Abstract

Recent advancements in Open Information Extraction (OIE) techniques have resulted in the construction of enormous open knowledge bases (KB) containing millions of semantic assertions. In order to utilize open KBs for a range of Natural Language Processing tasks, one major problem that must be solved is entity resolution, which is determining whether two different noun phrases refer to the same entity. This project aims to develop a fast and accurate canonicalizer for entity resolution by blending techniques of word embeddings, deep bidirectional learning and side information. This report will discuss our current progress, which includes the analysis of the strengths and weaknesses of traditional canonicalization methods based on manually defined features, namely IDF Token Overlap, and the embedding-based method called CESI, together with our experiments with the state-of-the-art language model called BERT.
Table of Contents

1  Introduction 1

2  Related Studies 2
   2.1  GHMS 2
   2.2  CESI 3
   2.3  BERT 4

3  Methodology 5
   3.1  Datasets 5
   3.2  Architecture 5
   3.3  Evaluation Metrics 6
      3.3.1  Macro 6
      3.3.2  Micro 6

4  Results 7
   4.1  Canonicalization Result 7
   4.2  Analysis of Results 8

5  Future Work 8
   5.1  Multi-level Blocking 8
   5.2  Fine-tuning with BERT 9

6  Conclusion 10

References 11
List of Figures

1  GHMS Algorithm .......................... 2
2  CESI Workflow ............................ 3
3  BERT Structure and Performance ............ 4
4  BERT for Entity Ambiguity ................. 4
5  Dataflow from coarse to fine ............... 5
6  Multi-level Blocking Procedure .......... 9
7  Sentence Level Classification & Token Level Classification .......... 10

List of Tables

1  Evaluation Results for 3 Experiments ........ 7
**Abbreviations**

<table>
<thead>
<tr>
<th>Term</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>KB</td>
<td>Knowledge Base</td>
</tr>
<tr>
<td>ER</td>
<td>Entity Resolution</td>
</tr>
<tr>
<td>OIE</td>
<td>Open Information Extraction</td>
</tr>
<tr>
<td>NLP</td>
<td>Natural Language Processing</td>
</tr>
<tr>
<td>HAC</td>
<td>Hierarchical Agglomerative Clustering</td>
</tr>
<tr>
<td>CESI</td>
<td>Canonicalization using Embeddings and Side Information</td>
</tr>
<tr>
<td>BERT</td>
<td>Bidirectional Encoder Representations from Transformers</td>
</tr>
</tbody>
</table>
1 Introduction

A Knowledge base (KB) is a machine-readable library of information created for computer systems to perform a range of tasks, such as question answering and language inference. Any KB can be categorized as either a curated KB or an open KB by the way it is constructed. Curated KBs are commonly abstracted as semantic networks composed manually, while open KBs store semantic assertions in the form of \((\text{subject}, \text{relation}, \text{object})\) triples, directly derived from unstructured online resources [1]. Advancements in Open Information Extraction (OIE) techniques [2] have resulted in the construction of enormous open KB containing hundreds of millions of assertions. As open KBs do not require any manual maintenance like curated KBs do, many researchers have recently turned to open KB as a more comprehensive source of knowledge [3].

However, raw open KBs normally contain a huge amount of noisy, redundant and inconsistent information, which stems from two common problems called entity ambiguity and name variations. Entity ambiguity refers to the difficulty of identifying one among many entities that a certain name may refer to, while name variations refers to the challenge of linking and grouping different surface manifestations of the same real world entity [4]. These two problems make open KBs inapplicable for the Natural Language Processing (NLP) requests they set out to serve. Therefore, a pre-processing step of entity disambiguation, termed entity resolution (ER), is essential to make an open KB more precise, structured, complete, and thus more reliable for various tasks.

This project investigates how ER can be achieved via canonicalization, i.e. mapping each word or phrase into a canonical cluster. The objective is to build upon existing methods to develop a more accurate canonicalizer that can practically handle any given open KB. The remainder of this report proceeds as follows. First, we review some previous works on open KB canonicalization. We then present the core methodologies adopted by this project and justify these engineering choices. Next, we describe our current status and report the interim results. We close with a summary on project progress and future research plan.
2 Related Studies

This section discusses two widely used canonicalization systems and a state of the art language model which has the potential to handle entity ambiguity problem in canonicalization.

