Lorify: A Knowledge Base from Scratch

Sean Monahan, Dean Carpenter
Language Computer Corporation
smonahan@languagecomputer.com

Abstract

In this paper we discuss our approach to the task of Cold-Start Knowledge Base Population and the challenges associated with it. We describe our knowledge base system Lorify and each of the components necessary to populate it from unstructured text. The pivotal component for building a large-scale knowledge base is scalable cross-document coreference. We address this with a novel clustering algorithm based on Markov-Chain Monte-Carlo, and show that it is capable of scaling to much larger sets of entities than typical algorithms. Finally, we detail the performance of this system on the TAC KBP 2012 evaluation.

1 Introduction

A Knowledge Base (KB) is a repository of knowledge that is suitable for both human and machine-readability. Knowledge Base Population (KBP) is the task of incorporating the information in a corpus of unstructured text into a structured representation. The additional task of Cold-Start KBP assumes that a majority of the information in the corpus will not supplement existing KB entries, but rather construct new ones.

The primary unit of a KB is an *entry*. Each entry contains all of the information known about a single item of interest, e.g. an entity or event. The archetype for KBs is the community-edited Wikipedia, where each entry corresponds with a unique URL. Each entry in a typical KB contains information about the item, which can broadly be split into several categories, *Facts*, *Mentions*, and *Summary*, each of which is closely aligned with a goal of the Textual Analysis Conference (TAC). *Facts* are structured information about the entry (e.g. in Wikipedia the infobox, categories, and lists; in TAC KBP slots). *Mentions* are occurrences of the entry in unstructured text (e.g. in Wikipedia the citations; in KBP entity links). Given a large corpus, there are potentially hundreds, thousands, or more mentions of a single entity.[1] The *Summary* is a free text portion of the entry which provides human-readable information about it (e.g. in Wikipedia the text of the page; in TAC the summarization task). Additionally, this information can come from multiple languages. In Wikipedia, the same concept has a different webpage for each language, however in a global KB, each of these pages are considered to be part of the same entry (e.g. in Wikipedia Cross-Language Links; in KBP Cross-Lingual Entity Linking).

Having a KB is important for a variety of reasons. While the utility of Wikipedia needs no justification, its coverage is limited to concepts of global importance. The ability to create a KB for a series of novels or for the website of a small town would greatly enhance the human reader's ability to capture large amounts of important knowledge about a subject in a significantly shorter span of time than reading all of the source material. A KB is similarly useful from a computational perspective, as systems such as those for question/answering can utilize the facts to answer questions, as part of the push towards Open Data (Chiarcos et al., 2012).

In this paper we examine the variety of challenges necessary to create a KB from scratch, and describe

[1] For example, the State of Maryland is mentioned over $10,000$ times in English Wikipedia.

our system to create LCC's **Lorify** KB. Additionally, we focus on the problem of grouping all of the mentions of an entity together over a large corpus, which we resolve using a Markov Chain Monte-Carlo (MCMC) approach to cross-document entity coreference. Finally, we present our results for the Cold-Start KBP task, as well for the entity linking and NIL clustering component.

2 Related Work

The premier Knowledge Base amongst experts and casual users alike continues to be Wikipedia. It is the largest and best repository of knowledge available, and it was created entirely by a large community of editors and readers. One disadvantage is that for new information to be added, the information must be learned by an editor, determined to be notable[2], and written into the appropriate page. Another is that this information is in the form of unstructured text, not suitable for machine-readibility.

The goal of the associated DBpedia (Bizer et al., 2009) project is to provide a structured data representation of Wikipedia, and provide access in a machine-readable format. Yago2 (Hoffart et al., 2011) adds another dimension by extending the knowledge with temporal and spatial qualities. Both of these resources, however, are still limited by the knowledge that has been added by editors.

Among projects which seek to learn information directly from unstructured text is CMU's NELL: Never-Ending Language Learning (Carlson et al., 2010). The project seeks to "Read the Web" to learn new categories and relations to associate entity mentions with those categories, as well as improving its own abilities over time. However, it does not consider the problem of linking different mentions of the same entity with a KB entry.

