Cross-Lingual Cross-Document Coreference with Entity Linking

Sean Monahan, John Lehmann, Timothy Nyberg, Jesse Plymale, and Arnold Jung
Language Computer Corporation
2435 North Central Expressway
Richardson, TX, USA
sean@languagecomputer.com

Abstract

This paper describes our approach to the 2011 Text Analysis Conference (TAC) Knowledge Base Population (KBP) cross-lingual entity linking problem. We recast the problem of entity linking as one of cross-document entity coreference. We compare an approach where deductive entity linking informs cross-document coreference to an inductive approach where coreference and linking judgements are mutually beneficial. We also describe our approach to cross-lingual entity linking comparing a native linking approach with an approach utilizing machine translation. Our results show that inductive linking to a native language knowledge base offers the best performance.

1 Introduction

Entity linking is the task of associating entity mentions in text with entries in a knowledge base (KB). For example, when seeing the text "movie star Tom Cruise", the text "Tom Cruise" should be linked the Wikipedia page http://en.wikipedia.org/wiki/Tom_cruise. This is useful because it enables the automatic population of a KB with new facts about that entity extracted from the text. Conversely, existing information stored in the KB can be used to aid in more accurate text extraction. Correlation of entities between documents also benefits other cross-document natural language processing tasks like question answering and event coreference.

Entity linking is challenging for three primary reasons. First, names are often *polysemous* in that they are shared by different entities. Given a name in text, it must be disambiguated among the possible meanings. Wikipedia contains over 100 people with the name "John Williams".

Second, entities are often characterized by *synonymy*, being referred to by different name variants or aliases. Recognizing all instances or mentions of an entity in text requires identifying all of its variants. Both "Cassius Clay" and "Muhammad Ali" refer to the same entity.

A third problem is identifying when an entity mentioned in text is not contained in the KB at all. Such a reference is said to be a *NIL mention*. Detecting NIL mentions is important not only to avoid creating spurious links, but also for identifying new candidates for addition to the KB. As many people as there are in Wikipedia, there are billions that are not.

To create new KB entries, a system also needs to correctly generate links between the co-referring NIL entities. This would enable not only the automatic growth of a KB in terms of knowledge about known entities, but also in terms of previously unknown entities. This extension to the base problem has been described as *entity linking with NIL clustering*.

Entity linking with NIL clustering can be recast as a cross-document coreference approach where the cross-document and linking components are mutually beneficial. In both approaches, the challenges of polysemy and synonymy must be resolved. The difference is that entity linking uses a set of pre-existing identifiers supplied by the KB, thus facilitating integration of different knowledge stores. In cross-document coreference the identifiers created are implied by the cluster membership and are relative to the corpus.

We take an *inductive* approach which treats the problem as cross-document coreference with entity linking. Rather than only clustering the detected NIL mentions, we cluster all entities while using output from our entity linker as suggestions but not fact. This is counter to the deductive approach which first links all of the entities and then clusters the remaining NIL mentions. The inductive approach is illustrated in Figure 1.

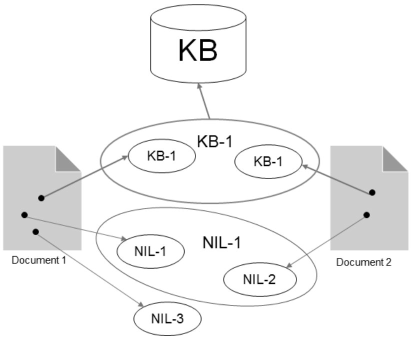

Figure 1: Inductive Entity Linking

In doing so, we effectively use clustering to improve our entity linker performance and attain a better end-to-end score. The difference between these two approaches is described in Algorithms 1 and 2.

