Cross-Document Coreference on a Large Scale Corpus

Chung Heong Gooi and James Allan
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
{cgooi,allan}@cs.umass.edu

Abstract

In this paper, we will compare and evaluate the effectiveness of different statistical methods in the task of cross-document coreference resolution. We created entity models for different test sets and compare the following disambiguation and clustering techniques to cluster the entity models in order to create coreference chains:

Incremental Vector Space
KL-Divergence
Agglomerative Vector Space

1. Introduction

Coreference analysis refers to the process of determining whether or not two mentions of entities refer to the same person (Kibble and Deemter, 2000). For example, consider the following short passage of text:

John Smith was appointed chair of the committee. Because of his past experience, Mr. Smith seemed the perfect choice. His good friend John, however, was not considered.

Coreference analysis attempts to decide whether *John Smith* and *Mr. Smith* refer to the same person, and whether *John* is also the same person. The task is often extended to include references such as *his* or even *his good friend*, though we do not explore that extension in this study.

Addressing this problem is important to support systems such as those that search for, extract, and process mentions of "people of interest" in news or transcripts (BBN 2001), or for other information organization tasks that might benefit from precise knowledge of how names occur, such as Topic Detection and Tracking (Allan 2002).

Cross-document coreference analysis pushes the task into considering whether mentions of a name in *different* documents are the same. The problem becomes more complex because documents might come from different sources, will probably have different authors and different writing conventions and styles(Bagga and Baldwin,1998), and may even be in different languages.

There has been little published work on cross-document coreference analysis and that has generally been evaluated on a small corpus of documents. A major contribution of this work is to develop a substantially larger (more than two orders of magnitude) corpus for evaluation. We show that the previous approach is effective but that a variation on it, agglomerative vector space, provides improved and much more stable results.

We begin in Section 2 by describing how cross-document coreference analysis is evaluated. We sketch prior work in Section 3 and describe our two evaluation corpora in Section 4. Section 5 discusses the three algorithms that we explore for this task and then Section 6 describes our experimental results on both corpora. In Section 7 we provide some additional analysis that attempts to explain some surprising results. We conclude in Section 8 with a description of our plans for future work.

2. Evaluation

Given a collection of named entities from documents, the coreferencing task is to put them into equivalence classes, where every mention in the same class refers to the same entity (person, location, organization, and so on). The classes are referred to as "coreference chains" because the entities are chained together.

To evaluate the coreference chains emitted by a system, we need truth data: the chains of entities that are *actually* referring to the same person. Evaluation then proceeds by comparing the true chains to the system's hypothesized chains.

We use the B-CUBED scoring algorithm (Bagga and Baldwin 1998) because it is the one used in the published research. The algorithm works as follows.

For each entity mention *e* in the evaluation set, we first locate the truth chain *TC* that contains that mention (it can be in only one truth chain) and the system's hypothesized chain *HC* that contains it (again, there can

be only one hypothesis chain). We then compute a precision and recall score for those two chains. Precision is the proportion of mentions in *HC* that are also in *TC* and recall is the proportion of mentions in *TC* that are also in *HC*. If the chains match perfectly, recall and precision will both be one. If the hypothesis chain contains only the single mention *e*, then its precision will be one, and its recall will be $1/|TC|$, the inverse of the size of the truth chain. Note that it is not possible to have a precision or recall of zero since entity *e* is always in common between the two chains. Our implementation of the B-CUBED algorithm is used specifically to evaluate an existing set of coreference chains and does not utilize any smoothing to handle system output which contains no entities.

Overall precision and recall values are determined by averaging the individual values over all mentions in the evaluation set. These are the primary evaluation measures for cross-document coreference analysis.

3. Related Work

TIPSTER Phase III first identified cross-document coreference as an area for research since it is a central tool to drive the process of producing summaries from multiple documents and for information fusion (Bagga and Baldwin, 1998). The Sixth Message Understanding Conference (MUC-6) identified cross-document coreference as a potential task but it was not included because it was considered to be too difficult (Bagga and Baldwin, 1998).