In contrast to NELL, one of the strengths of our system is that it can determine when two mentions refer to the same entity. The entity can either be an existing KB entry (entity linking) or a new KB entry (NIL clustering). Some of the earliest work on entity linking was done by Cucerzan (2007), and entity linking by itself was a task at TAC KBP (2009,2010). An overview of the state-of-the-art entity linking approaches has recently been summarized by Ji and Grishman (2011).

Closely related to NIL clustering is the area of cross-document coreference, which determines when two entity mentions in different documents refer to the same entity. The earliest attempts at solving this problem use the vector-space model for calculating similarity (Bagga and Baldwin, 1998), and more recently have progressed into classification models such as Mayfield et al. (2009). Gooi and Allan (1998) expanded the vector-space model by exploring the use of agglomerative clustering. Singh et al. (2011) used the vector space model for factor potentials in their graphical model for cross-document coreference.

The Entity Linking with NIL clustering task at KBP (2011,2012) combined both the state-of-the-art entity linking and cross-document coreference approaches. Monahan et al. (2011) showed that these two tasks are interrelated; that cross-document coreference can improve entity linking, and vice versa.

Once the entity mentions have been associated with the appropriate KB entries, a key task is extracting the facts about those entity mentions that should be inserted into those entries. Fact extraction from unstructured text was the focus of the KBP slot filling task (2009-2012), along with the subtasks of surprise slot filling (2010) and temporal slot filling (2011). A good overview of the techniques used for the slot filling problem can be found in Ji and Grishman (2011).

3 A Unified Framework for Knowledge Base Creation

In this section we describe the pipeline for populating Lorify from scratch, and how this is used to accomplish the Cold-Start KBP task. Given a corpus of documents from which to construct a KB, we break down the task into the following components: document zoning, entity extraction, coreference, fact extraction, entity linking, cross-document coreference, and information fusion, which are detailed in the following sections and illustrated in Figure 1. The first column illustrates the unstructured data which comes into the system, and the KB which is produced from it. The second column indicates the components which are typically consid-

[2] http://en.wikipedia.org/wiki/Wikipedia:Notability

ered part of the slot filling task, and the third indicates the components of entity linking with NIL clustering. The backwards arrows from the entity clustering component illustrate how this component can feed information back into the previous steps.

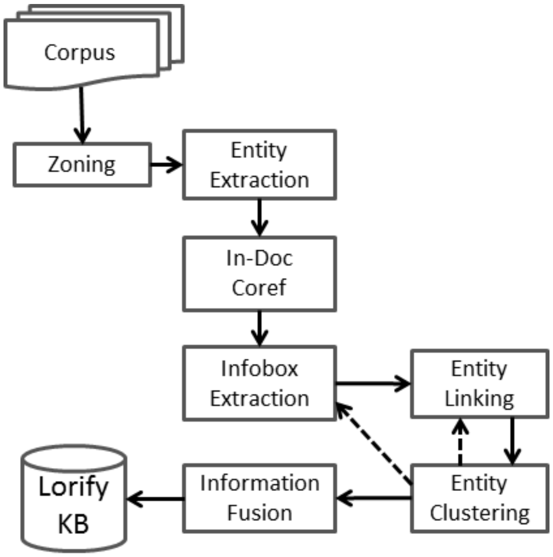

Figure 1: Lorify Pipeline

3.1 Document Zoning

Given a corpus of documents from which to construct a KB, the first step is to partition the input data into textual zones and non-textual zones, so that the subsequent steps only operate over well-formed natural language text. On a standard news article (e.g. CNN), there exists a header, footer, and potentially multiple sidebars, in addition to tables, figures, and captions, none of which are part of the text of the article. In the KBP Cold-Start data, for example, many of the documents contain copyright notices.

This step is not absolutely necessary; a system could cluster the footers mentioning University of Pennsylvania, or the input could consist of only the text portions of the documents (e.g. ACE sgm files). However, we consider this to be an important preprocessing step for this task. Our system for zoning utilizes the densiometric zoning approach of Kohlschtter and Nejdl (2008). The system computes the zone density, and uses a heuristic to select the appropriate zones which contain unstructured natural language text.