Algorithm 1 Deductive Approach

```
1. Link each entity mention to KB or assign NIL.
2. Cluster NIL mentions.
3. Assign each NIL cluster a unique NIL id.
```

Algorithm 2 Inductive Approach

```
1. Link each entity mention to KB or assign NIL.
2. Cluster ALL mentions with links as features.
3. Vote in each cluster to assign KB id or NIL.
```

We demonstrate our inductive approach for the TAC 2011 Knowledge Base Population (KBP) Entity Linking evaluation. In 2011, the entity linking task gained the additional requirement of NIL clustering. We participated both in the English monolingual as well as the English-Chinese cross-lingual tasks using a system which is largely language-independent. In the cross-lingual task, entity mentions from Chinese documents must also be mapped back to an English KB. We also report improvements to our entity linking system originally used in 2010 and show how those enhancements affected our end-to-end score.

2 Related Work

Over the past few years, TAC's Knowledge Base Population task has been at the forefront of development in the area of entity linking. State-of-the-art approaches have recently been summarized by Ji and Grishman (2011). Several entity linking efforts preceded TAC, and used Wikipedia as a KB as well. Cucerzan (2007) formed an extensive mapping of surface text to Wikipedia pages and used it to maximize agreement between context and candidates being disambiguated. Milne and Witten (2008) used Wikipedia concepts as context terms to cross-link documents with Wikipedia articles. Lehmann et al. (2010) utilized a similar contextual model along with a number of other features in a system which achieved top entity linking performance at TAC 2010 KBP.

Our approach to cross-document coreference was shaped in part by the challenge of implementing supervised learning with highly imbalanced data sets.[1] A variety of techniques including under-sampling negative examples and over-sampling positive examples have been proposed to handle skewed distributions, e.g. Akbani et al. (2004). We chose to implement supervised learning over tractable subsets of mentions—in this case, we limited supervised learning to pairs of mentions that share the same text.

There are still relatively few examples of supervised cross-document coreferencing in the literature. Mayfield et al. (2009) implemented an SVM classifier for pairs of entity mentions in their cross-document coreference system. Entity mention clusters were formed by the transitive closure of the positive mention pairings classified by their model.

[1] Consider a set of 2,000 mentions with an average of four mentions per cluster and where a cluster is taken to represent an entity. In this case, there are 1,999,000 unique pairwise combinations of mentions. In a random draw of two mentions, there is only a 0.0002% chance that the pair will belong to the same cluster.

The authors noted in closing that better clustering algorithms would likely improve system performance significantly.

Model features for cross-document coreference inevitably start with term vectors. Bagga and Baldwin (1998b) and Bagga and Biermann (2000) were among the earliest authors to demonstrate a generalized vector space model for cross-document coreference resolution. The vector space model—the use of term vectors and the cosine similarity between vectors to determine whether two mentions refer to the same entity—continues to be widely used. For example, Singh et al. (2011) used the vector space model for factor potentials in their graphical model for cross-document coreference.

Gooi and Allan (1998) expanded on the vector space model—most importantly for the present work—by exploring the use of agglomerative clustering using cosine similarity as the distance function. They were able to show that agglomerative clustering is superior to what they termed the incremental vector space model proposed by Bagga and Baldwin (1998b).

3 Cross-Document Entity Coreference

Our cross-document coreference system uses a multi-stage clustering algorithm, where first entities are clustered, and then groups of entities are clustered. This approach is largely language-independent and can therefore be used in many languages and also between languages. The algorithm is shown in Algorithm 3.

First, entity mentions are grouped together based on the mention text, normalized by converting to lower case and removing punctuation and spaces.

Algorithm 3 Four-Stage Coreferencing

```
1. Group mentions by normalized name
2. Resolve polysemy by supervised
   agglomerative clustering
3. Resolve synonymy by merging clusters
   produced by stage 2
4. Resolve KB links for each merged cluster
```

Second, polysemy is resolved among the mention groups using supervised agglomerative clustering. For example, four mentions initially grouped together because they share the text “National Security Council” could be resolved into two separate clusters representing two different entities, perhaps identified by the canonical names “National Security Council (Romania)” and “National Security Council (India)”.

Third, synonymy is resolved across the clusters by generating a graph where the clusters from the second round form the vertices and edges are determined by a set of heuristics. Connected vertices of the graph are then considered to comprise single entities.

Fourth and finally, entities are linked to the KB. For each entity mention, the entity linker produces a link with a link confidence, or NIL. A majority vote algorithm utilizes these links and confidences to assign the KB identifier. The production of these links is described in depth in Section 4. This stage is not necessary for classic cross-document coreference but is integral to the KBP entity linking task.

