COALA: A Tool for Inter-document Coreference Resolution Evaluation

Bradley M. Andrews[2], James Fan[1], J. William Murdock[1] and Chris Welty[1]

[1]IBM Watson Research Center
19 Skyline Drive, Hawthorne, NY 10532
{fanj,murdockj,welty}@us.ibm.com

[2]University of California, Berkley: School of Information
102 South Hall, Berkeley, CA 94720-4600
brad.andrews@gmail.com

Abstract

A significant obstacle to scientific progress in machine reading is an objective evaluation method. Precision and recall, while for the most part quantitative, are often measured with respect to some "gold standard" or "ground truth" – itself typically a human annotated corpus. For more complex tasks, such as inter-document coreference resolution, or open ended tasks such as machine reading, relying on a ground truth is often (if not always) impractical. Yet a data-driven approach still requires techniques for evaluation. To address this, we present here a new approach to evaluation of linguistic analysis implemented in a tool we have developed called COALA. The approach basically requires establishing a baseline *system* that produces some results, and evaluations are performed by incrementally changing that system and comparing the results manually. In order to reduce the load on the human evaluator, our tool implements basically an intelligent and task-specific "diff" between the two results, allowing the evaluator to focus only on the changes and evaluate them.

Introduction

A significant obstacle to scientific progress in machine reading is an objective evaluation method. Precision and recall, while for the most part quantitative, are often measured with respect to some "gold standard" or "ground truth" – itself typically a human annotated corpus. This approach has well-known problems: the cost of creating the gold standard is often quite high, and as a result there are not enough of them; agreement between human annotators can vary depending on the complexity of the task – it is rarely above 90% and for complex tasks can drop below 50%; because there are so few gold standards, there is not a great deal of understanding of what precision and recall scores mean when a system is used on a different data-set.

For more complex tasks, such as inter-document coreference resolution, establishing or relying on a ground truth is often (if not always) impractical. In order to annotate coreference across documents, for example, a human annotator needs to remember or record what entities occurred in all of the previous documents in order to recognize whether they occur in the document currently being read; this is extremely difficult, time consuming, and failure prone. As a result of the high cost, there is little benchmark data available for inter-document coreference resolution. Most of the data used in coreference resolution work are annotated for an intra-document coreference task only. The lack of benchmark data makes it difficult to compare different inter-document coreference resolution systems. It is even hard to compare the progress of one's own system with its earlier versions.

For open ended tasks such as machine reading the evaluation problem is even worse. It may very well be impossible to establish a ground truth, as e.g. the number of possible relations that may exist in a single sentence, let alone a document or corpus, can defy rigorous human treatment *a-priori*. For example, in the sentence:

"Chris landed in Paris, France."

if a human annotator were asked to capture all the relations in this sentence, we might expect *"chris landedIn ParisFrance"*, and *"chris landedIn France",* and perhaps even *"Chris arrivedIn ParisFrance."* But what about *"Chris arrivedIn Europe"*, or something more formal like *(or "Chris arrivedIn OrlyAirport" "Chris arrivedIn DeGaulleAirport")*, an inference possibly drawn from the base extraction and some world knowledge about the airports in Paris, France. This is a variation of the well-known dictionary game; a person can potentially go on for hours expanding on the meaning of a single sentence.

Clearly for open-ended tasks like machine reading (MR) and especially in the case of MR systems that use inference and background knowledge, a ground truth is impossible to achieve, and would in fact be detrimental (by not recognizing such correct inferences as correct). Yet a data-driven approach still requires techniques for evaluation. To address this, we present here a new approach to evaluation of linguistic analysis implemented in a tool we have developed called COALA (COreference ALgorithm Analyzer).

In a nutshell, the approach requires establishing a baseline *system* that produces some results, and evaluations are performed by incrementally changing that system and comparing the results manually. Evaluations are therefore comparative rather than absolute, e.g. adding a dictionary

lookup to some system resulted in a 15% relative increase in precision. The results will always be relative, since we don't require the baseline system to be evaluated. We don't consider this a problem, since we believe so-called absolute precision and recall measures based on a gold standard are, in fact, relative to the corpus, the linguistic style of the corpus (e.g. news, blogs, email, fiction, etc), the task (e.g. NE detection, relation detection, coreference analysis, MR, etc.), and the inter-annotator agreement. And furthermore, we don't know of a better way to evaluate results in these areas.