3.2 Entity Mention Extraction

The primary unit of the Cold-Start KB is an entity. Each entity is associated with multiple mentions, and named entity recognition (NER) is used to extract the mentions of those entities from text. In the entity linking with NIL clustering task, the text of the entities are provided, and mention extraction is solved simply by finding that text in the document. For Cold-Start KB creation, the entities must be found with knowledge of the text. The type of entities in the TAC KB are person, organization, and geo-political entity, but other entity types such as *temporal* must also be extracted in order to support fact extraction. We use LCC's CiceroLite NER system (Lehmann et al., 2007) for named entity extraction, which is a statistical algorithm based on a maximum entropy classifier.

3.3 Within Document Coreference

Within a document in the corpus, the same entity may be referred to in multiple ways, the most common of which are pronominal and nominal coreference (e.g. he/her, the man), and each of the references must be resolved. The importance of this is most apparent in fact extraction, with sentences like "He married her in 2005". We use LCC's coreference system which focuses on reliably extracting the name to name coreference using a heuristic algorithm, and extracting pronominal coreference using a statistical algorithm based on (Hobbs, 1976).

3.4 Fact Extraction

Once the mentions of the entity have been detected, the facts about those entities can be extracted. For clarity, we speak of *relationships* when referring to facts where the target is itself in the KB (e.g. spouse), and *attributes* when the target is not in the KB (e.g. number of employees). Our system extracts 72 types of facts (of which the required KBP slots are a subset), using the techniques described in (Lehmann et al., 2010). These extractors were tuned to have a very high precision. In addition, our system also extracts "generic" facts, associated with a semantic relation parser. If the entity is the subject of a predicate, a generic fact is created between that entity and the object of the relation, with the type being the predicate. For example, (Jim, went to, Dallas). Generic facts are used by cross-document coreference, question/answering, summarization, and a variety of other NLP tasks.

3.5 Entity Linking

Once the information about the entity has been extracted, it can be linked to a KB (in TAC this is Wikipedia), or determined to be NIL (meaning it has no corresponding Wikipedia page). For Cold-Start, this step is optional, because many of the entities, especially the people, are not associated with Wikipedia. We use the linking system described in (Lehmann et al., 2010) and (Monahan et al., 2011), which is based on a variety of surface features to find candidate Wikipedia pages, semantic features to determine the most likely candidate, and a machine learning classifier to determine if the result should be NIL.

One variant to this model would be to link directly to the Cold-Start KB. For example, proceeding document by document, when an entity mention is encountered, it is either linked to an existing KB entry, or is put into a newly created KB entry. However, when actually comparing an entity mention to a KB entry consisting of only one other entity mention, the comparison is fundamentally the same as that of comparing two NIL clusters, since both have only a small amount of information about that entity. When linking to an existing KB like Wikipedia, where entities are often referenced many thousands of times, the comparison is significantly different. For this reason, we consider the two paradigms of clustering and linking to small KB entries to be fundamentally equivalent.

3.6 Cross-Document Coreference

Once all of the information from within the document has been extracted, the next step is to cluster the entity mentions across the documents in order to consolidate them into a single KB entry. In this and future sections, we define a cluster to be a group of entity mentions. Once the clustering algorithm is completed, each cluster becomes associated with a single entity and KB entry.

The cross-document coreference component (or NIL clustering if entity linking is enabled) must solve the two primary problems detailed in (Monahan et al., 2011). The first problem is synonymy, or determining when two entity mentions could refer to the same entity. Naive models which assume any two mentions could do so are computationally infeasible, requiring n^2 comparisons. A proposal model which restricts the space of synonyms allows the algorithm to procede much faster, but the performance is limited by the recall of the proposals. The second problem is that of polysemy, determining when two entity mentions which appear to be the same (e.g. the same name), are actually different. When combined with the proposal model, this task becomes that of determining when two entity mentions which are proposed to be the same belong to the same cluster. Further details of this algorithm are provided in Section 4.

3.7 Feedback Loop

Once the clusters have been created, it possible to use information from one entity mention in a cluster and apply it to other mentions. In (Monahan et al., 2011), it was shown that on the 2011 KBP task the performance of entity linking can be improved by clustering, and that entity linking features are also useful for clustering. There are several other potential ways feedback could be used. The first is correcting mistakes in the named entity recognition system. If a span of text is “Newton”, NER may determine it is a person using the original document, but by examining the context across multiple documents, if that entity belongs in the cluster with a location, the entity type can be corrected in the original document. If the facts extracted with that entity were associated with a person, they can be removed and new facts extracted that are associated with a location.