This multi-stage approach helps resolve the issues of skewed distributions typical to coreference problems. By restricting the second stage in the manner described, the number of positive and negative training examples is balanced. Over the 2009 and 2010 TAC KBP training data, an approach which compares all positive and negative training examples has only 0.07% positive examples. In our algorithm, the second stage has 66% positive examples.

Another advantage to this multi-stage approach arises from the fact that many similarity features do not work when the source documents are in different languages — at least not without the aid of machine translation or transliteration. For example, the standard term vector methods to compute document similarity fail when the terms are in different languages. By limiting the model to classifying mentions with the same text, the cross-lingual training pairs are effectively avoided.

3.1 First Stage: Group Mentions By Normalized Name

In the initial stage entity mentions are partitioned into subsets each containing mentions with identical normalized text strings. For English-language mentions the text strings are lowercased. In a cross-lingual setting, each partitioned subset contains mentions of only one language.

3.2 Second Stage: Resolve Polysemy by Supervised Agglomerative Clustering

In the second stage, mentions within subsets are clustered by a supervised agglomerative clustering algorithm with the standard pairwise model (Culotta et al., 2007). For each subset of mentions, individual mentions are initially placed into a singleton cluster. Clustering is accomplished using an average-linkage-between-group algorithm with a logistic regression classifier for the distance function. The features for the classifier are described in Section 3.2.1.

The distance between clusters M_1 and M_2 is calculated as the average—across all pairs of mentions between the two clusters—of the values of function f, where f applies the classifier to pairs of mentions.

$$d(M_1, M_2) = \frac{1}{|M_1| \cdot |M_2|} \sum_{m_1 \in M_1} \sum_{m_2 \in M_2} f(m1, m2)$$

Clustering proceeds in a greedy fashion and halts when the current largest value of d is less than a threshold parameter τ.[2]

3.2.1 Features

The logistic regression classifier is trained using 24 features. The following describes in a general manner the important feature categories. These features are used for both Chinese and English mentions.

3.2.2 Entity Type Features

Named entity recognition is applied to the document context for each mention. We test whether two mentions share the same general entity type[3], have conflicting types, or if one or both have an unknown type. Two other features return the relative percentage of entity names or types in common between the two mention documents.

3.2.3 Entity Linking Features

For each mention, the entity linker described in Section 4 provides either a proposed link to Wikipedia with an associated confidence, or it indicates NIL which means that no link was found above a certain confidence threshold. The first entity linking feature tests whether two mentions share the same link. We expand on this feature with other features that: (a) calculate whether shared links mutually exceed or do not exceed a threshold confidence parameter; and (b) calculate a joint confidence that measures the product of confidence values if two mentions share a proposed link to the same entry.

3.2.4 Term Similarity Features

We use two different term similarity features. The first uses term vectors for each document that contains a mention. The vector consists of all porter-stemmed terms in the document weighted by the standard TFIDF algorithm. The cosine similarity function is applied to pairs of these vectors. In addition, a bag-of-words vector is created for both documents and binary cosine similarity is used to compute the feature.

3.2.5 Local Context Features

One of the shortcomings of using document level term vector similarity is that entity disambiguation often requires more context-specific information. Entity references occur in many different document contexts, and a single document context typically contains many different entities. Features that operate at the document-level cannot be exclusively relied upon.

We therefore include several features in our model that operate at a sentence-level context. In particular, we isolate noun phrases that contain entity references and subject them to a number of tests. One feature examines the noun phrases in which the mentions are embedded and tests whether the phrases are equivalent. A second feature tests whether the embedding phrases disagree. For example, given two mentions with the text “Novosti”, the embedding phrases ”Vecernje Novosti” and “Moskovskiy Novosti” trigger the feature. A third feature tests whether one embedding phrase is a subset of the other.

3.3 Third Stage: Resolve Synonymy

In the third stage the clusters from the second stage are structured as a graph where each vertex represents a single cluster. Edges are created between the vertices wherever the following condition is met

[2] We chose $\tau = 0.45$ after observation.