2.1 GHMS

Before embedding techniques were adopted, canonicalization was commonly performed by applying Hierarchical Agglomerative Clustering (HAC) over certain manually-defined features. The clustering requires a pairwise function that quantifies the similarity between two candidate entities. In 2015, Galarraga et al. (GHMS) [5] proposed a collection of similarity functions and compared their performances. GHMS mainly involves three steps: blocking, clustering and merging (see Figure 1). During canonicalization, GHMS algorithm first assigns assertions into canopies based on token comparison. It then runs the HAC algorithm to identify clusters inside each canopy and merge those that contain overlapped entities.

The best performing similarity function, studied by GHMS, was the standard Inverse Document Frequency (IDF) Token Overlap, a pairwise feature based on word frequency and string matching. More formally, for two mentions $m_1, m_2$ with subject names $n_1$ and

<table>
<thead>
<tr>
<th>Assertion</th>
<th>Subject</th>
<th>Relation</th>
<th>Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>Barack Obama</td>
<td>Grew up in</td>
<td>Honolulu</td>
</tr>
<tr>
<td>A2</td>
<td>Barack Obama</td>
<td>is the president of</td>
<td>US</td>
</tr>
<tr>
<td>A3</td>
<td>President Obama</td>
<td>Was born in</td>
<td>1962</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

![Figure 1: GHMS Algorithm](image)
the similarity function $f(m_1, m_2)$ is given by the Jaccard coefficient

$$f(m_1, m_2) = \frac{\sum_{w \in t(n_1) \cap t(n_2)} (\ln(1 + df(w)))^{-1}}{\sum_{w \in t(n_1) \cup t(n_2)} (\ln(1 + df(w)))^{-1}}$$

where $df(w)$ denotes the frequency at which the word $w$ appears in general texts. However, IDF token overlap is essentially a weighted string comparison which usually fails to handle the case where identities have the same meaning but different representations such as Mumbai and Bombay. Therefore, a more meaningful similarity measurement that encodes the contextual information is needed to improve the performance of canonicalization.

### 2.2 CESI

CESI [6] (Canonicalization using Embeddings and Side Information) is one of the first canonicalization systems that utilizes wording embedding methods. As shown in Figure 2, CESI first maps entities to high-dimensional space by assigning vector representation based on pre-trained GloVe [7] model and side information. It then uses HAC algorithm to identify clusters in the hyper-space.

![Figure 2: CESI Workflow](image)

Apart from using side information, the meat of CESI lies in the use of pre-trained GloVe model which leads to a better performance compared with GHMS algorithm. GloVe [7] is a widely used word2vec model [8] that is trained on Wikipedia data and encodes more than six billion words. Different from IDF token method, the GloVe embedding can successfully handle the Mumbai and Bombay problem since it also considers the contextual information. However, the GloVe model is trained with the assumption that
a single word can only have one vector representation. Thus, it cannot properly solve entity ambiguity such as the confusion between fruit Apple and company Apple.

2.3 BERT

Realizing the potential of word embedding in canonicalization, we then turn to a more powerful language model BERT [9] which has redefined the state of the art for 11 NLP tasks (see Figure 3). BERT model implements the concepts of transfer learning. Transfer learning is a learning strategy to learn a new task through the transfer of knowledge from a related task that has already been learned. In the BERT case, it learn enough pre-knowledge of language and can achieve satisfactory result on a specific downstream after fine tuning.

![Figure 3: BERT Structure and Performance](image)

![Figure 4: BERT for Entity Ambiguity](image)

Most importantly, BERT model demonstrates the potential of solving entity ambiguity problem. Different from the tradition RNN model which only encodes on a single direction, BERT introduces a new training technique named bi-directional training which better encodes the sentence structure and contextual information. Figure 4 shows that BERT model successfully distinguishes the fruit Apple and company Apple.
3 Methodology

We propose an end-to-end learning model that canonicalizes words and phrases in a coarse-to-fine fashion.