ISOQuest's NetOwl and IBM's Textract attempted to determine whether multiple named entities refer to the same entity but neither had the ability to distinguish different entities with the same name. Entity detection and tracking looks at the same tasks as cross document coreferencing.

Much of the work in this study is based on that by Bagga and Baldwin (1998), where they presented a successful cross-document coreference resolution algorithm to resolve ambiguities between people having the same name using the vector space model. We have implemented a simplified version of their algorithm that achieves roughly equivalent accuracy, but will show that the algorithm does not work as well when translated to a substantially larger corpus of documents.

There has been significant work recently in the information extraction community on a problem known as Entity Detection and Tracking within the Automatic Content Extraction (ACE) evaluations (NIST 2003). That work includes an optional sub-task referred to alternately as either Entity Tracking or Entity Mention Detection. The goal is to pull together all mentions of the same entity across multiple documents. This task is a small and optional part of the complete ACE evaluation and results from it do not appear to be published.

4. Corpora

To evaluate the effectiveness of our various techniques for cross document coreference, we use the same "John Smith" corpus created by Bagga and Baldwin (1998). In addition, we created a larger, richer and highly ambiguous corpus that we call the "Person-x corpus."

4.1 John Smith Corpus

Bagga and Baldwin tested their coreference algorithm against a set of 197 articles from 1996 and 1997 editions of the New York Times, all of which refer to "John Smith". All articles either contain the name John Smith or some variation with a middle name or initial. There are 35 different John Smiths mentioned in the articles. Of these, 24 refer to a unique John Smith entity which is not mentioned in the other 173 articles (197 minus 24).

We present results on this corpus for comparison with past work, to show that our approximation of those algorithms is approximately as effective as the originals. The corpus also permits us to show how our additional algorithms compare on that data. However, our primarily evaluation corpus is the larger corpus that we now discuss.

4.2 Person-x Corpus

Since the task of annotating documents is time consuming, we used a different technique to create a large test set with different entities of the same name. The technique used to construct this corpus is similar to the well known technique of creating artificial sense tagged corpora. Artificial sense tagged corpora is used to evaluate word sense disambiguation algorithms and is created by adding ambiguity to a corpus. Similarly, we consider the task of coreferencing multiple occurrences of "John Smith" to be similar to coreferencing multiple occurrences of "person-x", where the occurrences of "person-x" are a disguise of multiple named entities such as "George Bush" and "Saddam Hussein". This approach simplifies the task of looking for a large collection of "John Smith" articles and obtaining the actual coreference links between the many "John Smith" entities. It also allows us to create a vastly larger corpus of documents mentioning the "same person."

We first obtained from 10,000 to 50,000 unique documents from the TREC 1, 2 and 3 volumes using the Inquery search engine from UMass Amherst for each of the following subjects: art, business, education, government, healthcare, movies, music, politics,

religion, science and sports. Then, we ran the documents through Identifinder, a named entity extraction system developed by BBN, to tag the named entities in the documents.

Next, we selected one person entity randomly from each document. Since Identifinder occasionally tags an entity incorrectly, we manually went through each selection to filter out entities that were not people's names. We also manually filter out cases where the tagged entity is only one word (e.g., John, Alex, etc.).

We replaced the occurrences of the selected entity in each document with "person-x" as follows:

In the late 1970s, the company hired producers <enamex type="person">jon peters</enamex> and <enamex type="person">**peter guber** </enamex> to start a studio from scratch.

In the late 1970s, the company hired producers <enamex type="person">jon peters</enamex> and <enamex type="person"> **person-x** </enamex> to start a studio from scratch.

We also replaced all additional occurrences of the same name and names that matched (except for a middle initial or name) in that document with "person-x". For example, in the case above, other occurrences of *Peter Guber* or names such as *Peter X. Guber* would be replaced by "person-x".