[3] For KBP, there are three general types defined for entities: PERSON, ORGANIZATION, and GEO-POLITICAL-ENTITY.

$$\sum_{m_1 \in M_1} \sum_{m_2 \in M_2} \alpha_k I_k(m_1, m_2) > \kappa, k \in (1, 2, \ldots)$$

and where $\kappa \propto |M_1| \cdot |M_2|$. This condition is defined over a set of k indicator functions and evaluated pairwise over the graph's vertices.

3.3.1 Third Stage Features

The indicator functions utilized in the third stage are

$$I_1 = \begin{cases} 1 & \text{if } m_1 \text{ and } m_2 \text{ link to the} \\ & \text{same KB or Wikipedia entry} \\ & \text{with confidence} > \lambda.^4 \\ 0 & \text{otherwise} \end{cases}$$

$$I_2 = \begin{cases} 1 & \text{if } m_l \text{ and } m_m \text{ are embedded} \\ & \text{in a longer common phrase.} \\ 0 & \text{otherwise} \end{cases}$$

Several other functions were experimented with, but not used. These include acronyms and Dice similarity, which proved to be too imprecise. After this graph is constructed the connected vertices are merged together to form the final entity clusters.

3.4 Fourth Stage: Resolve KB Links

The fourth stage is optional for traditional cross-document coreference evaluation, but integral to the entity linking task with NIL clustering. This stage links each cluster to a KB entry. The clusters produced by stage 3 are linked to the KB according to a majority vote algorithm with random tie-breaking. Each mention in a cluster contributes a single vote. A mention votes for a NIL cluster if the mention was not previously linked to the KB by the linker. Otherwise, a mention votes for the particular KB entry assigned by the linker.[5]

4 Native Language Entity Linking

We use a language-independent entity linking approach to identify entities and associate them with a native language knowledge base (NKB) in the language of the entity's text. This approach uses different languages of Wikipedia as the NKBs since it is the largest multi-lingual resource of this type. While the approach discussed in this paper describes the Chinese system, it is trivial to extend to any language with sufficient coverage in Wikipedia.[6]

This approach is an extension of Lehmann et al. (2010), which dealt only with English. In addition to reducing language dependence, we report on several other enhancements made since 2010. Several details are omitted, including a full listing of the features used and system performance in the TAC KBP 2010 evaluation. This information can be found in the previous publication.

Our approach to linking employs three stages. First, the system generates all candidate NKB entries to which the given entity mention or query might refer. Next, it ranks the candidates and identify the most likely one, incorporating a variety of evidence. Finally, it detects if the top-ranked candidate is the correct one, or if the actual reference is unknown in the NKB and NIL should be returned.

4.1 Candidate Generation

In candidate generation, we attempt to identify every potentially correct NKB entry for the query mention string. Following are the candidate generators used to map entity strings to potential referents.

Normalized Articles and Redirects map the normalized forms of each article's name and redirect page names to the original page name. A normalized name is lowercased and stripped of whitespace and disambiguation labels. We also converted diacritics to their 7-bit representation.

Surface Text to Entity Map (STEM) associates all hyperlink anchor texts to their target pages. Popular targets are more frequently referenced, which can yield a wide variety of name variants.

Disambiguation Pages maps every disambiguation page name to the hypertext anchors on that page which are superstrings of that page name.

Search Engine Results maps queries to URLs from the Google API, which are constrained to the native Wikipedia website.[7] This approach was only

[3] We chose $\lambda = 0.75$ after observation.

[5] We tried both unweighted and weighted voting using the linker's confidence value as the weight and a default NIL weight for mentions that did not have a proposed link. We found that unweighted voting performed slightly better.

[6] 39 languages contain at least 100,000 pages as of October 2011.

[7] e.g. `http://zh.wikipedia.org`

used in web-enabled runs.

Both the disambiguation page and search engine sources use Dice similarity and acronym tests to ensure the generated candidates were sufficiently similar to the target. For Chinese entities, these are not used given the shorter character length and lack of acronyms; however these constraints do not introduce precision problems or prevent other matches from taking place in our experience.

