Hence, bisection allows the hierarchical process to adjust the model selection at the global level.

Alternative KB:
To assess the impact of dependency on Freebase (feature extraction of best matching entity), we

Dataset	k=2	k=3	k=4	k=5
John Smith	**76.47**	73.24	65.29	60.7
WePS-2	**83.92**	82.61	78.37	73.19
NYT	**62.24**	59.34	52.60	46.64

Table 6: B^3 F1 scores (%) for different # sub-clusters k.

Dataset	Freebase			Yago		
	P(%)	*R(%)*	*F1*	*P(%)*	*R(%)*	*F1*
WePS-2	86.3	82.1	83.9	86.6	82.5	84.0
NYT	59.8	64.9	62.2	61.3	60.8	61.0

Table 7: CROCS B^3 F1 scores with Freebase vs. Yago

ran alternative experiments on the WePS-2 and NYT datasets with the Yago KB (`www.yago-knowledge.org`). We obtain all approximate matches for a mention group and rank them based on the keyphrase similarity model (Section 5) using sentences of the mention group and extracted features (from the Yago $hasLabel$ property and infoboxes in Wikipedia pages of the $sameAs$ link). Results in Table 7 show similar performance, depicting no preference of $CROCS$ to any particular KB.

7.6 Algorithmic Variants

The $CROCS$ framework supports a variety of algorithmic building blocks, most notably, clustering methods (eg., k-means) or graph partitioning for the bisection steps, and most importantly, sampling-based methods versus methods that fully process all data points. The comparative results for the three different datasets are presented in Table 8.

For the John Smith corpus (with $\theta = 1.0$), all algorithms except sample-based k-means achieved similar performances in accuracy and runtime. The best method was the full-fledged spectral clustering, with about 2% F1 score improvement.

With the WePS-2 dataset, we obtain a similar picture w.r.t. output quality. However, this dataset is large enough to bring out the run-time differences. Sampling-based methods, including CROCS, were about $4\times$ faster than their full-fledged counterparts, albeit with a meager loss of about 2% in F1 score.

The NYT dataset finally portrays the scenario on huge datasets. Here, only the sample-based methods ran to completion, while all the full-fledged methods were terminated after 20 hours. The fastest of them, the simple k-means method, had processed only about 5% of the data at this point (needing about 400 hours on extrapolation). In contrast, CROCS, using sample-based spectral clustering or graph partitioning, needed about 19.6 hours for the 220,000 documents. The sampling-based k-means competitor was slightly faster (17.8 hours), but lost dramatically on output quality: with only about 42% F1 score compared to 62% F1 score for $CROCS$.

Hence, we observe that $CROCS$ is indeed well designed for scalable sampling-based CCR, whereas other simpler methods like k-means, lacking transitivity awareness, fail to deliver good output quality.

8 Related Work

Co-reference Resolution (CR): Existing intra-document CR methods combine syntactic with semantic features for identifying the best antecedent (preceding name or phrase) for a given mention (name, phrase, or pronoun). Syntactic features are usually derived from deep parsing of sentences and noun group parsing. Semantic features are obtained by mapping mentions to background knowledge resources such as Wikipedia. An overview of CR methods is given in (Ng, 2010). Recent methods adopt the paradigm of *multi-phase sieves*, applying a cascade of rules to narrow down the choice of antecedents for a mention (e.g., (Haghighi & Klein, 2009; Raghunathan et al., 2010; Ratinov & Roth, 2012)). The cluster-ranking family of methods (e.g., (Rahman & Ng, 2011b)) extends this paradigm for connecting mentions with a cluster of preceding mentions. Person name disambiguation in CR deals with only person names, title, nicknames, and surface forms variations (Chen & Martin, 2007).

Distant Knowledge Labels for CR: To obtain semantic features, additional knowledge resources such as Wikipedia, Yago ontology, and FrameNet corpus have been considered (Suchanek et al., 2007; Rahman & Ng, 2011a; Baker, 2012). To identify the entity candidate(s) that a mention (group) should use for distant supervision, CR methods such as (Ratinov & Roth, 2012; Lee et al., 2013) use matching heuristics based on the given mention alone to identify a single entity or all matching entities with confidence above some threshold. Zheng et al. (2013) generalizes this by maintaining a ranked list of entities for distant labeling, as mention groups are updated. Unlike CROCS, prior methods utilize only the candidates for the given mention (group) itself and distant knowledge features for co-occurring mentions are not considered.