3.8 Information Fusion

Given all of the facts associated with the mentions in a cluster, it is possible that some of the facts are contradictory or duplicates. It remains to select the appropriate values to display in the infobox. Given a fact and a list of potential values, these values must be normalized. For example, “born on Jan 1, 1975”, and “born in 1975” are logically consistent, as are “occupation:attorney” and “occupation:lawyer". For TAC KBP, the most specific value is preferred. For single valued slots, we must solve the additional task of selecting the best from a list of values. The primary features for this task are usually the frequency of each fact, along with the confidence scores associated with the fact extraction and clustering. For our system, see details in (Lehmann et al., 2010).

4 Scalable Cross-Document Coreference

In this section we discuss a clustering method which seeks to greatly reduce the time spent performing cross-document coreference. Of all the tasks described above, most of them only apply to a single document, which scales linearly with the number of documents. Fact merging could potentially be expensive, but the number of values for a specific type of fact is orders of magnitude lower than the number of mentions for that entity. A pairwise cross-document coreference step scales quadratically with the number of mentions, and more complex clustering models require exponential (e.g. Bell's number) of comparisons.

MCMC, and in particular Metropolis-Hastings (MH) is a statistical technique used for estimating complex probability distributions where direct calculation is infeasible. This technique was first used for cross-document coreference by Singh (2011). In this section we describe enhancements to this method in both the proposal and similarity models, as well as the creation of a singleton step and a basic temperature control model.

4.1 Overview

The goal of this algorithm is to take a set of entity mentions and cluster them such that each cluster refers to the same entity. The model first breaks the problem down by only allowing the comparison of two mentions if they meet some proposal. If a mention m_1 shares a proposal with another mention m_2, then m_1 could be moved to share a cluster with m_2. This movement is controlled by the similarity model, which determines how likely these two mentions are to be the same entity. Each movement is a Markov step, meaning the most statistically likely choice is not always chosen. Finally, the system has a temperature which makes the more likely statistical choice occur more often as time progresses. When the algorithm completes, each cluster is uploaded to the Lorify KB as a single entry.

4.2 Statistical Algorithm

The algorithm for MCMC clustering is shown in Algorithm 1. The algorithm initializes each entity mention to be in its own cluster, and then conducts numerous rounds of moving mentions between clusters. In our experiments each round consisted of 10,000 iterations. At each iteration of the algorithm, we perform either a movement step or a singleton step. In the movement step (as shown in Algorithm 2) an entity mention is proposed to move from cluster x to cluster y. In the singleton step (as shown in Algorithm 3) an entity mention is proposed to move from cluster x to a new cluster containing only itself.

Algorithm 1 MCMC clustering algorithm

```
0. Assign mention to default cluster
while $temperature \geq 0$ do
    for N iterations do
        1. Run movement step
        2. Run singleton step
    end for
    3. lower temperature
end while
```

Algorithm 2 Movement step.

```
1. Select arbitrary proposal $p$
2. Select two mentions with proposal $p$
   $m_i \in c_x$ and $m_j \in c_y$ s.t. $c_x \neq c_y$
3. Compute $\psi_a(x) = sim(m_i, c_x - m_i)$
4. Compute $\psi_a(y) = sim(m_i, c_y)$
5. Move $m_i$ to $c_y$ with probability
   $min(1, e^{(\psi_a(y)-\psi_a(x))\frac{1}{\tau}})$
```

Algorithm 3 Singleton step.

```
1. Select arbitrary proposal
2. Select mention $m_i \in c_x$ with this proposal
3. Compute $\psi_a(x) = sim(m_i, c_x - m_i)$
4. Move $m_i$ to $c_y$ with probability
   $min(1, e^{(\psi_a(x)-bias)\frac{1}{\tau}})$
```

The rationale behind the singleton step is that the move step can only decrease the number of clusters, because the empty clusters have no proposals to retrieve them. The singleton step is equivalent to having a global ”empty” cluster, which contains an empty proposal which matches all mentions. In this case, the bias of the similarity classifier is used to determine if the mention should remain in the cluster. The probability function was chosen so that if the entity mention is determined to be better in the new cluster, it is always moved, and if it is better in the current cluster, it is moved with some small probability. For the singleton proposal, the bias is set to a small value such as 0.2 which is determined experimentally.

