ualberta at TAC-KBP 2012: English and Cross-Lingual Entity Linking

Zhaochen Guo, Ying Xu, Filipe Mesquita,
Denilson Barbosa, Grzegorz Kondrak
University of Alberta
{zhaochen,yx2,mesquita,denilson,kondrak}@ualberta.ca

Abstract

On one hand, the proliferation of the Web has generated massive information in an unorganized way and is still growing in an accelerating pace. On the other hand, structured and queryable knowledge bases are very difficult to construct and update. Automatic knowledge base construction techniques are greatly needed to convert the rich Web information into useful knowledge bases. Besides information extraction, ambiguities about entities and facts also need to be resolved. Entity Linking, which links an extracted named entity to an entity in a knowledge base, is to solve this ambiguity before populating knowledge. In this paper, we describe ualberta's system for the 2012 TAC-KBP English and Cross-Lingual Entity Linking (EL) task, and report the result on the evaluation datasets.

1 Introduction

On one hand, the proliferation of the Web has generated massive information in an unorganized way and is still growing in an accelerating pace. On the other hand, structured and queryable knowledge bases are very difficult to construct and update. Automatic knowledge base construction techniques that automatically extracts information from the Web and populates facts into the knowledge base, are greatly needed to bring the knowledge from the rich Web to useful knowledge bases. Knowledge base construction requires not only information extraction, but also disambiguation of the facts (entity, and relationship). One crucial task is to correctly link extracted entity mentions to their true entities in the knowledge base. For instance, before populating the triple fact about *Michael Jordan* <*Michael Jordan, ownerOf, Charlotte Bobcats*> into the KB, we first need to correctly identify the true entity in the KB that the mention *Michael Jordan* refers to. Entity Linking aims to determine the KB entity for a given mention in a document, either returning the true KB entity or NIL indicating the true entity is not in the KB.

The Entity Linking shared task in TAC-KBP track at TAC 2012 is proposed to promote the research on Entity Linking. The task provides a reference KB constructed from a Wikipedia dump, and a task file containing 2229 queries. Each query is provided with a query id, the name of the query, the document containing the query, and also the position of the query in the document (start offset and end offset). Users can use the document as well as the reference KB in their approach for the entity linking task. The output is the entity each query refers to. It could be an entity in the reference KB, or NIL if the referred entity is not in the KB. For queries linked to NIL, the EL system is required to cluster together these queries referring to the same non-KB entities with cluster ID NILxxxx, in which each cluster only contains queries referring to the same entity.

Our entity linking system consists of four main components: *candidate selection*, *entity disambiguation*, *NIL clustering*, and *Chinese to English translation*. The candidate selection, according to our finding, plays an important role in the EL system. Effective candidate selection can find most true entities in the candidate set and still keep the candidate size small. We explored several meth-

ods to increase the recall of candidate selection from both the document side and the knowledge base side. For the entity disambiguation, we employed a collective approach that links all mentions in the document collectively aiming to find an optimal assignment for the mentions. We implemented the Cucerzan (2007)'s category-based approach, as well as a hyperlink-based approach, and compared the performance of these two approaches. We then employed a graph-based hierarchical clustering approach for the NIL clustering. The clustering approach utilizes the results of the entity linking and the graph of entities constructed from the document and the reference KB. Our Chinese to English translation component translates and transliterates Chinese queries and other named entities in the document to English and then uses the English EL approach to solve the cross-lingual EL task.

This report is organized as follows. Section 2 introduces our entity linking framework. Section 3, 4, 5, and 6 give detailed description of the components in our EL system respectively. The results are shown in section 7.

2 Entity Linking Framework

According to the task definition from the TAC-KBP, there is only one mention per document in each query. However, our approach performs the EL in a collective way which needs to exploit the linking results of other mentions in the same document, thus a NER is performed to extract named entities from the document prior to the candidate selection. We employed the Stanford CRF-NER [1] to extract mentions. For queries that are not in the extracted mention list, we explicitly create a mention and add it to the mention list.

3 Candidate Selection

After identifying the mentions in a document, the candidate selection component selects a set of candidates for each mention. Due to the name variations and spelling issues, names may be represented in several different ways which may result in mismatching of mentions and their true entities. To solve this problem and increase the recall, we expand the query in two aspects.

One is the knowledge base from which aliases of entities can be collected. The approach in Cucerzan (2007) is used for the alias collection. According to Wikipedia, aliases are mainly from four sources: entity title, redirect page, disambiguation page, and anchor text in Wikipedia pages. We also keep the frequency of an alias referring to an entity, and the source of the alias: name, redirect, disambiguation, and link. In the evaluation section, we show how the alias source affects the recall.