Cross-Document CR (CCR): Early works (Gooi & Allan, 2004) on CCR, introduced by (Bagga &

Dataset	Clustering Method	B^3 measure			ϕ_3 measure (%)	Run-time
		P (%)	*R (%)*	*F1 (%)*		
John Smith	**Spectral clustering**	79.6	80.1	79.85	73.52	8.11 sec
	k-means clustering	71.27	83.83	77.04	71.94	8.01 sec
	Balanced graph partition	75.83	79.56	77.65	70.63	7.83 sec
	Sampled k-means	63.57	65.52	64.53	59.61	5.12 sec
	Sampled spectral clustering	79.53	73.64	76.47	70.25	6.5 sec
	Sampled graph partitioning	71.42	77.83	74.49	68.36	6.86 sec
WePS-2	**Spectral clustering**	88.2	85.61	86.88	77.91	331 sec
	k-means clustering	85.7	84.01	84.85	76.45	296.56 sec
	Balanced graph partition	86.56	82.78	84.63	77.73	324.64 sec
	Sampled k-means	72.67	68.56	70.56	66.92	72 sec
	Sampled spectral clustering	86.2	82.11	83.92	74.7	85.8 sec
	Sampled graph partitioning	85.3	82.2	83.72	74.5	83.65 sec
New York Times	**k-means clustering***	39.34*	49.17*	43.72*	31.45*	>20 hrs
	Sampled k-means	40.45	45.34	42.76	40.61	17.8 hrs
	Sampled spectral clustering	59.78	64.92	62.24	51.02	19.6 hrs
	Sampled graph partitioning	61.45	62.71	62.07	50.88	19.7 hrs

* results after run terminated at 20 hrs (∼5% mentions processed)

Table 8: Accuracy and scalability of various algorithms embedded in $CROCS$

Baldwin, 1998), used IR-style similarity measures (tf×idf cosine, KL divergence, etc.) on features, similar to intra-document CR. Recent works such as (Culotta et al., 2007; Singh et al., 2010; Singh et al., 2011) are based on probabilistic graphical models for jointly learning the mappings of all mentions into equivalence classes. The features for this learning task are essentially like the ones in local CR. Baron and Freedman (2008) proposed a CCR method involving full clustering coupled with statistical learning of parameters. However, this method does not scale to large corpora making it unsuitable for Web contents. A more light-weight online method by (Rao et al., 2010) performs well on large benchmark corpora. It is based on a streaming clustering algorithm, which incrementally adds mentions to clusters or merges mention groups into single clusters, and has linear time complexity; albeit with inferior clustering quality compared to advanced methods like spectral clustering. Several CCR methods have harnessed co-occurring entity mentions, especially for the task of disambiguating person names (Mann & Yarowsky, 2003; Niu et al., 2004; Chen & Martin, 2007; Baron & Freedman, 2008). However, these methods do not utilize knowledge bases, but use information extraction (IE) methods on the input corpus itself; thus facing substantial noise due to IE quality variance on stylistically diverse documents like Web articles.

Spectral Clustering: (Luxburg, 2007) provides a detailed study on spectral clustering models and algorithms. Yan et al. (2009) proposed two approximation algorithms, based on the k-means technique and random projections, reducing the $O(n^3)$ time complexity to $O(k^3) + O(kn)$ where k is the number of clusters. In CCR, the number of clusters (truly distinct entities) can be huge and typically unknown; hence (Shamir & Tishby, 2011; Krishnamurty et al., 2012; Wauthier et al., 2012) developed active spectral clustering, where the expensive clustering step is based on data samples and other data points are merely "folded in". The term "active" refers to the active learning flavor of choosing the samples (notwithstanding that these methods mostly adopt uniform random sampling).

9 Conclusions

We have presented the $CROCS$ framework for cross-document co-reference resolution (CCR). It performs sample-based spectral clustering or graph partitioning in a hierarchical bisection process to obtain the mention equivalence classes, thereby avoiding model-selection parameters and the high cost of clustering or partitioning. $CROCS$ constructs features for mention groups by considering co-occurring mentions and obtaining distant semantic labels from KB's (for semsum's).