The technique still requires manual evaluation of the results, and our tool, COALA, is designed to reduce and simplify this load. COALA basically implements an intelligent and task-specific "diff" between the two results, allowing the evaluator to focus only on the changes and evaluate them. In the simple sentence example above, the evaluator *judges the knowledge produced* by a system with some new capability (like an added background KB) rather than having to imagine all the correct knowledge that could possibly be extracted from a sentence.

Thus far in our work we have only used COALA to evaluate progress in coreference resolution. Coreference resolution is an important part of machine reading in its own right, and we believe the COALA approach is suitable for evaluating the wider array of MR techniques as well.

In this paper, we first describe our general research agenda to combine scalable language technologies with semantic web technologies, and some aspects of our work on inter-document coreference resolution in order to set the proper context for our current use of COALA. We then describe the technique used in the tool for comparing results and finally some first results in our evaluation of COALA itself.

The Overall Picture

We have been working on combining large-scale information extraction from text with knowledge-based systems. In doing so, we believe we can address the "knowledge acquisition bottleneck" and extend the application domain of the extracted information beyond simple search and retrieval. This is not a new idea; however our focus is less on theoretical properties of NLP or KR systems in general, and more on the realities of these technologies *today*, and how they can be used together for large scale inputs.

Architecture

We envision a system made of five main components: goal analyzer, information extractor, knowledge integrator, knowledge base and reasoner (see Figure 1).

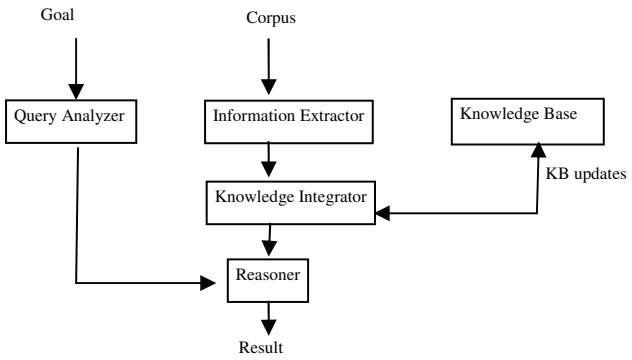

Figure 1: System architecture

Query analyzer

The query analyzer parses a query from the user (e.g., a natural-language question) and converts it into a formal representation to be passed on to the reasoner. There has been a significant amount of work done in this area, and we are using components that consistently rank in the top three at competitions such TREC (Chu-Carroll et al. 2005).

Information Extractor

The information extractor annotates entities and relations found in the corpus, and outputs them. It is made of a collection of individual annotators embedded in the open-source Unstructured Information Management Architecture (UIMA) (Ferrucci and Lally 2004). The annotators overlap to varying degrees in the types of entities and relations they discover, and in the cases of overlap, need to have their results combined. The information extractor creates fragments of knowledge by providing shallow processing, and it scales linearly with the size of the domain corpus.

Knowledge Base

The knowledge base is a formal representation of both the domain specific knowledge and top level background knowledge. The domain knowledge can be obtained from an authoritative source, such as a subject matter expert or a text book. The background knowledge can be obtained from sources such as the Component Library (Barker, Porter and Clark 2001), CYC, FrameNet and WordNet.

Knowledge Integrator

The knowledge integrator may find implicit relations between new and prior knowledge; it may link knowledge fragments produced by the information extractor, and it may elaborate, confirm or invalidate prior knowledge. The outputs can be used as updates to the knowledge base, and they can also be used by the reasoner to produce results for given goals. A key functionality of the knowledge integrator is finding the relation between two knowledge fragments, and coreference (both inter-document and intra-document) resolution is essential for solving this task.

Reasoner

The reasoner takes in the formal representation of a query and the extracted knowledge integrated with an existing knowledge base, and it produces a final result. The key challenge in creating a reasoner suitable for our system is that the reasoner must tolerate imperfections, such as knowledge gaps and inconsistencies. Most of the existing logic-based reasoners require precise inputs, and are unlikely to return anything meaningful after encountering imperfect knowledge.