In the training dataset, we found that a few true candidates are missed because the abbreviation of the entity is not in the dictionary. To solve this problem, we explicitly build a set of abbreviations for the entities, and add the set into the dictionary. For any entity in the dictionary having at least two words with capital letters, we concatenate all capital letter together to form its abbreviation. This can increase the recall. However, it also brings in too many noisy candidates and greatly increases the candidate size. For example, very few PERSON entities have abbreviation as their alias, e.g. *MJ* for *Michael Jordan*, and *Michael Jackson*. Our results show that the abbreviation expansion increases the running time of the EL as well as decreases the accuracy. Thus we discard this expansion in our system.

The other way to increase the recall is from the document side. From the training dataset we find that many queries are also mentioned by its full name in the same document. For example, a person is introduced with its full name at the beginning and then referred to with only its first name or last name later. Also an abbreviation and its full name may coexist in the same document. Identifying the full name of a mention will greatly increase the chance to find the true entity in the knowledge base. For instance, *ABC* could refer to various entities, but we can directly tell its true entity when given its full name *All Basotho Convention*. For this task, we use co-reference resolution tools (GATE's OrthMatcher [2]) for the query expansion and improve the recall of candidate selection.

Given the expanded query set and alias collection, we use a fuzzy query over the alias collection in-

[1] `http://nlp.stanford.edu/software/CRF-NER.shtml`

[2] `http://gate.ac.uk/`

stead of the direct search over a lookup tables to handle the case of typos or spelling variations. For the implementation, Apache Lucene [3] is used to build an inverted index for the alias collection and search over the index using Dice Coefficient. Then the top-K (e.g. 5) results are used to obtain the candidates.

In general, we found that the candidate selection is actually playing a more important role in the entity linking process. A good candidate selection approach can greatly increase the recall and reduce the size of candidate set so as to further improve the accuracy and efficiency of the entity disambiguation.

4 Entity Disambiguation

Our entity disambiguation component employs a collective approach that aims to find an assignment for all mentions in a document which can maximize an objective function. We exploit not only the local features such as bag-of-words, but also the global feature such as the coherence of entities. The objective function is defined as follows.

$$E^* = \arg\max \sum_{i=1}^{N} (\phi(m_i, e_i) + \sum_{e_j \in E} \psi(e_i, e_j))$$

in which E is an assignment for mentions in the document, m_i is the given query mention, and e_i is a potential candidate. $\phi(m_i, e_i)$ measures the local similarity between m_i and e_i, and $\sum_{e_j \in E} \psi(e_i, e_j)$ measures the coherence between candidate e_i and the rest of entities e_j in the assignment E.

We employ the approach proposed in Cucerzan (2007) to compute the local similarity $\phi(m_i, e_i)$. For entity e_i, a context vector is built from the named entities in its Wikipedia page. For mention m_i, its context vector consists of named entities extracted using NER tools and named entities matching against the entity dictionary (includes entity name and their redirect name). Note that all mentions in the same document have the same context vector.

The global coherence of E is measured by the relatedness between entities. In our system, we implemented two relatedness measures, a category-based measure (Cucerzan, 2007), and a hyperlink-based measure (Milne and Witten, 2008). Details are given below.

[3] `http://lucene.apache.org/`

4.1 Category-based Relatedness

Category in the Wikipedia is a topic indicating the characteristics of an entity so as to effectively browse and find entities. The category-based relatedness is based on the assumption that semantically related entities should share common topics. For example, *Chicago Bulls* is sharing several categories such as *NBA* and *basketball* with the basketball player *Michael Jordan*, but no categories with the researcher *Michael Jordan*. For entities in the KB, the category information can be easily extracted from the *category* section of an entity page, and also the *list* and *table* page.

When measuring the relatedness between e_i and the rest entities in E (which is not given a priori), every entity pair $< e_i, e_j >$ needs to be compared. Finding the optimal assignment is proven to be a NP-Hard problem (Cucerzan, 2007). To simplify the problem, Cucerzan proposed to union the categories of all mentions' candidates and set as the category set of E. In this way, measuring the coherence of entities in E is converted to the measurement of the relatedness between e_i and E.

Same as the local similarity, we built a category vector for entities, and compute the relatedness between entity e_i and the assignment E as following:

$$sr(e_i, E) =< e_i.T, \bar{E} > \quad (1)$$

in which $e_i.T$ is the category vector of entity e_i, and $\bar{E}$ is the category vector of the assignment E as defined above.