4.2 Candidate Ranking

After generating the candidates, the system ranks them to identify the most likely sense. We first rank candidates by combining the *link combo* feature (described below) with a bonus if a high precision alias is encountered.[8] This ranking also eliminates candidates which are identified to be semantically inconsistent with the mention.

We further rank candidates using the logistic regression classifier trained to detect NIL entities, described in Section 4.3. This classifier's outcome label confidence is used to re-rank the top N candidates, which have been identified and initially ranked with the heuristic.[9] This classifier is distinct from the cross-document coreference classifier.

Five categories of feature groups are used.

Surface Features focus on the entity mention independent of context. One indicates the *link probability* based on the percent of mention string links in STEM which target the candidate sense. Other features test the similarity between the mention string and the candidate sense's name using Dice and acronym-similarity tests.

Contextual Features utilize portions of the source document outside of the entity mention. For example, they can compare the entity mention document context and the candidate backing document context.[10] While vectors of stemmed terms are often used to model context similarity, our approach is based on Milne and Witten (2008) which models context terms as Wikipedia page concepts. In this case, terms are richer in that they are disambiguated and are referenced by other disambiguated terms through Wikipedia's hyperlink graph.

Another feature indicates the *link similarity*. First, context terms are selected from low ambiguity spans of text.[11] Next, the vector of these context terms is compared to the candidate page's in-bound Wikipedia links using the Google Normalized Distance (Cilibrasi and Vitanyi, 2007).

Two other features represent contextual evidence found in the document. One represents other known name variants. The other represents entities with a known connection to this entity via a fact database such as DBpedia.[12]

In 2011, we added several extra features to better exploit local context surrounding the entity mention. In the sentence, "He moved from Missouri to Springfield, Illinois", both "Missouri" and "Illinois" were considered equally as context terms for "Springfield". However, the city "Springfield" is part of the larger span "Springfield, Illinois". The LargerEntitySpan (LES) feature detects which of the KB candidates *Springfield (Illinois)* or *Springfield (Missouri)* occur with the context "Springfield, Illinois".

The LES also works in the opposite direction. Given the entity "Georgia" in the context of "Atlanta, Georgia", the country *Georgia* is not considered to be a possible candidate. To improve the recognition of this feature we added normalization logic which utilizes a dictionary of common abbreviations. Examples of this are "Dallas, TEX", or "Washington Corp". Is this case, the normalized lookup will search for the entities "Dallas, Texas" and "Washington Corporation".

Semantic Features combine surface and contextual evidence to provide the entity types for the mention and candidate, along with their compatibility. The semantic type for the entity mention is determined using LCC's CiceroLite NER system (Lehmann et al., 2007) in English or Chinese. The candidate sense's entity type feature is set using a cascade of resources including DBpedia and LCC's WRATS ontology.[13] Using this cascaded approach, we observe 97% precision with 95% recall

[8] We selected a weight of 0.2.

[9] We use $N = 3$.

[10] The backing document for a NKB entry is its Wikipedia article.

[11] Ambiguity is measured in terms of link probability and observed frequency.

[12] `http://www.dbpedia.org`

[13] WRATS contains Wikipedia page names with one of twelve semantic types and classification confidence, with 93% accuracy.

in English. For Chinese, we use Wikipedia cross-language links to map WRATS into the NKB.[14]

Generation Features indicate the origin of the candidate sense since some candidate sources are more noisy than other. Another feature indicates the number of sources which generated the candidate.

Other Features combine features from the previous groups. For example, the *link combo* joint feature provides the weighted average between link similarity and link probability.

4.3 NIL Detection

We train a binary logistic classifier to learn the likelihood that our top-ranked candidate is not merely the best option, but the actual reference. The features are previously described in Section 4.1. For training data, we used all candidates which our system ranked to position N or higher, which also had non-NIL keys.[15] If the classifier rejects the candidate, the linked entity target is considered to be an *unknown NIL.*

4.4 Mapping from NKB to TAC KB

The English linker performs a simple mapping from English Wikipedia to KB entries. The Chinese linker links to Chinese Wikipedia. To map these links to the KB, the Wikipedia Cross-Language Links (CLLs) are used. CLLs map from a Chinese entry to the corresponding English entry, which in turn is mapped to the KB. This process is illustrated in Figure 2.