We now have a large set of documents containing a reference to "Person X" and we know for each document "which" person entity it is referring to. We actually verified that names of the same name were the same entity, though with the large number of entities, the task was potentially overwhelming. However, since the entities are categorized according to domain (by the query that found the document), determining the actual coreference links becomes significantly easier. In an article discussing sports, the multiple occurrences of the name "Michael Chang" are most likely to be pointing to the tennis player—and the same tennis player.

These mappings from "Person X" to their true names will serve as our evaluation/true coreference chains. Since we know the name that "Person X" replaced, we assume that if those names are identical, they refer to the same person. So all references to "Person X" that correspond to, say, "Bill Clinton," will be put into the same coreference chain.

We manually removed documents whose Person X entity pointed to a different person than the person in its corresponding chain. Consider the scenario where we have four documents, three of which contains Person X entities pointing to John Smith (president of General Electric Corporation) and the other pointing to John Smith (the character in Pocahontas). The last John Smith document will be removed from the chain and the entire corpus. The final Person X corpus contains 34,404 unique documents. Hence, there are 34,404 "Person X"s in the corpus and they point to 14,767 different actual entities. 15.24% of the entities occur in more than one domain subject.

Number of occurrences	Percentage of entities
1	46.66
2	18.78
3	9.03
4	4.55
5	1.86
6	1.16
7	0.83
8	0.46
9 or more	16.67

Table 1: Breakdown of distribution by number of occurrences within the Person X corpus.

Table 1 displays the distribution of entities versus their occurrences in our corpus. Slightly over 46% of entities occur only once in the collection of 34,404 entities. That compares to about 12% in the John Smith corpus. Of the total of 315,415 unique entities that Identifinder recognized in the entire corpus, just under 49% occurred precisely once, so our sample appears to be representative of the larger corpus even if it does not represent how "John Smith" appears.

A potential shortcoming that was noted is that variation of names such as "Bob Matthews" versus "Robert Matthews" may have been missed during the construction of this corpus. However, this problem did not show up in a set entities randomly sampled for analysis.

5. Methodology

In all cases, we represented the mention of an entity (i.e., an occurrence of "John Smith" or "Person-x" depending on the corpus used) by the words around all occurrences of the entity in a document. Based on exploratory work on training data, we choose a window of 55 words centered on each mention, merged those, and called the result a "snippet." (In many cases the snippet incorporates text from only a single occurrence of the entity, but there are documents that contain 2-3 "person-x" instances, and those are merged together.) We then employ three different methods for comparing snippets to determine whether their corresponding mentions are or are not to the same entity. In the remainder of this section, we describe the three

methods: incremental vector space, KL divergence, and agglomerative vector space.

5.1. Incremental vector space

Our intent with the incremental vector space model is to approximate the work reported by Bagga and Baldwin (1998). Their system takes as input properly formatted documents and uses the University of Pennsylvania's CAMP system to perform *within*-document coreference resolution, doing more careful work to find additional mentions of the entity in the document. It then extracts all sentences that are relevant for each entity of interest based on the within-document coreference chains produced by CAMP. The sentences extracted form a summary that represents the entity (in contrast to our 55-word snippets). The system then computes the similarity of that summary with each of the other summaries using the vector space model. If the similarity computed is above a predefined threshold, then the two summaries are considered to be coreferent.

Each of the summaries was stored as a vector of terms. The similarity between two summaries S_1 and S_2 is computed as by the cosine of the angle between their corresponding vectors. Terms are weighted by a tf-idf weight as $tf \cdot log(N/df)$, where tf is the number of times that a term occurs in the summary, N is the total number of documents in the collection, and df is the number of documents that contain the term.

Because we did not have the same within-document coreference tools, we opted for a simpler variation on Bagga and Baldwin's approach. In our implementation, we represent *snippets* (combined 55-word sections of text) as vectors and use this model to represent each entity. We calculated term weights and similarity in the same way, however. The only difference is the text used to represent each mention.