4.2 Hyperlink-based Relatedness

Though the category-based approach can somehow capture the relatedness of two entities, there are several issues with this approach. First, the category information in Wikipedia is not well organized and formatted, which results in many noisy categories for entities. Second, the naming convention of category is not well defined, so two categories indicating the same topic do not contribute to the relatedness. For example, *Michael Jordan* is in the category *Basketball players at the 1984 Summer Olympics* and its team player *Scottie Pippen* is in the category *Basketball players at the 1996 Summer Olympics*. Though *Michael Jordan* and *Scottie Pippen* are closely related, these two categories do not match literally and contribute to their relatedness measure.

To solve this issue, Milne and Witten (2008) proposes to measure the relatedness between entities using the hyper-link structure in Wikipedia. In their assumption, two entities are semantically related if they are linked from or link to a common entity. Formally, the relatedness is defined as following:

$$sr(e_i, e_j) = \frac{\log(\max(|A|, |B|)) - \log(|A \cap B|)}{\log(|W|) - \log(\min(|A|, |B|))} \quad (2)$$

in which A and B are the sets of Wikipedia pages linking to e_i and e_j respectively, and W is the whole Wikipedia page set.

Same as the category-based measure, finding the optimal assignment E is a NP-Hard problem. Thus, we seek an iterative heuristic approach. For each mention, we first rank the candidates based on their local similarity, and then choose the top N candidates (N is set experimentally). To measure the relatedness between e_i and the rest entities e_j in E, we temporarily set e_j to the top candidate, and compute the relatedness. For each candidate e_i of mention m_i, we measure their similarity with m_i according to formula 4, and re-ranking the candidates using the new similarity. This procedure is repeated until the ranking of candidates for every mention does not change any more. Then the top ranking entity is determined as the true entity for a mention.

5 NIL clustering

For queries whose true entity is not in the reference KB, an additional task is the NIL clustering which clusters these queries into different sets so that each set contains only queries referring to the same entity. This task is very similar to the Web People Search task except that the NIL clustering handles not only *person* but also *organization* and *location*. It is also a variation of the cross-document coreference resolution problem. Traditional approaches commonly cluster mentions by measuring their context similarity such as bag-of-words or NEs. These approaches could encounter the context sparsity problem in which finding unambiguous contexts for two mentions is difficult. In our implementation we explore a relational clustering approach for this task.

Our approach exploits the entity linking results for the NIL clustering. The first step is to construct a graph of mentions in each document in which the node is a mention and the edge is the co-occurrence relationship between mentions. The co-occurrence relationship could be extracted from the sentence level, paragraph level, or the document level. Note that the graph constructed now is disconnected from each other (only mentions in the same document are connected). As a pre-processing step, we connect the disconnected graph using the entity disambiguation results by linking mentions from different documents through their referred entities. Figure 1 shows a sample graph built over the document collection. As we can see from the figure, *Washington Redskins* in *doc1* is linked to *Redskins* in *doc2* because they both link to the entity *Washington Redskins* in the reference KB. This relationship can provide additional evidence for clustering the query mention *Ashburn* in *doc1* and *doc2*.

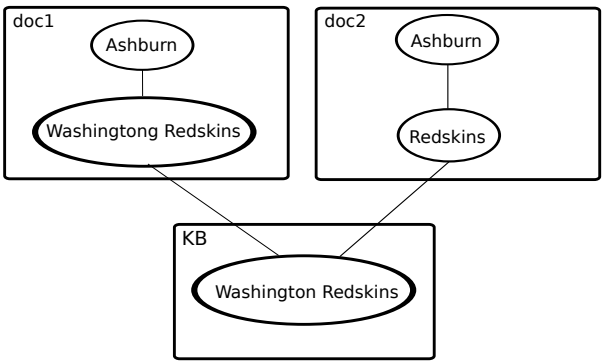

Figure 1: A sample graph built over documents and KB.

After building the graph, we then use the *agglomerative hierarchical clustering* for the NIL clustering. Algorithm 1 gives a high level description of the relational NIL clustering algorithm. The basic process is to perform pair-wise comparisons over mention pairs and merge mentions whose similarity is greater than a threshold. The similarity is measured as follow:

$$\begin{aligned} sim(m_i, m_j) = \alpha \quad * \quad & sim_A(m_i, m_j) \\ +(1-\alpha) \quad * \quad & sim_R(m_i, m_j) \end{aligned} \quad (3)$$

in which $sim_A(m_i, m_j)$ is the attribute similarity of m_i and m_j, and $sim_R(m_i, m_j)$ is their relational similarity. α is the weight.