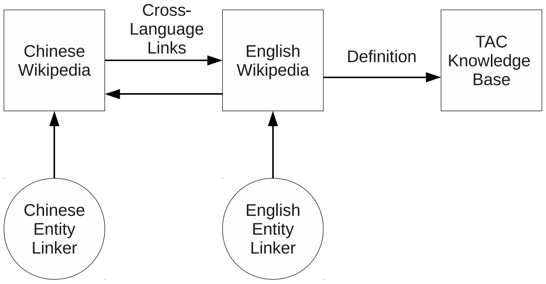

Figure 2: Mapping Entity Links to TAC KB.

In practice, there are several problems with this approach. One problem is missing, incorrect, or ambiguous CLLs. Another is with entities that exist in English Wikipedia which do not exist in the native Wikipedia. A simple machine translation approach was experimented with to improve this case, and more complex approaches, including the use of transliteration, are possible.

5 Translated Entity Linking

An alternative to native language entity linking is to translate the mention document into English and link directly to English Wikipedia. There are several advantages and disadvantages to this approach. First, an English entity linking system can be used for any language for which translation exists. Next, it avoids the issue of having to use transliteration or cross-language links to convert from the NKB. Finally, the system can successfully link to a reference which is contained in the English Wikipedia, even if it is not contained in the NKB. Conversely, if the entity is only known in the NKB, the translation approach will not produce a specific reference. In addition, this system can suffer based on the fidelity of the translation.

5.1 Implementation

Translation based entity linking was implemented using the Bing API[16] to translate both queries and their documents. We then use our English entity linking system to link these queries in the translated documents. This system was compared to the Chinese entity linking system over the 2011 KBP Chinese entity linking training data, and showed a 2% score loss, although 8.5% of the responses were different between the two systems. This suggested that a combined approach which uses both native language entity linking in addition to a translated version might produce higher performance.

5.2 Combined System

Entity linking responses from the native language system and the translated system are either KB identifiers (KBID) or NIL. Given this input, there are a variety of different algorithms which can be used to select the final response. Our experiments showed

[14] Using a semi-supervised approach, WRATS can easily be constructed for foreign languages given a subset of pages mapped from English.

[15] The value of N is determined by the ranking settings. NIL keys cannot be used, since the actual target is unknown.

[16] `http://www.microsofttranslator.com/dev/`

that the best strategy is to always choose a KBID over NIL if possible. This fits with the understanding that some entities are only in the NKB or the English KB. If the systems produce different KBIDs, the more confident KBID is returned. The confidences originate from the machine learning classifier, and because separate models are trained for each language, confidences are normalized to be the difference between the average confidence produced by the model over all of training examples.

6 Experimental Results

We evaluated our cross-document coreference system with entity linking on both the TAC 2011 KBP English and Chinese-English Cross-Lingual Entity Linking tasks. In this section, we will refer to our cross-document coreference system from Section 3 as the *clusterer*, and the entity linking system from Section 4 as the *linker*. The combination of the clusterer and linker is used to produce these results. The results were scored using the *B-Cubed+*[17] algorithm, which provides precision, recall, and F-measure metrics.

6.1 English Results

The English task required 2,250 entity mentions to be clustered together and, if applicable, linked to the TAC KB, a subset of the 2008 English Wikipedia. LCC submitted runs for three system variants, which produced results seen in Table 1. LCC1 was the default cross-document system. LCC2 sought to decrease the number of false positives by increasing the τ parameter. LCC3 had no web access and is the same as system LCC1 except for the search engine candidate generator.

In TAC, the primary system comparisons were drawn from the no web access systems, and LCC3 achieved top performance across all submissions in the task with an F-score of 84.6%. Interestingly, LCC2 did not gain any precision, as on the development set, but did suffer from a loss in recall.