4.3 Proposal Model

To break down the complex clustering problem into more manageable increments, we utilize a proposal model. This is based on the intuition that two mentions with different names are very unlikely to refer to the same entity. Whatever exceptions exist to this rule (e.g. aliases, names in multiple languages, screennames), can be encoded in the proposal model. Therefore, in order for the algorithm to move a mention into a cluster, it must share a proposal with one of the members of that cluster. Note that entity mentions without proposals in common may end in the same cluster through a chain of proposals. Singh utilized a proposal model which required the entity mention text to not have a “large string edit distance”. Our proposal model used the following features:

Morphological Fingerprint: For the entity name, the orthographic case is normalized, the individual words are sorted alphabetically, and punctuation is removed.

Entity URI: If the linking system is used, this proposal selects two entities which were linked to the same KB entry. The linking system is capable of linking aliases and different language variants.

Entity Alias: An entity with an alias found in the corpus (e.g. per:alternate_names) will have proposals for each of the different names. This allows the entity to serve as a “bridge” between two different clusters, e.g. Muhammad Ali to Cassius Clay.

4.4 Similarity Model

The similarity model is used to compare pairs of mentions that the proposal model identifies as potentially being coreferential. The features for such models typically fall into the categories of document level features (e.g. Bag of Words), context features (e.g. words in the same noun phrase), and fact features (e.g. a spousal relation). Here, the similarity model is the same as the second stage of the NIL Clustering step in (Monahan et al., 2011), using a logistic regression classifier trained on the TAC KBP entity linking and NIL clustering data (2009-2011).

4.5 Temperature Control

Given the probabilistic nature of the Markov clustering step in the Metropolis-Hastings algorithm, at any given point, a mention can be moved to a cluster that is not the highest probability. When the algorithm completes, the goal is for each mention to be in the most coherent cluster. This is accomplished through the use of a temperature. Initially, the temperature is high, and the mention is likely to jump between clusters. As the temperature decreases, the mention is likely to settle in the correct cluster, in the manner of simulated annealing.

The temperature drops over time in the following way. At time t_0, the temperature is τ_0, which is the initial temperature. The system is given a total time T over which to operate. The temperature is dropped to 0 over this time span, and clustering continues for a short time with the temperature at 0. After each iteration of clustering steps, the temperature is dropped such that the temperature at time t_i is $\tau_i = \frac{T-t_i}{T}\tau_0$. The experiments were run with $T = 5$ minutes for the entity linking data sets and $T = 12$ hours for the Cold-Start data set. The temperature was initially set to $\tau_0 = 0.25$. With these settings the system proved capable of clustering a data set with $200,000$ entity mentions. Future work will study how to automatically determine the appropriate time T, and initial temperature τ_0 for a given corpus.

5 Results

5.1 Cold-Start Knowledge Base Population

In this section we present our Cold-Start KBP submissions. None of the submissions used the web, and none used external entity resources for slot filling. There were four total submissions, varying whether or not the system used entity linking, and whether or not the system used document zoning.

Due to the size of the input data, all of the systems used the MCMC clustering algorithm. The F-measure, precision, and recall for each run are shown in Table 1. These are the scores for Combined LDC queries and derived queries at hop level 0. Also shown is the top performing system in the evaluation (McNamee et al., 2012), which utilized contextual aware entity linking with several different relation extraction engines.

System	F1	P	R	Linking	Zoning
top	49.7	48.0	51.5	yes	no
lcc2012-1	14.4	62.7	8.2	no	yes
lcc2012-2	16.5	66.4	9.4	yes	yes
lcc2012-3	17.6	62.0	10.3	no	no
lcc2012-4	18.0	67.7	10.4	yes	no

Table 1: Cold-Start Knowledge Base Population results.

These results are indicative of the system's focus on high-precision fact extraction. Additionally, the results show that entity linking is an important feature for this system, and results in a 4-5 point improvement over the equivalent system. Finally, the results of the zoning feature are somewhat inconsistent, between experiments 1 and 3 the zoning increases precision while hurting recall, but between experiments 2 and 4 the zoning hurts both precision and recall.

