Entity Resolution and Location Disambiguation in the Ancient Hindu Temples Domain using Web Data

Ayush Maheshwari, Vishwajeet Kumar, Ganesh Ramakrishnan
Indian Institute of Technology Bombay
Mumbai, India
{ayushm,vishwajeet,ganesh}@cse.iitb.ac.in

J. Saketha Nath *
IIT Hyderabad
Hyderabad, India
saketha@iith.ac.in

Abstract

We present a system for resolving entities and disambiguating locations based on publicly available web data in the domain of ancient *Hindu Temples*. Scarce, unstructured information poses a challenge to Entity Resolution(ER) and snippet ranking. Additionally, because the same set of entities may be associated with multiple locations, Location Disambiguation(LD) is a problem. The mentions and descriptions of *temples*[1] exist in the order of hundreds of thousands, with such data generated by various users in various forms such as text (Wikipedia pages), videos (YouTube videos), blogs, *etc.* We demonstrate an integrated approach using a combination of grammar rules for parsing and unsupervised (clustering) algorithms to resolve entity and locations with high confidence. A demo of our system is accessible at tinyurl.com/templedemos[2]. Our system is open source and available on GitHub[3].

1 Introduction

Entity Resolution (ER) is the process of associating mentions of entities in text with a dictionary of entities. Here the dictionary might be either manually curated (such as Wikipedia) or constructed in an unsupervised manner (Bhattacharya and Getoor, 2007). It is a well studied problem with wide applications. This problem is of particular significance for domains in which the information available on the Web is relatively scarce.

In the domain of ancient *Hindu Temples*[4], which are present in the order of hundreds of thousands, the corresponding sources of information are often diverse and scarce. There are more than six hundred thousand temples in the country:however, sufficient information exists only for a few of them on the Web. Furthermore,

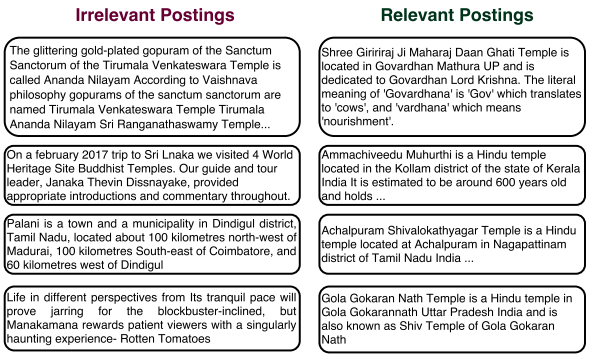

Figure 1: Sample descriptions (posts) from YouTube videos on various temples. Irrelevant posts are snippets which cannot be associated with any temple, whereas the relevant posts are about *Giriraj Dham, Ammachiveedu Muhurthi, Shivalokathyagar* and *Gorakhnath* temples respectively.

a significant fraction of such data (∼60%), is generated by the crowd over social multi-media platforms such as YouTube and Twitter. This data is ridden with subjective evaluations, opinions, and speculations. See Figure 1 for examples which we contrast with relatively objective and factual passages. The irrelevant posts in Figure 1 are speculative, subjective/opinionated or irrelevant. Our initial challenge is to weed out such speculative information carefully while holding on to sparse, factual and historical information. Additionally, the problem becomes more complex when the information about the domain is either poorly structured or unstructured. In Figure 2 we present an example snippet containing multiple *temple names* and multiple *temple locations*. We observe that a snippet can sometimes contain multiple mentions of similar *temple names* and *temple locations*. Due to similar temple names present at multiple locations, we also face the problem of Location Disambiguation (LD).

In this work, we present a novel approach to perform ER and LD for ancient temples using text and multimedia content publicly available on the Web. We retrieve information about temples from various sources such as Google Maps, YouTube *etc.*, and preprocess it. Using

*This work was done while author was at IIT Bombay

[1]Throughout the paper by 'temples' we mean entities in the domain of ancient Hindu Temples.

[2]Demo of the Snippet Ranking system can be accessed at tinyurl.com/entityr

[3]https://github.com/vishwajeet93/templeSearch

[4]Note: Here *Temple* is an entity with two attributes *viz.*, 1) Temple Name and 2) Temple Location

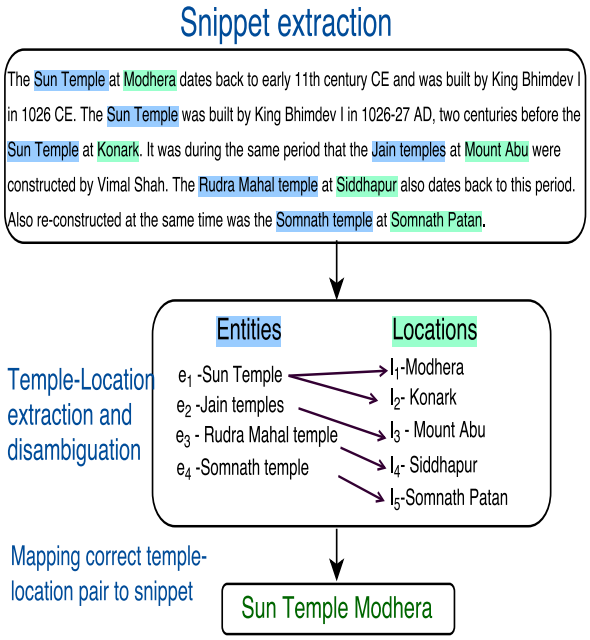

Figure 2: An example of temple name and location disambiguation. Snippet contain multiple mentions of same temple. Temple is present at multiple locations which makes disambiguation challenging.