For both cases, the system operates incrementally on the list of entities as follows. We first create one coreference chain containing a single entity mention (one vector). We then take the next entity vector and compare it against the entity in the link. If the two vectors have a similarity above a pre-defined threshold, then they are regarded to be referring to the same entity and the latter will be added into the same chain. Otherwise, a new coreference link is created for the entity.

We continue creating links using this incremental approach until all of the entities have been clustered into a chain. At each step, a new entity is compared against all existing coreference chains and is added into the chain with the highest average similarity if it is above the predefined threshold. Our implementation differs from that of Bagga and Baldwin in the following ways:

Bagga and Baldwin use a single-link technique to compare an entity with the entities in a coreference chain. This means they include an entity into a chain as soon as they find one pair-wise entity to entity comparison that is above the predefined threshold. We utilize an average-link comparison and compared an entity to each other entity in a coreference chain, then used the average similarity to determine if the entity should be included into the chain.

They utilized the CAMP system developed by the University of Pennsylvania to resolve within document coreferencing and extract a summary for each entity. In our system, we simply extract the snippets for each entity and do not depend on within document coreferencing of an entity.

5.2 KL Divergence

The second technique that we implemented for entity disambiguation was based on Kullback-Leibler Divergence. For this technique, we represent the snippets in the form of a probability distribution of words, creating a so-called entity language model (Allan and Raghavan, 2002). The KL divergence is a classic measure of the "distance" between two probability distributions. The more dissimilar the distributions are, the higher the KL divergence. It is given by the equation:

$$D(q \,||\, r) = \sum_x q(x) \log \frac{q(x)}{r(x)}$$

where x ranges over the entire vocabulary. The smaller the distance calculated by KL divergence, the more similar a document is with another. If the distance is 0, then the two distributions are identical. To deal with zero probabilities, we need some type of smoothing, and we chose to use the asymmetric skew divergence, mixing one distribution with the other as determined by a α (Lee,2001): $D(r \,||\, \alpha q + (1 - \alpha)r)$

Skew divergence best approximates KL divergence when the parameter α is set to a value close to 1. In our experiments, we let α=0.9

We used the incremental approach of Section 5.1, but with probability distributions. Each of the distributions created (from a snippet) was evaluated against the distributions for existing coreference chains. Smaller distances computed through skew divergence indicate that the entity is similar to the entities in the chain. If the distance computed is smaller than a predefined threshold, then the new entity is added into the coreference chain and the probabilistic distribution of the coreference chain's model is updated accordingly. We start with one entity in one coreference chain and continue comparing, inserting, and creating coreference chains until all of the entities have been resolved.

Note that the KL divergence approach is modeled directly after the incremental vector space approach. The difference is that the vector is replaced by a probability distribution and the comparison uses divergence rather than cosine similarity.

5.3 Agglomerative vector space

In our explorations with the previous algorithm, we noticed that if early coreference chains contained misplaced entities, those entities attracted other entities with high similarity and "polluted" the coreference chain with entities that are not part of the truth chain. We therefore switched to an agglomerative approach that builds up the clusters in a way that is order independent. This approach is typically known as bottom-up agglomerative clustering. It is also done in the vector space model, so we again represent the snippets by vectors.

We first create a coreference chain containing one entity for every entity to be resolved. For each coreference chain, we then find its nearest neighbor by computing the similarity of the chain against all other chains using the technique described above in Section 5.1. If the highest similarity computed is above a predefined threshold, then we merge those two chains together. If any merging was performed in this iteration, we repeat the whole process of looking for the most similar pair and merging then in the next iteration. We continue this until no more merging is done—i.e., the highest similarity is below the threshold.

The only difference between this approach and that in the previous section is that the agglomerative technique requires more comparisons and takes more time. On the other hand, it minimizes problems caused by a single spurious match and it is order independent.