5.2 Entity Linking with NIL Clustering

In this section we present results for the entity linking with NIL clustering task in three languages. This task is a pivotal component for solving the overall Cold-Start KBP task, testing both the entity linking and entity clustering components. For several of our submissions we used the MCMC clustering system described in section 4, and compare this to an agglomerative clustering approach. Additionally we present results for Chinese and Spanish.

5.2.1 English Entity Linking with NIL Clustering

For English Entity Linking with NIL Clustering, we submitted 4 runs, with and without utilizing the web feature for entity linking, and using agglomerative or MCMC clustering. Runs 1 and 2 used the web and Runs 2 and 4 used MCMC clustering. Table 2 shows the B^3 F1 and accuracy score for each run, along with the competition high and median. The top performing system of Cucerzan (2012) utilized a clustering technique which merged all entries that matched a set of heuristics similar to those utilized here.

The MCMC clustering achieves nearly the same score as the agglomerative clustering, which is a solid result, given that it is a probabilistic algorithm designed to run on large datasets. Unlike in the previous three years, the web feature actually hurts the overall performance, both accuracy and F-measure.

System	B^3 F1	**Accuracy**	Web	Cluster
top	73.0	76.6	No	Merge
lcc2012-3	68.9	75.7	No	Agglom
lcc2012-4	68.5	73.1	No	MCMC
lcc2012-1	68.0	74.7	Yes	Agglom
lcc2012-2	67.7	72.3	Yes	MCMC
median	53.6	60.1	No	-

Table 2: English Entity Linking with NIL Clustering.

These scores represent a significant decrease in scores from the 2011 system, where LCC scored 84.6%. Table 3 shows statistics illustrating the difference between 2011 and 2012 data. The third column shows the total number of clusters (either linked or NIL) in the data. The fourth column shows the number of unique names in the queries, which was much lower in 2012, meaning that the ambiguity was much higher. The last column shows the number of clusters per name, which more than doubled in 2012. For comparison, a completely unambiguous name has a score of 1.0, so by this metric, the ambiguity of names in 2012 is significantly higher than 2011.

Year	#Mentions	#Clusters	#Names	C/N
2011	2,250	1,514	1325	1.14
2012	2,226	1,941	808	2.40

Table 3: 2011 vs. 2012 Data Ambiguity for mentions, clusters, and names.

Table 4 shows scores for two experiments which further indicate these points. The no clustering experiment did not cluster any of the NILs, putting each NIL into its own cluster. This has the same accuracy as the LCC submission, but a 2.5 point gain in F-measure. Because of the significantly higher ambiguity in 2012, the clustering algorithm was trained on the lower ambiguity data from previous years, and hurts the performance. To correct this issue, we experimented with changing the bias in the acceptance probability in the MCMC algorithm, from 0.2 to 0.8. This resulted in a higher F-measure than the baseline of no clustering, with the accuracy not changing. In 2011, we reported that clustering could be used to improve entity linking, but in 2012 the accuracy stayed the same throughout.

System	F	Accuracy	Clustering
lcc2012-3	68.9	75.7	Agglom
No clustering	71.4	75.7	None
High Bias	71.8	75.7	MCMC

Table 4: Post-Evaluation Experiments

Finally, when the scores further separate the newswire from web content, the LCC score of 64.6% on web data was the competition high. This value, almost 10 points lower than the total high, indicate that the web data was signficantly more difficult to cluster and link. More data is needed to draw conclusions from this result.

5.2.2 Chinese Entity Linking with NIL Clustering

Our Chinese system had 4 submissions, with the scores shown in Table 5. Each of the systems utilized the native language linking component to Chinese Wikipedia that was described in (Monahan et al., 2011), and 3 of the submissions combined this result with translation. The web feature provides an insignificant gain over the equivalent experiment without it. The native language with translation provided a 23.4 point gain over the native language only approach. Finally, the MCMC system performed 1.8 points lower the agglomerative cluster system. However, the similarity model trained for Chinese used significantly less data than the English model (9,906 to 2,998 examples), and none of the other parameters of the model were adjusted. Also shown is the top performing system (Fahrni et al., 2012), which utilized a Markov Logic Network for joint entity disambiguation, NIL detection, and NIL clustering.