this information, we extract videos present on YouTube about the temples along with its metadata such as title and description, and map videos to the corresponding temple. Next, we rank various textual snippets pertaining to the same temple on the basis of the relevance of each snippet. We demonstrate our approach through a system that accurately disambiguates videos to temples and ranks results in order of their relevance to a user query. We measure the effectiveness of our approach in terms of precision, recall and F-score. Our main contributions are as follows:

- A novel approach to perform ER and LD for temples using evidence from multiple snippets extracted from various web sources. Evidence may actually be subjective evaluations, opinions or speculations, not actual facts. We design heuristics with low false positive rates that help us filter out such misleading instances.
- A method to disambiguate temple and location names, and accurately associate relevant videos.
- A novel CNN-based(Convolutional Neural Network) technique to rank multiple snippets pertaining to the same temple by computing similarities.
- A system to search information (snippets, location, videos *etc.*) about temples.

2 Related Work

(Getoor and Machanavajjhala, 2013; Benjelloun et al., 2009; Wang et al., 2012) proposed to use crowd sourced data to resolve entities. (Wang et al., 2012) propose a hybrid human-machine approach to determine the most relevant, matching entity pairs. Instead of directly asking users to resolve entities, the system adopts a two-step approach; the system estimates the most likely mentions for an entity. This information is presented to the users to verify matching pairs of entities.
(Inkpen et al., 2017) proposed a set of heuristic rules to disambiguate location names in Twitter[5] messages. Their heuristics rely on geographical (latitude-longitude, geographic hierarchy) and demographic information (population of a region). (Awamura et al., 2015) used spatial proximity and temporal consistency clues to disambiguation location names. Our approach jointly resolves entity and disambiguate location names using publicly available web data.

3 Our Approach

We propose a novel technique to address the problem of entity resolution and location disambiguation. To extract the basic location and video data related to each temple, we use the Google Maps[6] and YouTube API[7] respectively. We disambiguate the name and location of each temple using publicly available data on the Web and leverage Google Maps to assign videos to the correct temple.

The *temples names* and *temple locations* are extracted from the snippets using text processing techniques. Thereafter, we use the *K*-medoids algorithm[8] to cluster snippets belonging to the same temple. Given a new temple, we retrieve the set of snippets related to the temple. These snippets are fed as input to a CNN based ranking system to score and rank snippets based on the queried temple.

3.1 Data collection

Most information pertaining to temples, as available on the Web is in the form of videos uploaded by individuals on video sharing websites such as YouTube, blogs, and Wikipedia pages. In most cases, the content uploaded by a user either (i) does not contain the specific name or location of the temple or (ii) contains multiple temple names and locations. In contrast, moderated content on sites such as Wikipedia is well-organized and contains unambiguous information.

Additionally, descriptions of temples are splintered over personal websites, Google Maps[9] and government websites, just to name a few sources. We crawled the Web to fetch mentions of temple names. We extracted

[5] https://www.twitter.com
[6] https://developers.google.com/maps/
[7] https://developers.google.com/youtube/
[8] https://en.wikipedia.org/wiki/K-medoids
[9] https://maps.google.com

temples and their locations from place annotations available on Google Maps. Through this, we were able to enlist over four hundred thousand temples located across the country. We use *temple name* and *location* to extract information about the temples present in YouTube[10] videos.

3.2 Temple Name and Location Disambiguation

We manually designed and wrote rules for parsing the textual data (from sources mentioned earlier) and extracted temple names. For this, we employed the JAPE grammar in the GATE tool (Cunningham et al., 2002). For illustration, consider the sentence: *The **Shankaracharya temple** is housed in the **Srinagar** district on the hill known as Takht-e-Suleiman*. In this illustration, the temple name and location (highlighted in bold) are extracted using manually composed parsing rules based on JAPE grammar.

Owing to user subjectivity, consistency and quality of the content varies widely. In our case, snippets within the corpus are replete with distinct mentions of the same entity. There are multiple variants of a single *temple name* in a single snippet. For example, *Vaishno Devi Mandir*, *Vaishno Devi Temple* or *shrine of Mata Vaishno Devi* are variants of the same *temple name*. To correctly attribute multiple variants to a single *temple name* (such as *Vaishno Devi Temple*), we pre-process these mentions and map them to a canonical temple entity by following a two-step approach. First, we build a vocabulary containing spelling variants and synonyms. As an example, **sh** and **h** are commonly used interchangeably (eg: *Shiva* and *Siva*). Similarly, *temple* and *mandir* are used interchangeably as synonyms (the latter being a word from Samskrit). Second, we wrote JAPE Grammar rules to parse temple names into their canonical forms. For instance, *Vaishno* uniquely identifies variants of the *Vaishno Devi Temple*. We follow a similar technique to disambiguate locations.