We compared two approaches of coupling the clusterer with the linker. In the inductive approach, we gave the clusterer the freedom to recluster all entities, using the linker output only as features. In the deductive approach, the linker output was considered to be ground truth and only NIL mentions were clustered. Table 2 shows that the inductive system gains 0.6% F over the deductive system. In this table, "Avg" corresponds to the KBP 2010 micro-average, the metric use to score entity linking without NIL clustering. There is a 0.4% gain when using the inductive approach, meaning that by clustering the mentions, improvements were made to the linker's initial output.

Submission	**P**	**R**	**F**
LCC1	86.7	87.1	86.9
LCC2	86.7	86.2	86.4
LCC3 (no web)	84.4	84.7	84.6

Table 1: English NIL Clustering Scores.

Another experiment measured the benefit provided by the entity linking features used in stage 2 of the clusterer. That is, NIL clustering was performed without suggestions from the entity linker, although linking was still used to provide KB identifiers in stage 4. In Table 2, the comparison of *Inductive-LF* to the normal inductive approach shows that with linking features, the system gains 1.9% F, amounting to an 11% error reduction.

System	**P**	**R**	**F**	**Avg**
Inductive	84.5	84.6	84.6	86.1
Deductive	84.2	83.7	84.0	85.7
Inductive-LF	82.1	83.2	82.7	84.7

Table 2: Inductive vs. Deductive Approaches.

Enhancements to the 2010 entity system, described in Section 4, in turn improved the end-to-end NIL clustering system. Table 3 shows a micro-average increase of 2.4%, which resulted in an additional 2.6% F for NIL clustering, or 14.4% error reduction.

System	**P**	**R**	**F**	**Avg**
2010 System	81.7	82.2	82.0	83.7
2011 System	84.5	84.6	84.6	86.1

Table 3: Impact of Linker Improvements on 2011 Eval.

6.2 Cross-Lingual Results

The cross-lingual task consisted of clustering 1,481 entities in Chinese documents and 695 entities in English documents, and linking back to the KB when applicable. These entities formed 979 clusters,

[17] `http://nlp.cs.qc.cuny.edu/kbp/2011/KBP2011_TaskDefinition.pdf` modifies the B-Cubed+ scoring metric (Bagga and Baldwin, 1998a).

with only 26 of these clusters being cross-lingual. 22 of these cross-lingual clusters were connected to the KB, meaning that less than 1% required true cross-lingual cross-document coreference.

We submitted runs for three system variants which are seen in Table 4. CL-LCC1 was the default system with no web access, while CL-LCC2 did have web access. CL-LCC3 used the combined native plus translated entity linking system. Table 4 shows the results of these three systems. Our no web system, with a B-Cubed+ F of 78.8% was the top submission.

Submission	**P**	**R**	**F**
CL-LCC1 (no web)	**78.6**	79.0	78.8
CL-LCC2	80.7	81.2	80.9
CL-LCC3	78.8	81.3	80.0

Table 4: Cross-Lingual NIL Clustering Scores.

Table 5 shows the micro-average for the combined system, as well as for only the English or Chinese portions of the data set. Interestingly, the English-only portion of CL-LCC1 performed the same as the equivalent system in the English task, although the web-enabled versions scored higher here. The Chinese system was 3% lower than the English system.

Submission	**Combined**	English	Chinese
CL-LCC1 (no web)	82.4	84.6	81.30
CL-LCC2	84.3	87.34	82.92
CL-LCC3	83.9	87.48	82.17

Table 5: Cross-Lingual Micro-Average Scores.

Table 6 shows the results of the translation experiments on the evaluation set, which were lower than expected. On the development set, the combined system showed a 1.67% F improvement over the native language system. This same system lost 0.61% F on the 2011 evaluation set. It is worth noting that the native language and translated system both performed better on the evaluation set, 1.95% and 0.85% respectively.

System	**Dev Set**	**Eval Set**
Chinese System	80.9	**82.9**
Translated System	79.0	79.8
Voting System	**82.6**	82.2

Table 6: Linking on 2011 Chinese Data.

7 Conclusion

Entity linking is an important task where information mined from text can be connected to and stored with preexisting knowledge of entities in the world. TAC's Knowledge Base Population task provides an excellent benchmark against which to build such a system. In 2011, this task was expanded to include the requirement of NIL clustering along with the optional task of cross-lingual linking from Chinese to English.