System	B^3 F1	Acc	Web	transl	cluster
top	74.0	-	No	-	MLN
lcc2012-1	66.8	80.2	Yes	Yes	Agglom
lcc2012-2	66.7	80.2	No	Yes	Agglom
lcc2012-4	65.1	80.3	No	Yes	MCMC
lcc2012-3	43.3	60.8	No	No	Agglom

Table 5: Chinese Entity Linking with NIL Clustering.

5.2.3 Spanish Entity Linking with NIL Clustering

Our Spanish system had 4 submissions, with scores shown in Table 6. all of which utilized native language linking combined with translation. The system used for Spanish linking and clustering was identical to the one reported for English and Chinese last year. Our no-web submission lcc2012-3 received a 64.1 F-measure, which proved to be the competition high for this task. The agglomerative clustering again performed 1-2 points higher than the MCMC. The Spanish model was trained on the 1,449 examples provided by KBP, as compared to the 9,906 examples available for English.

System	B^3 F1	Web	cluster
top	**64.1**		
lcc2012-1	64.3	Yes	Agglom
lcc2012-3	**64.1**	No	Agglom
lcc2012-4	62.9	No	MCMC
lcc2012-2	62.1	Yes	MCMC

Table 6: Spanish Entity Linking with NIL Clustering.

6 Conclusion

In this paper we presented a system for creating our Lorify knowledge base from scratch and described each of the necessary components. We reported our results using this system for the 2012 TAC KBP Cold-Start task. Within this task, we focused mostly on the cross-document coreference component, which was also evaluated using the entity linking with NIL clustering task. We showed that our scalable MCMC algorithm performed roughly on par with an agglomerative clustering system over small data sets. Additionally we showed it was capable of clustering large data sets which agglomerative clustering could not. The results of this algorithm are presented for English, Chinese, and Spanish.

References

A. Bagga and B. Baldwin. 1998. 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.

C. Bizer, Jens Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. 2009. Dbpedia a crystallization point for the web of data. pages 154,165.

A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E.R. Hruschka Jr., and T.M. Mitchell. 2010. Toward an architecture for never-ending language learning. In *In Proceedings of the Conference on Artificial Intelligence (AAAI)*.

C. Chiarcos, J. McCrae, P. Cimiano, and C. Fellbaum, 2012. *Towards open data for linguistics: Lexical Linked Data.*

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.

S. Cucerzan. 2012. MSR System for Entity Linking at TAC 2012. In *TAC (Text Analysis Conference) 2012 Workshop*.

A. Fahrni, T. Gckel, and M. Strube. 2012. HITS' Monolingual and Cross-lingual Entity Linking System at TAC 2012: A Joint Approach. In *TAC (Text Analysis Conference) 2012 Workshop*.

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.

J. Hobbs. 1976. Pronoun resolution. In *Research Report 76-1 Department of Computer Sciences, City College, City University of New York.*

J. Hoffart, F. M. Suchanek, K. Berberich, E. Lewis Kelham, G. de Melo, and G. Weikum. 2011. Yago2: Exploring and querying world knowledge in time, space, context, and many languages. In *Demo Paper in the proceedings of the 20th International World Wide Web Conference (WWW 2011) Hyderabad, India*.

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.

C. Kohlschtter and W. Nejdl. 2008. A densitometric approach to web page segmentation. In *CIKM 2008: Proceedings of the 17th ACM Conference on Information and Knowledge Management, Napa Valley, California, USA*.

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*.

P. McNamee, V. Stoyanov, J. Mayfield, T. Finin, T. Oates, T. Xu, D. W. Oard, and D. Lawrie. 2012. HLTCOE Participation at TAC 2012: Entity Linking and Cold Start Knowledge Base Construction. In *TAC (Text Analysis Conference) 2009 Workshop*.

S. Monahan, J. Lehmann, T. Nyberg, J. Plymale, and A. Jung. 2011. Cross-Lingual Cross-Document Coreference with Entity Linking. In *Proceedings of 2011 Text Analysis Conference*.

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.