3.1 Datasets

For the evaluation of algorithms, we will use smaller datasets for efficiency and interpretability. ReVerb [10] is an OIE project that extracts 15 million assertions from the 500 million web pages that the ClueWeb09 corpus contains. A series of smaller datasets can be obtained by downsizing the 15M set via random sampling. To facilitate benchmarking and comparison, a smaller subset of ReVerb called ReVerb45K, which consists of 7.5K entities and 45K assertions, will be used as the standard dataset for extensive experiment and performance evaluation.

3.2 Architecture

![Diagram of dataflow from coarse to fine]

At the core of our approach is the notion of breaking canonicalization into two phases, each solved by a dedicated process.
The design of this two-phase architecture is motivated by our discussion in Section 2. CoarseNet is devised to learn word embeddings by utilizing the given KB as well as relevant side information. Since word embeddings techniques are powerful in capturing the semantics within a knowledge base, the first phase in our framework is to obtain a coarse taxonomy through clustering over learned embeddings. The classification after phase 1 is "coarse" because a single cluster may contain multiple entities. Currently, we are using the pretrained GloVe embeddings for phase 1.

In phase 2, we utilize sentence-level embedding and high-quality side information to further distinguish semantically-similar or string-similar entities that are organized into the same cluster after phase 1. Another focus of this filtration step is to solve the entity ambiguity problem by leveraging sentence embedding using BERT.

### 3.3 Evaluation Metrics

This section introduces two metrics for evaluating canonicalization results. A concrete evaluation example is presented at the end of this section. In this section, C denotes the canonicalization result to be evaluated and E denotes the actual canonicalization (also known as ground truth). F1 score is defined as the harmonic mean of precision and recall.

#### 3.3.1 Macro

Macro precision is the fraction of pure clusters in C, i.e., clusters that are true subset of ground truth clusters [6]. Macro recall is defined the same as macro precision except for the exchange of role of C and E.

$$ P_{macro}(C, E) = \frac{|\{c \in C : \exists e \in E : e \supseteq c\}|}{|C|} $$

$$ R_{macro}(C, E) = P_{macro}(E, C) $$

#### 3.3.2 Micro

Micro precision measures the purity of clusters in C, i.e., fraction of entities that appear in both canonicalization clusters and ground truth clusters based on the assumption that the majority of entities in a clusters are canonicalized correctly [6]. Micro recall is
defined as the same as micro precision except for the exchange of role of C and E.

\[
P_{micro}(C, E) = \frac{1}{N} \sum_{c \in C} \max_{e \in E} |c \cap e|
\]

\[
R_{micro}(C, E) = P_{micro}(E, C)
\]

4 Results

This section presents the canoncailization results for CESI, BERT and a mixed use of GloVe and BERT (our coarse-to-fine approach), as well as their evaluation.

4.1 Canonicalization Result

All three experiments were conducted under the same hardware condition on a computer equipped with Intel i7-7700HQ CPU, 8G of DDR4 memory and Linux operating system. Both CESI and GloVe + BERT methods were able to successfully canonicalize ReVerb45K [10] within 8G of memory and a descent time around 5 minutes. While the BERT method failed due to memory error during the first attempt. We conducted a second experiment on a machine with 16G memory, and the BERT method ended up using 13G of memory and taking over 30 minutes for a single canonicalization.

Table 1 shows the macro and micro score for the three experiments. The GloVe + BERT method achieves a very similar result with the CESI method both of which use the GloVe as their baseline. However, the two BERT models consistently underperformed compared with CESI across different metrics. A Macro F1 of 0.391, which is even lower than the GHMS method, indicates the impurity of result clusters.

<table>
<thead>
<tr>
<th></th>
<th>CESI</th>
<th>BERT</th>
<th>GloVe + BERT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Macro F1</td>
<td>0.6261</td>
<td>0.391</td>
<td>0.6260</td>
</tr>
<tr>
<td>Micro F1</td>
<td>0.8056</td>
<td>0.6378</td>
<td>0.7462</td>
</tr>
</tbody>
</table>

Table 1: Evaluation Results for 3 Experiments
4.2 Analysis of Results

Through the experiments, we confirmed the ability of GloVe embeddings in encoding the semantic similarity between known phrases. GloVe will continue to serve as our tool in phase 1 to produce a coarse classification of words and phrases.