For measuring the attribute similarity, we simply use the string similarity of the mention name (edit distance). Future work will add more lexical fea-

Algorithm 1 The relational clustering algorithm

Require: Graph $G = \{m_1, \ldots, m_n\}$
Ensure: A partition of G: $\{c_1, c_2, \ldots\}$, in which $c_i = \{m_{i1}, m_{i2}, \ldots\}$.

```
for all m_i ∈ G do
    for all m_j ∈ G(j ≠ i) do
        if sim(m_i, m_j) > threshold then
            merge(m_i, m_j)
        end if
    end for
end for
```

tures such as bag-of-words and NEs. For the relational similarity, we measure the Jaccard Coefficient over the neighbouring sets ($Nbr(m_i)$ and $Nbr(m_j)$ of the two mentions.

$$sim_R(m_i, m_j) = \frac{|Nbr(m_i) \cap Nbr(m_j)|}{|Nbr(m_i) \cup Nbr(m_j)|}$$

From the algorithm, we can see that the pair-wise clustering process requires a quadratic time complexity. To improve the performance, a blocking technique (McCallum et al., 2000) is employed to group similar mentions together and then compare mentions in the same block. Mentions in different blocks will not be compared so as to reduce the unnecessary comparisons and thus improve the efficiency.

6 Chinese Cross-lingual Entity Linking

Our Chinese cross-lingual EL system uses a translation model to translate Chinese queries and documents to English, and then apply the English EL system on the translated queries to generate the final results. The translation model is a combination of three parts: Pinyin transliteration, translation, and transliteration model. In the following, we first introduce the query processing, and then the document processing.

6.1 Query Processing

We use three bilingual dictionaries for our task. The first is the Chinese character to Pinyin dictionary (Pinyin dictionary), the second is LDC Chinese-English Translation Lexicon (LDC dictionary), and the last is a dictionary we gathered from the Chinese Wikipedia, for which we map the Chinese document title to its corresponding English document title (Wiki-dictionary).

The first step is to check if a query is a Chinese Wiki-title, if yes, the translation is the query's corresponding English Wiki-title. Otherwise, we emply ICTCLAS [4] (a Chinese segmentation tool) to segment the query. The Pinyin dictionary is used to translate characters if the segmentation is composed of unigrams and the first character is a possible Chinese surname according to LDC dictionary. Otherwise, we will combine both a transliteration model and a translation model. The results of both models are kept as translation input to the English EL system.

The transliteration model is based on the system DIRECTL+ (Jiampojamarn et al., 2010). DIRECTL+ is an online discriminative training system that incorporates joint n-gram features and many-to-many alignments generated by M2M-ALIGNER (Jiampojamarn et al., 2007). The system is trained on NEWs (Named Entity Workshop) 2012 Chinese to English data set.

The translation model is based on Moses [5], a statistical machine translation tool. We use the Wiki-dictionary as the training data. Other bilingual corpora such as LDC2004 Multiple Translation corpus are also explored. Our evaluation found that title dictionary performs the best. ICTCLAS is used for Chinese segmentation before training the alignment.

6.2 Document Processing

We also translate the document from Chinese to English. Different from the query process, after we segment the document using ICTCLAS, only words in the Wiki-dictionary are translated. Here we did not use any other dictionaries to increase the precision. Future work will explore other dictionaries for works not in the Wiki-dictionary.

7 Results

Below we report the evaluation results about our system.

[4] http://www.ictclas.org/
[5] http://www.statmt.org/moses/

	no-link	link-2	link-3	link-3+type	link-5	link-5+coref
Recall	0.755	0.949	0.941	0.938	0.898	0.924
Purity	33.16	42.10	36.14	31.90	29.46	31.66

Table 1: Candidate selection results on the TAC 2011 dataset.

7.1 Candidate Selection

Two metrics are defined to measure the performance of candidate selection as follows:

$$recall = \frac{\text{\# of queries with true candidates selected}}{N}$$

$$purity = \frac{\sum_1^N candsize(m_i)}{N}$$

in which N is the total number of queries, "# of queries with true candidates selected" is the number of queries whose true candidate is in their candidate set, and $candsize(m_i)$ is the number of entities in m_i's candidate set.

Here the *recall* measures how accurate the candidate selection is, and the purity measures the effectiveness of the system.