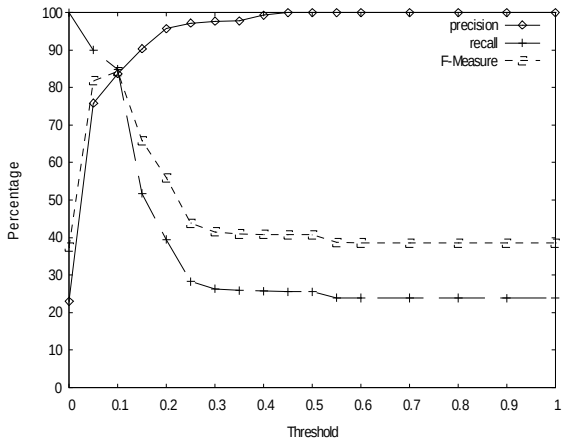

Figure 1: Results of cross-document coreferencing on the John Smith corpus using the incremental vector space approach.

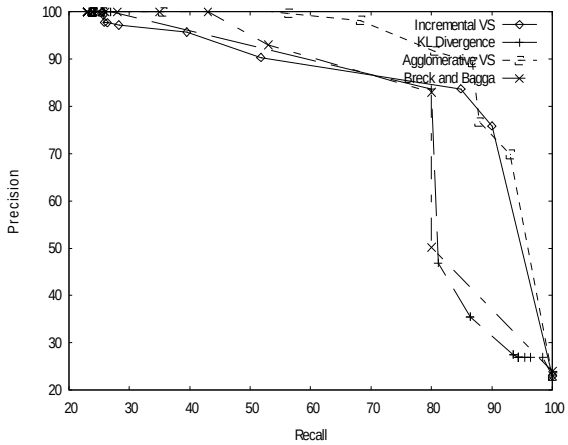

Figure 2: Recall and precision tradeoff of three algorithms on the John Smith Corpus. Results from Baldwin and Bagga (1998) are estimated and overlaid onto the graph.

6. Experiments and Results

To evaluate our various techniques for the task of cross-document coreferencing, we used the two test corpora mentioned in Section 4 and the three coreference approaches described in Section 5. The coreference chains are then evaluated using the B-CUBED algorithm to measure precision and recall as described in Section 2. We present the results by corpus.

6.1 John Smith Corpus Results

Our main goal for the John Smith corpus is to demonstrate that we have successfully approximated the algorithm of Bagga and Baldwin (1998). Figure 1 shows how recall and precision trade off against each other as the decision threshold (should a name be put into a chain) varies in the incremental vector space approach. This graph is nearly identical to the tradeoff curves shown by Bagga and Baldwin, so we believe our variation on their approach is sufficiently accurate to draw conclusions. A key point to note about the graph is that although there is an excellent recall/precision tradeoff point, the results are not stable around that threshold. If the threshold is shifted slightly higher, recall plummets; if it is lowered slightly, precision drops off rapidly.

Figure 2 provides an alternative view of the same information, and overlays the other algorithms on it. In this case we show a recall/precision tradeoff curve. Again, in all cases the tradeoff drops off rapidly, though the agglomerative vector space approach takes longer to fall from high accuracy.

Figure 3 provides another comparison of the three approaches by highlighting how the F-measure varies

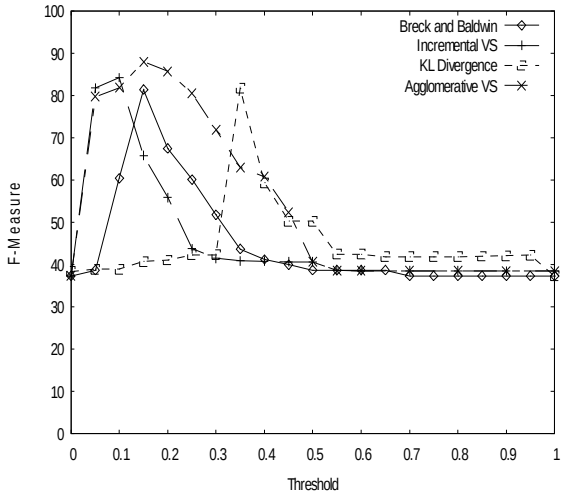

Figure 3: Comparison of F-Measure on the John Smith Corpus.

with the threshold. Note that the agglomerative vector space approach has the highest measure and has a substantially less "pointed" curve: it is much less sensitive to threshold selection and therefore more stable.