Feature generation from multiple KB's and catering to streaming scenarios (e.g., news feeds or social media) are directions of future work.

References

J. Artiles, J. Gonzalo, S. Sekine: 2009. WePS 2 Evaluation Campaign: Overview of the Web People Search Clustering Task. WWW 2009.

A. Bagga, B. Baldwin: 1998. Entity-Based Cross-Document Coreferencing Using the Vector Space Model. *In COLING-ACL*, pages 79–85.

C. F. Baker: 2012. FrameNet, Current Collaborations & Future Goals. *LREC*, 46(2):269–286.

A. Baron, M. Freedman: 2008. Who is Who & What is What: Experiments in Cross-Document Co-Reference. *In EMNLP*, pages 274–283.

A. Buluc, H. Meyerhenke, I. Safro, P. Sanders, C. Schulz: 2013. Recent Advances in Graph Partitioning. *Karlsruhe Institute of Technology, Technical Report*.

J. Cai, M. Strube: 2010. Evaluation Metrics for End-to-End Coreference Resolution Systems. *In SIGDIAL*, pages 28–36.

Y. Chen, J. Martin: 2007. Towards Robust Unsupervised Personal Name Disambiguation. *In EMNLP-CoNLL*, pages 190–198.

M. Cornolti, P. Ferragina, M. Ciaramita: 2013. A Framework for Benchmarking Entity-Annotation Systems. *In WWW*, pages 249–260.

S. Cucerzan: 2007. Large-Scale Named Entity Disambiguation Based on Wikipedia Data. *In EMNLP-CoNLL*, pages 708–716.

A. Culotta, M. L. Wick, A. McCallum: 2007. First-Order Probabilistic Models for Coreference Resolution. *In HLT-NAACL*, pages 81–88.

P. Domingos, S. Kok, D. Lowd, H. Poon, M. Richardson, P. Singla: 2007. Markov Logic. *Probabilistic ILP*. Springer-Verlag, pages 92–117.

P. Domingos, D. Lowd: 2009. *Markov Logic: An Interface Layer for Artificial Intelligence*. Morgan and Claypool Publishers, 2009.

J. R. Finkel, T. Grenager, C. D. Manning: 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. *In ACL*, pages 363–370.

C. H. Gooi, J. Allan: 2004. Cross-Document Coreference on a Large Scale Corpus. *In HLT-NAACL*, pages 9–16.

P. D. Grünwald: 2007. *The Minimum Description Length Principle*. MIT University Press.

A. Haghighi, D. Klein: 2009. Simple Coreference Resolution with Rich Syntactic and Semantic Features. *In EMNLP*, pages 1152–1161.

A. Haghighi, D. Klein: 2010. Coreference Resolution in a Modular, Entity-Centered Model. *In HLT-NAACL*, pages 385–393.

H. Hajishirzi, L. Zilles, D. S. Weld, L. S. Zettlemoyer: 2013. Joint Coreference Resolution and Named-Entity Linking with Multi-Pass Sieves. *In EMNLP*, pages 289–299.

J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, G. Weikum: 2011. Robust Disambiguation of Named Entities in Text. *In EMNLP*, pages 782–792.

N. Hourdakis, M. Argyriou, G. M. Petrakis, E. E. Milios: 2010. Hierarchical Clustering in Medical Document Collections: the BIC-Means Method. *Journal of Digital Information Management*, 8(2):71–77.

G. Karypis, V. Kumar: 1998. A Fast and Highly Quality Multilevel Scheme for Partitioning Irregular Graphs. *Journal on Scientific Computing*, 20(1):359–392.

B. W. Kernighan, S. Lin: 1970. An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal*.

D. Koller, N. Friedman: 2009. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

A. Krishnamurty, S. Balakrishnan, M. Xu, A. Singh: 2012. Efficient Active Algorithms for Hierarchical Clustering. *In ICML*, pages 887–894.

H. Lee, Y. Peirsman, A. Chang, N. Chambers, M. Surdeanu, D. Jurafsky: 2011. Stanford's Multi-Pass Sieve Coreference Resolution System at the CoNLL-2011 Shared Task. *In CoNLL*, pages 28–34.