Applications

We believe machine reading systems, specifically unsupervised text extraction, can benefit from knowledge integration and inference. MR results linked together through resources like WordNet and augmented with background knowledge like geography, political leaders, etc., can significantly increase the utility of MR over the MR results alone in applications such as:

- Question answering: Unlike most of the current question answering systems, answers distributed across multiple documents can be easily found and retrieved because of the integrated knowledge. In addition, the knowledge base and the reasoner provide the facility to compose and infer answers not explicitly stated.
- Search: Current search engines operate on the assumption that there exists a single document that contains all the information needed for a particular search. With the assistance of the knowledge integrator, this assumption is no longer needed. Information spread across multiple documents can be gathered and presented as the result of a search along with the explanation of how different pieces are related.
- Summarization: Current text summarization techniques focus on the most relevant words or sentences in the text. Using our system, we can analyze the relation between the most relevant words and the rest of the corpus, and produce synthesized summarizations that capture the content of the corpus more naturally and more thoroughly.

It is important to note, however, that our claims for this paper are not contingent on our approach to information extraction and knowledge integration and their connection to MR, rather we are presenting this material as background to understanding how we evaluated the COALA tool.

The Challenge of Efficient Evaluation of Inter-document Coreference Resolution

As described in the previous section, the knowledge integrator plays a key role in the proposed system, and coreference resolution is essential to effective knowledge integration. However, it is challenging to develop an effective coreference resolution system, especially for inter-document coreference. One of the main obstacles is the high cost associated with inter-document coreference resolution evaluation. It is easy for a developer to spot a particular flaw in a set of coreference results, but it is difficult to evaluate how a system change that fixes that flaw impacts the results overall. Often we find that correcting one faulty coreference breaks others.

Coreference Resolution

A coreference occurs when two textual entities refer to the same conceptual instance. Coreference may occur within a document, such as the following example.

> *Scarlett Johansson* was born in … *She* has a sister named …

The pronoun *she* refers to the same person as *Scarlett Johansson*. Coreference may also occur across multiple documents such as the following example.

> Document 1: *President George W. Bush* visited …
>
> Document 2: *The 43rd president of the United States* gave a speech …

The term *President George W. Bush* in the first document refers to the same person as the term *the 43rd president of the United States* in the second term.

Coreference resolution is the task of identifying anaphoric textual entities (*she* and *the 43rd president of the United States*) and their antecedents (*Scarlett Johansson* and *President George W. Bush*). Inter-document coreference resolution is important because it erases the document boundary for a variety of tasks. It allows users to gather information about an entity across multiple documents. However, inter-document coreference resolution is very different from intra-document coreference resolution as many of the intra-document coreference resolution techniques rely on consistent content and discourse contextual features, none of which is applicable in inter-document coreference resolution.

Coreference Resolution Evaluation

Coreference resolution evaluation is not easy because it involves comparing a key made of a set of coreference chains (a list of words that refer to the same entity) with the set of coreference chains from the system being evaluated.

Scoring inter-document coreference has an additional challenge – there is little annotated data available, hence there is no key to compare with for the output of a given system. Annotation for an inter-document coreference task requires an annotator to read through all the documents in a corpus, and relate the content of one document with another. For a corpus made of n documents, there are $O(n^2)$ pairs of documents to read and compare so that all

occurrences of coreference can be annotated. This is more costly than annotating for intra-document coreference task and it does not scale for large corpora.

Related Work

There has been much previous work on coreference resolution systems, but only a few inter-document coreference systems were developed over the years, and none of them addressed the complexity of resolving all occurrences of inter-document coreference. Bagga (Bagga and Baldwin 1998) used a vector space model to disambiguate people names. Given two documents and a name in question, he extracted all the sentences containing the name and its intra-document referents, and used cosine similarity to measure the distance between the two sets of extracted sentences. If the two sets were similar, then an inter-document coreference relation was established. Mann (Mann and Yarowsky 2003) utilized additional features such as date of birth, nationality, etc, in an unsupervised clustering algorithm for name disambiguation. Niu (Niu, Li and Srihari 2004) built a name disambiguation system using maximum entropy model. The "semantic integration" for "robust reading"' (Li, Morie and Roth 2005) had a broader scope. It resolved the coreference of a variety of types, such as people, place, etc. Two methods were used: a pair wise classifier and a global clustering algorithm.