The agglomerative vector space achieved a peak F measure of 88.2% in comparison to the incremental approach that peaked at 84.3% (comparable to Bagga and Baldwin's reported 84.6%). We also created a single-link version of our incremental algorithm. It achieved a peak F measure of only 81.4%, showing the advance of average link (when compared to our approach) and the advantage of using within-document coreference to find related sentences (when compared to their work).

6.2 Person-x Results

We next evaluated the same three algorithms on the much larger Person X corpus. The recall/precision graph in Figure 4, when compared to that in Figure 2, clearly demonstrates that the larger corpus has made the task much harder. However, the agglomerative vector space approach has been impacted the least and maintains excellent performance.

Figure 5 shows the F-measure graph. In comparison to Figure 3, all of the techniques are less sensitive to threshold selection, but the two vector space approaches are less sensitive than the KL divergence approach. It is unclear why this is, though may reflect problems with using the skewed divergence for smoothing.

7. Further Exploration

We conducted additional analysis to explore the issues surrounding cross-document coreferencing. We ran experiments with the John Smith corpus to explore

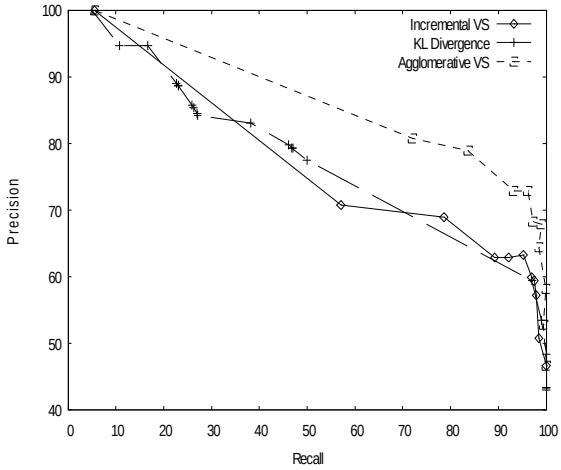

Figure 4: Recall and precision tradeoff for the person-x corpus.

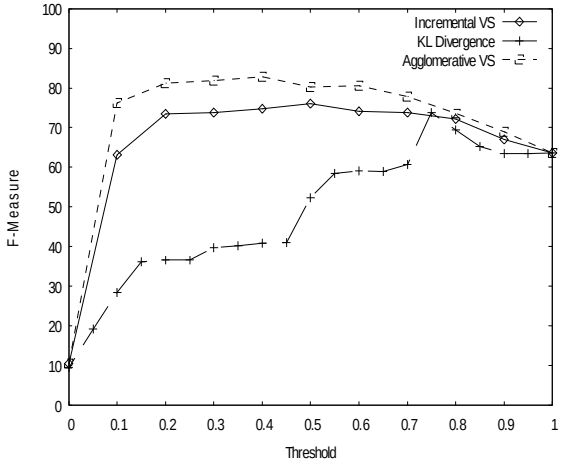

Figure 5: Comparison of F-Measure on the Person-x Corpus.

the question of the effectiveness of a model based on the amount of text used to represent an entity.

7.1 Window size and recall/precision

Allan and Raghavan (2002) showed that the size of the snippets correlates positively with the "clarity" (non-ambiguity) of the model. As the size of the snippet increases, the ambiguity of the model increases, presumably because it is more likely to include extraneous information from the surrounding text.