We have shown how the entity linking with NIL clustering task can be cast as a cross-document coreference problem, where the clusterer and linker are mutually beneficial. Our 4-stage clustering process was used to split the problems of polysemy and synonymy and resolve them separately. We attained the best end-to-end system score across all submissions by using a linker's output inductively in a process which reclustered all entities.[18] Also, cross-document coreference clusters were improved by the use of the linker. In addition, we reported refinements to our 2010 linker, including better use of local context, which reduced system error by almost 15% in the end-to-end task.

We also showed that our approach is mostly language-independent and performs well in both English and Chinese. Our entity linker was modified to perform native language entity linking, and used to achieve top results for the cross-lingual entity linking task. In an extra experiment, we performed entity linking on translated text and combined this output in a voted system. While this showed promise on the development data set, it did not result in gains in the evaluation.

8 Acknowledgements

The authors gratefully acknowledge the support and funding of part of this work by the Air Force Research Laboratory (AFRL). Any opinions, findings, and conclusion or recommendations expressed in this material are those of the authors and do not necessarily reflect the view of the AFRL or the US government.

[18] This reclustering was disallowed in KBP 2010.

References

R. Akbani, S. Kwek, and N. Japkowicz. 2004. Applying support vector machines to imbalanced datasets. In *Machine Learning: ECML 2004*. Springer, September 20–24.

A. Bagga and B. Baldwin. 1998a. Algorithms for scoring coreference chains. In *Proceedings of the Linguistic Conference Workshop at the First International Conference on Language Resources and Evaluation*, pages 563–566.

A. Bagga and B. Baldwin. 1998b. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics*, pages 79–85.

A. Bagga and A. Biermann. 2000. A methodology for cross-document coreference. In *Proceedings of the Fifth Joint Conference on Information Sciences*, pages 207–210.

R.L. Cilibrasi and P.M.B. Vitanyi. 2007. The Google Similarity Distance. *IEEE Transactions on Knowledge and Data Engineering*, 19:3:370–383.

S. Cucerzan. 2007. Large-scale named entity disambiguation based on wikipedia data. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 708–716.

A. Culotta, M. Wick, and A. McCallum. 2007. Large-scale cross-document coreference using distributed inference and hierarchical models. In *Proceedings of the Annual Meeting forthe Association for Computational Linguistics*, pages 81–88.

C.H. Gooi and J. Allan. 1998. Cross-document coreference on a large scale corpus. In *Proceedings of Human Language Technology Conference / North American Association for Computational Linguistics Annual Meeting*, Boston, Massachusetts.

H. Ji and R. Grishman. 2011. Knowledge Base Population: Successful Approaches and Challenges. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1148–1158. Association for Computational Linguistics.

J. Lehmann, P. Aarseth, L. Nezda, Sarmad Fayyaz, Arnold Jung, Sean Monahan, and Meeta Oberoi. 2007. Language Computer Corporation's ACE 2007 System Description. In *Proceedings of 2007 Automatic Content Extraction Conference*.

J. Lehmann, S. Monahan, L. Nezda, A. Jung, and Y. Shi. 2010. LCC Approaches to Knowledge Base Population at TAC 2010. In *Proceedings of 2010 Text Analysis Conference*.

J. Mayfield, D. Alexander, B. Dorr, J. Eisner, T. Elsayed, T. Finin, C. Fink, M. Freedman, N. Garera, P. McNamee, S. Mohammad, D. Oard, C. Piatko, A. Sayeed, Z. Syed, and R. Weischedel. 2009. Cross-document coreference resolution: A key technology for learning by reading and learning to read. In *AAAI 2009 Spring Symposium on Learning by Reading and Learning to Read*.

D. Milne and I.H. Witten. 2008. Learning to link with Wikipedia. In *ACM Conference on Information and Knowledge Management (CIKM'2008)*.

S. Singh, A. Subramanya, F. Pereira, and A. McCallum. 2011. Large-scale cross-document coreference using distributed inference and hierarchical models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 793–803, Portland, Oregon, June 19–24. Association for Computational Linguistics.