While these systems have focused on the specific problem of inter-document named entity coreference resolution, we are interested in the more general problem of how to resolve all occurrences of inter-document coreference including named or unnamed person, organization, places, objects, events, etc. Welty and Murdock (2006) have shown early results of how such a coreference resolution system can be built using graph matching algorithm over RDF representation of the underlying documents.

Because these previously developed systems were only interested a specific aspect of the inter-document coreference problem, it was possible to hand annotate a corpus. Bagga (Bagga and Baldwin 1998) annotated a small set of 197 articles from New York Times for the evaluation of resolving all references to "John Smith". Artificially generated data were also used for evaluation of these systems. Gooi (Gooi and Allan 2004) compared and evaluated the effectiveness of different statistical methods in the task of cross-document coreference resolution. In addition to the corpus used in (Bagga and Baldwin 1998), Gooi used a technique similar to that of artificial sense tagged corpora creation to create a much large evaluation corpus. He first obtained 10,000 to 50,000 unique documents from the TREC 1, 2 and 3 volumes for the following subjects: art, business, education, government, healthcare, movies, music, politics, religion, science and sports, then he used "person-x" to replace a set of randomly selected names, such as "Peter Guber" in the corpus. Assuming each name in each domain referred to different person, the resulting corpus contained multiple occurrences of "person-x" referring to different entities, and it was used to evaluate the resolution of "person-x" coreference.

Because we are interested in resolving all occurrences of inter-document coreference, annotating a large corpus for inter-document coreference becomes a prohibitively expensive process, and we must seek new evaluation methods without fully annotated corpora.

COALA

COALA (COreference ALgorithm Analyzer) is a "diff"-like tool to compare the results of coreference from two different algorithms. The analysis of coreference techniques relies on the use of annotated document corpus results represented as RDF triples.

To run a new analysis, one needs two separate populated rdf graphs. Graph 1 is the baseline coreference result on a corpus; graph 2 is the new coreference result on the same corpus. The goal of an analysis is to determine what changed in the new graph.

COALA displays differences between graphs by first grouping spans (strings of text with start and end positions) into entity bags and putting related instances and triples from both graphs into entity bags. An entity bag is defined by a set of spans and the associated RDF instances or triples in the two graphs. An entity bag may have more than one span associated with it, but no span can be associated with more than one entity bag in the same analysis. Similarly, one entity bag may contain more than one coreference chain, but no coreference chain may be associated with more than one entity bag.

After COALA creates all of the entity bags, it iterates through every bag to determine if the bag is interesting or not. An "interesting" bag is one where the elements (instances, triples, or spans) diverge between two different graphs. COALA only displays the interesting bags.

For example, consider the following sample documents:

Document 1: Bob Smith met John Doe and then he left.
Document 2: R. Smith is in town.

For these documents, we will consider the following spans:

Span A: (Document 1: 0, 10) // "Bob Smith"
Span B: (Document 1: 14, 21) // "John Doe"
Span C: (Document 1: 30, 32) // "he"
Span D: (Document 2: 0, 8) // "R. Smith"

Also consider the following graphs:

Graph X:
Person x1 has mentions A, C, & D
Person x2 has mention B

Graph Y:
Person y1 has mentions A, C

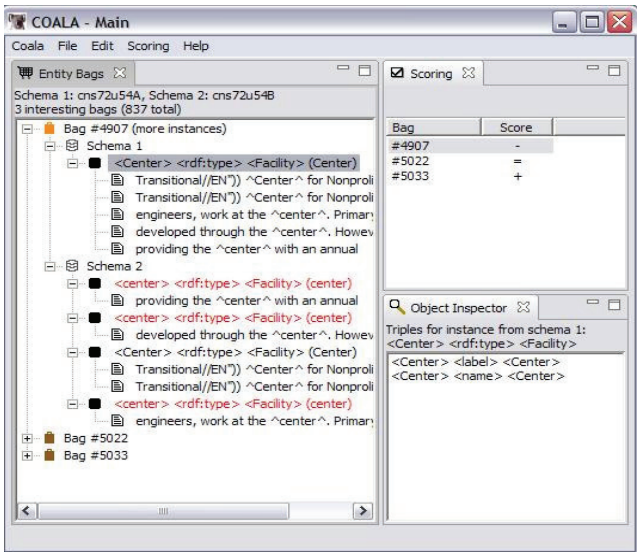

Figure 2: A screen shot of COALA.