In our experiment with the John Smith corpus, we used the incremental vector space approach with a threshold of 0.1 and evaluated precision/recall using various window sizes for the snippets. Figure 6 shows the variation. We discovered that the F-Measure peaks at 84.3% with a window size of 55 words. This is the window size that we used for all of our other experiments.

7.2 Domain-specific sub-corpora

The person-x corpus may appear to be biased due to the manner of its construction. Since the documents were selected by subject, one may argue that the task of clustering entities will be much easier if the entities are clearly from different genres. However, if this is true, then it may account for about 85% of the entities in the person-x corpus that occur only in one domain subject. We hypothesized that coreferencing entities in the same genre domain can be considered to be harder in terms of achieving high precision because the consistency of the contents between documents in the same genre domain makes it significantly harder to create a unique model for each entity to aid the task of distinguishing one entity from another.

In order to see how our techniques measure up against this, we reevaluated the effectiveness of our methods of cross-document coreference resolution on a modified version of the person-x corpus. We clustered the documents into their original genre domain (recall that they were created using simple information retrieval queries). Then, we evaluated the precision/recall for each of the clusters and averaged the results to obtain a final precision/recall score. This eliminates the potential bias that clustering the entities becomes easier if the entities are clearly from different genres. Hypothetically, it also makes the task of cross-document coreferencing more challenging than in reality when performed on actual corpora that is not clustered according to genre. Table 2 shows the breakdown of documents and entities in each genre.

The results of the experiments show that clustering documents by their domain specific attributes such as domain genre will hurt cross-document coreferencing. The highest F-Measure that was achieved with agglomerative vector space dropped 6% to 77% and incremental dropped a similar 5%. The KL divergence approach, on the other hand, showed a modest increase of 3% to 77%, equaling the agglomerative approach. The reason for this may be because KL divergence relies more on the global property of the corpus and this approach is more effective when the nearest neighbor computation is degraded by the consistency of the word distributions between documents in the same genre domain.

Genre	Number of Documents	Number of person-x entities
Art	3346	1455
Business	315	182
Education	6177	2351
Government	3374	945
Healthcare	914	405
Movies	677	2292
Music	976	366
Politics	4298	949
Religion	2699	1030
Science	7211	2783
Sports	4417	2009

Table 2: Breakdown of document and entity distribution in the domain subject specific clusters.

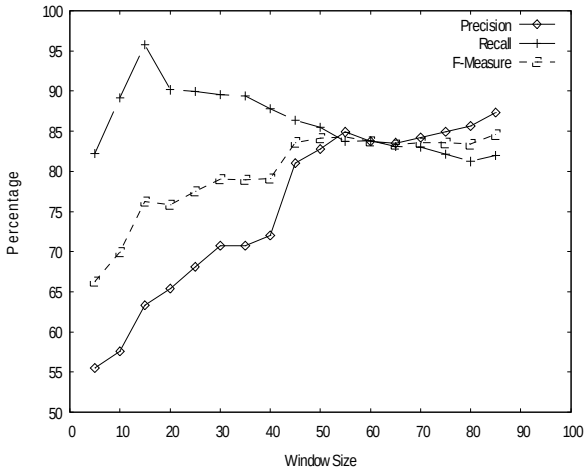

Figure 6: Relationship between the window size of the snippet and recall/precision on the John Smith corpus.

7.3 Runtime comparison.

An important observation in our comparison among the algorithms is running time. While we have shown that the agglomerative vector space approach produced the best results in our experiments, it is also important to note that it was noticeably slower. The estimated running time for the agglomerative vector space experiment on the large corpus was approximately 3 times longer than that of the incremental vector space and KL-Divergence. The runtimes of incremental approaches are linear whereas the runtime of our agglomerative vector space approach is $O(n^2)$.