H. Lee, A. Chang, Y. Peirsman, N. Chambers, M. Surdeanu, D. Jurafsky: 2013. Deterministic Coreference Resolution based on Entity-centric, Precision-ranked Rules. *Computational Linguistics Journal*, 39(4):885–916.

H. Lee, M. Recasens, A. X. Chang, M. Surdeanu, D. Jurafsky: 2012. Joint Entity and Event Coreference Resolution across Documents. *In EMNLP-CoNLL*, pages 489–500.

H. A. Loeliger: 2008. An Introduction to Factor Graphs. *In MLSB*.

X. Luo: 2005. On Coreference Resolution Performance Metrics. *In HLT-EMNLP*, pages 25–32.

U. von Luxburg: 2007. A Tutorial on Spectral Clustering. *Statistics and Computing Journal*, 17(4):395–416.

G. S. Mann, D. Yarowsky: 2003. Unsupervised Personal Name Disambiguation. *In CoNLL,HLT-NAACL*, pages 33–40.

D. N. Milne, I. H. Witten: 2008. Learning to Link with Wikipedia. *In CIKM*, pages 509–518.

V. Ng: 2010. Supervised Noun Phrase Coreference Research: The First Fifteen Years. *In ACL*, pages 1396–1411.

C. Niu, W. Li, R. K. Srihari: 2004. Weakly Supervised Learning for Cross-document Person Name Disambiguation Supported by Information Extraction. *In ACL*, article 597.

K. Raghunathan, H. Lee, S. Rangarajan, N. Chambers, M. Surdeanu, D. Jurafsky, C. Manning: 2010. A Multi-Pass Sieve for Coreference Resolution. *In EMNLP*, pages 492–501.

A. Rahman, V. Ng: 2011a. Coreference Resolution with World Knowledge. *In ACL*, pages 814–824.

A. Rahman, V. Ng: 2011b. Ensemble-Based Coreference Resolution. *In IJCAI*, pages 1884–1889.

D. Rao, P. McNamee, M. Dredze: 2010. Streaming Cross Document Entity Coreference Resolution. *In COLING*, pages 1050–1058.

L. A. Ratinov, D. Roth, D. Downey, M. Anderson: 2011. Local and Global Algorithms for Disambiguation to Wikipedia. *In ACL*, pages 1375–1384 .

L. A. Ratinov, D. Roth: 2012. Learning-based Multi-Sieve Co-reference Resolution with Knowledge. *In EMNLP-CoNLL*, pages 1234–1244.

M. Richardson, P. Domingos: 2006. Markov Logic Networks. *Journal of Machine Learning*, 62(1-2):107–136.

E. Sandhaus: 2008. The New York Times Annotated Corpus Overview. *Linguistic Data Consortium*.

G. E. Schwarz: 1978. Estimating the Dimension of a Model. *Annals of Statistics*, 6(2):461–464.

O. Shamir, N. Tishby: 2011. Spectral Clustering on a Budget. *Journal of Machine Learning Research* - Proceedings Track 15:661–669.

S. Singh, M. L. Wick, A. McCallum: 2010. Distantly Labeling Data for Large Scale Cross-Document Coreference. *CoRR abs/1005.4298*.

S. Singh, A. Subramanya, F. Pereira, A. McCallum: 2011. Large-Scale Cross-Document Coreference Using Distributed Inference and Hierarchical Models. *In ACL*, pages 793–803.

F. M. Suchanek, G. Kasneci, G. Weikum: 2007. YAGO: a Core of Semantic Knowledge. *In WWW*, pages 697–706.

B. Taneva, M. Kacimi, G. Weikum: 2011. Finding Images of Difficult Entities in the Long Tail. *In CIKM*, pages 189–194.

F. L. Wauthier, N. Jojic, M. I. Jordan: 2012. Active Spectral Clustering via Iterative Uncertainty Reduction. *In KDD*, pages 1339–1347.

D. Yan, L. Huang, M. I. Jordan: 2009. Fast approximate spectral clustering. *In KDD*, pages 907–916.

J. Zheng, L. Vilnis, S. Singh, J. D. Choi, A. McCallum: 2013. Dynamic Knowledge-Base Alignment for Coreference Resolution. *In CoNLL*, pages 153–162.