In the two settings with BERT, we are essentially performing sentence resolution rather than entity resolution with the pretrained model, so the scores do not come at much surprise. One direct consequence is that different triples of the same entity can be found in several clusters. Also, BERT tends to group two entities together if the triples they are in share similar predicates and objects. For example, (‘onion’, ‘also contain generous amount of’, ‘vitamin’) can be found in the same cluster as (‘cilantro’, ‘be an excellent source of’, ‘vitamin’). However, with the sentence-level embeddings, BERT is the first model that has the potential of resolving the entity ambiguity issue. To fully release the power of BERT in ER tasks, the key lies in our ability to transform these sentence-level embeddings into equivalence indicators between entities.

5 Future Work

Based on the observation of experiments results, we identify two major problems. One is how to canonicalize entities with enough prior knowledge in the language models while not being affected by a single sentence structure, and the other is how to provide a good estimation for entities that language models have never seen before? This section discusses our solutions towards these two problems respectively.

5.1 Multi-level Blocking

As discussed in section 4.2, BERT sometimes emphasizes too much on sentence structure and mis-classifies entities with clearly different meanings but similar sentence structure. To reduce the influence of single sentence, we propose a multi-level blocking algorithm based on GHMS method. As shown in Figure 6, we first obtain the level 1 canopies (coarse cluster) by applying GloVe classification and then use GHMS blocking method to further divide them to level 2 canopies. We then extract the sentence vector with BERT for every sentence inside a level 2 canopy and take the average vector as the level 2 canopy vector. Since a level 2 canopy vector encodes the contextual information from all its candidates, it is usually a more comprehensive expression of the canopy instead of easily affected by a single sentence. In the final step, we run the
HAC algorithm to obtain the final result so that highly similar clusters have a chance to be merged back. Different from the GHMS method, our multi-level blocking approach can deal with the *Mumbai* and *Bombay* problem in ideal condition (see Figure 6 lower example).

![Multi-level Blocking Procedure](image)

**Figure 6: Multi-level Blocking Procedure**

5.2 Fine-tuning with BERT

To equip BERT with the necessary knowledge for our canonicalization task, we have proposed two downstream tasks inspired by the examples provided in BERT paper whose design has been proven to be valid and effective.

The first downstream task is the sentence level classification (see Figure 7 left example). This design is inspired by the MRPC task [11] which aims at determining whether two sentences are semantically equivalent. Our modification is to label the data based on subject equivalence instead of sentence equivalence. Given a sentence pair, this design will output a class label for the sentence pair with 0 indicates subject in-equivalence and 1 indicates subject equivalence. The other downstream task is the token level classification (see Figure 7 right example) which is inspired by the NER task [12]. We label the data based on token equivalence. The only difference from sentence level classification is that this design will output the label for every token in the sentence pair.
6 Conclusion

Entity resolution via canonicalization is a relatively new research area but of much interest. Most existing canonicalization methods suffer from either low accuracy or low efficiency problems which make them inapplicable to large scale open knowledge base. Other critical issues such as entity ambiguity, recall-precision trade-off also remain to be solved in this area.

Up till now we have produced promising results. Three different canonicalization methods, named CESI, BERT and GloVe + BERT, have been experimented. Both CESI and GloVe + BERT produces descent cluster results compared with the traditional GHMS method which cannot capture the contextual information. Although BERT fails to achieve a satisfying score, it still demonstrates the possibility of solving entity ambiguity problem. As a result, we consider the mixture use of GloVe and BERT to be promising and plan to use it as our baseline model for further improvement.

In the next phase, we plan to firstly implements the multi-level blocking method, aiming to reduce the influence imposed by sentence structure. Then, two fine-tuning models, sentence level classification and token level classification, will be experimented. These novel methods proposed, hopefully, will generate satisfactory results in the future.
References