Is the improvement in our results worth the difference in runtime? The noticeable run time difference in our experiment is caused by the need to cluster a large number of Person-x entities (34,404 entities). In reality, it would be rare to find such a large number of entities across documents with the same name. In the analysis of our reasonably large corpus, less than 16% of entities occur more than 10 times. If the mean size of entities to be disambiguated is relatively small, then there will not be a significant degrade in runtime on the agglomerative approach. Thus,

our conclusion is that the tradeoff between coreference quality versus runtime in our agglomerative approach is definitely worthwhile if the number of same-named entities to be disambiguated is relatively small.

8. Conclusion and Future Work

We were able to compare and contrast our results directly with previous work of Bagga and Baldwin by using the same corpus and evaluation technique. In order to perform a careful excursion into the limited work on cross document coreferencing, we deployed different information retrieval techniques for entity disambiguation and clustering. In our experiments, we have shown that the agglomerative vector space clustering algorithm consistently yields better precision and recall throughout most of the tests. It outperforms the incremental vector space disambiguation model and is much more stable with respect to the decision threshold. Both vector space approaches outperform KL divergence except when the entities to be clustered belong to the same genre.

We are pleased that our snippet approach worked well on the task of cross document coreferencing since it was easier than running a *within* document coreference analyzer first. It was also interesting to discover that previous techniques that worked well on a smaller corpus did not show the same promising recall and precision tradeoff on a larger corpus.

We are interested in continuing these evaluations in two ways. First, colleagues of ours are working on a more realistic corpus that is not just large but also contains a much richer set of marked up entities. We look forward to trying out techniques on that data when it is available. Second, we intend to extend our work to include new comparison and clustering approaches. It appears that sentence-based snippets and within-document coreference information may provide a small gain. And the subject information has apparently value in some cases, so we hope to determine how to use the information more broadly.

Acknowledgements

The John Smith corpus was provided by Breck Baldwin, and we are grateful to him for digging through his archives to find the data. This work was supported in part by the Center for Intelligent Information Retrieval, in part by SPAWARSYSCEN-SD grant number N66001-02-1-8903 and in part by Advanced Research and Development Activity under contract number MDA904-01-C-0984. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

References

Allan, James, ed. *Topic Detection and Tracking: Event-based Information Organization*, Kluwer Academic Publishers, 2002.

Allan, James and Raghavan, Hema. Entity Models: Construction and Application. Center for Intelligent Information Retrieval, Department of Computer Science, University of Massachusetts, 2002.

Bagga, Amit and Breck Baldwin. Algorithms for Scoring Coreference Chains. In Proceedings of the Linguistic Coreference Workshop at The First International Conference on Language Resources and Evaluation (LREC'98), pp563-566, 1998.

Bagga, Amit and Breck Baldwin. Entity-Based Cross-Document Coreferencing Using the Vector Space Model. Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics (COLING-ACL'98),pp.79-85, 1998.

Bagga, Amit and Breck Baldwin. Coreference as the Foundations for Link Analysis Over Free Text Databases. In Proceedings of the *COLING-ACL'98 Content Visualization and Intermedia Representations Workshop* (CVIR'98), pp. 19-24, 1998.

Bagga, Amit, and Biermann, Alan. A Methodology for Cross-Document Coreference. In Proceedings of the *Fifth Joint Conference on Information Sciences* (JCIS 2000), pp. 207-210, 2000.

Bagga, Amit and Baldwin, Breck. How Much Processing Is Required for Cross-Document Coreference? In Proceedings of the Linguistic Coreference Workshop (LREC'98), pp. 563-566, 1998.

BBN Technologies. Rough 'n' Ready™: Audio Indexing for Meetings and News. http://www.bbn.com/-speech/roughnready.html (2001)

Kibble, Rodger and Kees, van Deemter. Coreference Annotation: Whither? Proceedings of LREC2000, Athens, pp. 1281-1286, 2000.

Lee, Lillian. On the Effectiveness of the Skew Divergence for Statistical Language Analysis, Technical Report, Department of Computer Science, Cornell University, 2001.