Person y2 has mention B
Person y3 has mention D

Graph Z:
Person z1 has mentions A, D
Person z2 has mentions B, C

All three graphs above represent different but reasonable interpretations of the two documents. If we compare graphs X and Y in COALA, we get one interesting bag that contains Person x1, Person y1 (because it shares a span with x1), and Person y3 (because it also shares a span with x1). We also get one uninteresting bag with Person x2 and Person y2; this bag is uninteresting because there is no disagreement about types or spans of the instances.

If we compare graphs X and Z in COALA, we get a single interesting bag containing all four instances in those graphs: Person z1 and Person z2 must be in the same bag as Person x1 because they each share a span with it and Person x2 must be in that bag because it shares a span with Person z2.

The previous example shows how bag membership can cascade: z2 is added to the bag because it has a match with x1 and then x2 is added because it has a match with z2. Bags can grow arbitrarily big this way. This is a key design feature of COALA. As much as possible, we intend for a user of COALA to be able to obtain all of the information needed to understand a particular coreference issue within a single bag; in the case of graph X vs. graph Z, the issue is the antecedent of "he," so we expect that a user will need to examine what each graph claims about each of the potential for antecedents.

For each interesting bag, COALA allows a user to give a score of "+", "-" or "=". A score of "+" indicates the new (graph 2) result has made a positive difference relative to the old (graph 1), a "-" indicates a negative difference, and a "=" indicates no difference. A user may also view the document-level analysis results from the original text document to obtain a better understanding of the context. Figure 2 shows a screen shot of COALA being used to compare two graphs. In this example, the user is examining the different results from graph 1 and graph 2 in the first bag, and the user has given a score of "-" to the difference. Figure 3 shows the user examining the annotations on the original text document through the UIMA annotation viewer [UIMA, 2006], embedded in COALA.

After the user has finished scoring all the interesting bags, he tallies a total score. COALA translates a score of "+" into +1 point, a score of "-" into -1 point, and a score of "=" into 0 points. The average score provides a metric of the relative effectiveness of the two algorithms: if it is positive, then the new graph has made a positive impact compared to the original, and vice versa.

The total score is a measurement of the overall performance of the new graph with respect to the old. It does not distinguish the difference between precision and recall, and it is closer to *F*-measure in that sense.

Evaluation

We believe that COALA facilitates the evaluation of cross-document coreference resolution systems by enabling users to score only the differences between two systems' outputs. The differences represent a small fraction of the total corpus. We evaluated this hypothesis by comparing the number of interesting bags with the total number of bags found.

Experimental setup

The corpus we used is a set of unclassified news abstracts/summaries gathered by the Center for Nonproliferation Studies. There are a total of 50 documents in the corpus, and the average size of documents in this corpus is approximately 2 kilobytes.

The data used in the experiment is described in [Yatskevich et al. 2006]. The first graph represents the results of linguistic coreference. The second graph represents the results of both linguistic coreference and knowledge-based coreference postprocessing. COALA allows us to easily see the effects of the postprocessing algorithm by providing a comparison between the graph that included the results of that algorithm and the one that did not.

Results and analysis

Between the two graphs, COALA found 1,468 entity bags. Out of these entity bags, COALA found 50 of them to be interesting. The result shows that COALA has drastically reduced the efforts required to evaluate the performance of an inter-document coreference resolution system with respect to another. Instead of annotating nearly all the