Table 1 shows the results of our candidate selection method with different configurations. Note that the evaluation dataset contains only the queries whose true entity is in the reference KB. In the experiment, we mainly check how the performance is affected by various factors: alias from the anchor text, their frequency, the type of entities, and the in-document coreference resolution. From the table, we can see that the anchor text is a valuable resource for the alias collection. Since there are many noisy aliases coming from the link source, we experiment with different frequency threshold to filter aliases, e.g. we only choose aliases from link with frequency higher than 3. As shown by the purity, higher threshold will filter out noisy aliases and slightly reduce the recall. For entity type, we found that simply filtering candidates by comparing the type is not practical since the accuracy of type extraction by NER is not high enough. Therefore, our system simply removes candidates whose type is not person, organization, location and miscellaneous. This strategy can remove many noisy candidates without sacrificing much recall. Finally, we found that in-document coreference resolution can greatly increase the recall (from 0.898 to 0.924).

In general, alias from anchor text is valuable, type filtering helps with reducing noisy aliases, and the in-document coreference resolution can increase the recall. One issue with our system is that the average number of candidates per mention is slightly high, which we will further investigate in the future.

7.2 Entity Disambiguation

	category	hyperlink
2010-eval	0.741	n/a
2010-train	0.688	0.778
2011-eval	0.660	0.714
2012-eval	**0.547**	n/a

Table 2: Entity Linking results on past TAC training and evaluation datasets.

Table 2 shows the results of category-based and hyperlink-based EL approaches on the past TAC datasets. Here we only evaluated the category-based approach at the time of submission. However, we further measure the hyperlink-based approach after the submission. As shown in the table, the hyperlink-based relatedness measure performed much better than the category-based measure. This is because the hyperlink structure in Wikipedia is a better indication for semantic relatedness (Milne and Witten, 2008). Also the low quality of Wikipedia category hierarchy has some negative impact on the performance of the category-based measure.

7.3 NIL Clustering

We did not conduct systematic evaluation for the NIL clustering, one reason is that the entity linking results largely depend on the performance of entity disambiguation rather than the NIL clustering. In the future, we will separately evaluate the NIL clustering algorithm.

	B^3+ $F1$
in KB	0.501
not in KB	0.598
NW docs	0.574
WB docs	0.494
PER	0.700
ORG	0.468
GPE	0.403
All	**0.547**

Table 3: Entity Linking results on the TAC 2012 dataset.

7.4 English Entity Linking on TAC-KBP 2012

Table 3 reports the performance of our EL system on the 2012 dataset. We can see that our system performs relatively better on the PER entity than the other two types. This is partially because ORG and GPE have a number of abbreviations and our system still cannot handle this case very well.

7.5 Chinese Cross-Lingual Entity Linking on TAC-KBP 2012

	B^3+ $F1$
in KB	0.329
not in KB	0.577
NW docs	0.434
WB docs	0.433
PER	0.474
ORG	0.325
GPE	0.485
All	**0.434**

Table 4: Chinese Entity Linking results on the TAC 2012 dataset.

Table 4 shows the results of our Chinese cross-lingual EL system. The result is much worse than the English EL task. The poor performance is mainly caused by the contexts (named entities translated from Chinese). We found that the transliteration and translation results from Chinese to English can hardly match the entities in the English Wikipedia KB, which results in a sparse context vector.

8 Conclusion

In this report, we described our system for the 2012 TAC-KBP English and Cross-Lingual Entity Linking task. We compared different candidate selection strategies and found that the candidate selection played an important role in an entity linking system: a good candidate selection method can significantly increase the accuracy and efficiency of entity linking. For entity disambiguation, entities are resolved collectively with both local and global features. In addition to the local context similarity, we implemented two semantic relatedness measures: a category-based measure and a hyperlink-based measure. We found that the hyperlink-based measure performed better than the category-based measure. We employed a relational clustering approach for the NIL clustering which can utilize the entity disambiguation results. Our Cross-Lingual EL system explored the translation and transliteration models; however, since the translated contexts from Chinese document do not match the context in the English Wikipedia, more accurate alignment is needed for the mapping.

References

Silviu Cucerzan. 2007. *Large-Scale Named Entity Disambiguation Based on Wikipedia Data*. In EMNLP-CoNLL'07, 708-716.

David N. Milne and Ian H. Witten. 2008. *An effective, low-cost measure of semantic relatedness obtained from Wikipedia links*. In WIKIAI'08.

Andrew McCallum, Kamal Nigam and Lyle H. Ungar. 2000. *Efficient clustering of high-dimensional data sets with application to reference matching*. In KDD'00, 169-178.

Sittichai Jiampojamarn, Colin Cherry and Grzegorz Kondrak. 2010. *Integrating Joint n-gram Features into a Discriminative Training Framework*. In HLT-NAACL'10, 697-700.

Sittichai Jiampojamarn, Grzegorz Kondrak and Tarek Sherif. 2007. *Applying Many-to-Many Alignments and Hidden Markov Models to Letter-to-Phoneme Conversion*. In HLT-NAACL'07, 372-379.