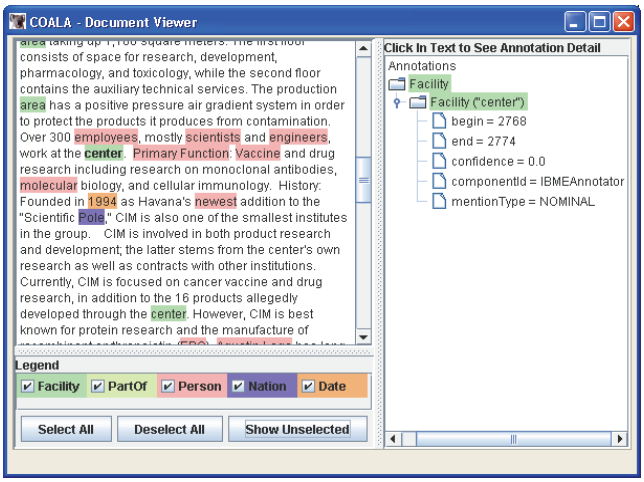

Figure 3: A screen shot of the document viewer.

entity bags, one only needs to judge 3.5% of the overall entity bags. This result is clearly dependent on the magnitude of the change between the two systems that produced the graphs. In reality, the two graphs being compared must overlap in some way for COALA to work effectively. Thus COALA's most useful application is in evaluating incremental changes to a system, such as ablating a particular component or technique, as in this example, or fixing a bug and testing the global impact of the fix.

With the assistance of the document viewer, judging an interesting bag is quite easy and speedy. A more aggressive postprocessor or a radically different linguistic coreference module would have resulted in a much larger percentage. However, these results do suggest that some interesting changes to coreference can be evaluated much more quickly using COALA than they could be using exhaustive manual comparison.

Summary and Future Work

We are interested in combining text extraction technology to populate large scale knowledge bases for a variety of applications, such as question answering, summarization, etc. Inter-document coreference resolution for person, organization, places, objects, events, etc, is an essential part of the effort. One of the main obstacles in the development of inter-document coreference system is the high cost associated with evaluation. In this paper, we present COALA, a tool for evaluating the difference between a new coreference system and an existing one. Our preliminary evaluation has shown that COALA has drastically reduced the effort involved in evaluating two coreference systems.

While we do think that coreference analysis can be a useful part of MR in general, we believe this evaluation technique and a suitably adapted COALA tool can be useful for evaluating and comparing other MR systems and their results. The most obvious and immediate use of COALA is in evaluating incremental changes to one particular technique, however with further study we also believe the basic technique of computing an intelligent "diff" based on the links to the underlying text spans can be useful for comparative evaluation between different MR systems.

References

Bagga, A., and Baldwin, B. 1998. *EntityBased Cross-Document Coreferencing Using the Vector Space Model*. In 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics, 79—85

Barker, K., and Porter, B. and Clark, P. 2001. *A Library of Generic Concepts for Composing Knowledge Bases*. In Proceedings of First International Conference on Knowledge Capture.

Chu-Carroll, J., Czuba, K., Duboue, P. and Prager, J. 2005. *IBM's PIQUANT II in TREC2005*. The Fourteenth Text REtrieval Conference (TREC 2005) Proceedings.

Ferrucci, D. and Lally, A. 2004. *UIMA: an architectural approach to unstructured information processing in the corporate research environment*. Natural Language Engineering 10 (3/4): 327–348.

Gooi, C. and Allan, J. 2004. *Cross-document Coreference on a Large Scale Corpus*. In Proceedings of HLT/NAACL, 2004.

Li, X., Morie, P., Roth, D. 2005. *Semantic Integration in Text: From Ambiguous Names to Identifiable Entities*. AI Magazine 26(1): 45-58

Mann, G. and Yarowsky, D. 2003. *Unsupervised personal name disambiguation*. In CoNLL, Edmonton, Alberta.

Niu, C., Li, W., Srihari, R. *Weakly Supervised Learning for Cross-document Person Name Disambiguation Supported by Information Extraction*. ACL 2004: 597-604

UIMA SDK User's Guide and Reference Version 2. 2006.http://dl.alphaworks.ibm.com/technologies/uima/UIMA_SDK_Users_Guide_Reference_2.0.pdf

Welty, C. and Murdock, J. W. 2006. *Towards Knowledge Acquisition from Information Extraction*. In Proceedings of the 2006 International Semantic Web Conference. Athens, Georgia. November, 2006.

Yatskevich, M., Welty, C. and Murdock, J. W. Coreference resolution on RDF Graphs generated from Information Extraction: first results. 2006. ISWC-06 workshop on Web Content Mining with Human Language Technologies.