3.3 Mapping Videos to Temple

For most queries, videos retrieved in the top search results are unrelated to the temple name and its location. This leads to the need to map videos to a correct temple. User generated content needs to be analyzed and filtered to remove unrelated videos. We achieve this by fetching the top-15 videos for each temple and extracting their title and description. We store each title and description pair into a document, say d. We repeat this for each video-temple pair to form a set of document $D = d_1,d_2,...,d_n$. Below, we describe our approach to map videos to *temples* with high confidence.

1. Extract *temple name* and *temple location* from the document d_i using disambiguation methods.
2. Use Google Maps API to list *temple names* located around the extracted *temple location*. The *temple names* and *temple location* form a tuple t stored in set T.
3. For each element $t \in T$, we calculate TF-IDF score for tuple t over each document $d \in D$, where D is the indexed set of documents.
4. We rank documents based on TF-IDF scores for each query $t \in T$ and map the top ranked d to the temple.

[10] https://youtube.com

1: **input:** set of snippets S, mentions of temple name $t_1,t_2,..,t_n$ and location $l_1,l_2,..,l_n$
2: Build a vocabulary of t and l.
3: Add generic variants of t and l to the vocabulary
4: Apply JAPE Grammar rules to parse temple mentions to canonical forms
Clustering Algorithm
5: Form query set q as a cross-product of t and l. Each query will have two fields, *viz.*, 'temple:t, location:l'
6: Based on CNN similarity scores, generate top-k matches for every query in q on all snippets S
7: **for all** $s_i \in \mathcal{S}$ **do**
8: Assign membership score of each snippet s_j to s_i
9: Assign the top-k scoring snippets to cluster containing s_i
10: Identify snippets belonging to cluster C_i using score matrix
11: **output:** Snippets classification into c clusters

Figure 3: Pseudo-code for temple name and location disambiguation and clustering algorithm for processing textual snippets.

3.4 Snippet Clustering

Textual snippets retrieved from publicly available data on the web are pre-processed to remove stop words before giving input to the text processing engine. We use a CNN-based ranking method, explained in Section 3.5, which produces a score matrix for each snippet in the cluster. We label each cluster using a snippet that we determine to be the centroid of that cluster and select the corresponding *Temple name* and *location* pair that identifies the cluster. The score matrix is finally sorted to determine the top-k snippets belonging to that cluster. Pseudo-code for the clustering algorithm is described in Figure 3.

3.5 Snippet Ranking

We use a CNN-based architecture to score and rank snippets such that the CNN assigns the highest score to the snippet having maximum overlap with the queried temple. More formally the similarity between query

q and snippet **s** is computed as:

$$sim(q,s) = \mathbf{q}^T W \mathbf{s} \quad (1)$$

For our CNN model, we use the short text ranking system proposed by Severyn (Severyn and Moschitti, 2015). The convolution filter width is set to 5, the feature map size to 150, and the batch size to 50. We set the dropout parameter to 0.5. We initialized word vectors using pre-trained word embeddings. Before passing our input to the CNN, we pre-process the text and exclude plural nouns, cardinal numbers and foreign words from the snippets. Pre-processing helps us handle out of vocabulary words. We use the Stanford Part of Speech (POS) (Toutanova et al., 2003) tagger to annotate each word with its POS tag. As an example, consider the following input snippet: *Temple of Lord Somnath one of Jyotirlinga temple of Lord Shiva is situated near the town of Veraval in Western part of Gujarat whose present structure is built in 1951*. The PoS tagger annotates words like *Lord* , *Somnath*, *Shiva*, *Veraval*, *Gujarat* as proper nouns. We provide the query-temple pair as an input to the CNN which outputs the associated similarity score. The highest score represents the most relevant snippet for the temple.

4 Experiments and Results

4.1 Data Set

Our dataset[11] consists of more than four hundred thousand temple names with their locations extracted from Google Places. It also contains more than two hundred thousand videos fetched from YouTube.

Model (Values in %)	Ground Truth	
	Temple	∼Temple
Predicted as Temple	77	12
Predicted as ∼Temple	9	2

Precision = 0.863, Recall = 0.89, F = 0.876

Table 1: Precision, recall and F measure for the mapped entities

4.2 Results

We sample 1000 videos randomly from the complete video set to compute precision, recall and F-measure and evaluate the performance of videos mapped to the temple as shown in Table 1. 77% of YouTube videos are mapped to the correct temple with its location, 9% videos are mapped incorrectly. 12% videos are not mapped to any temple while 2% videos are false negatives. False negatives correspond to videos not relevant to a temple though retrieved from YouTube. Overall, we observe good performance in terms of precision and recall numbers, despite the association of a single temple name with multiple locations and despite the presence of multiple temples in the same location.

[11] Our annotated data is available for further academic research on request

5 Demonstration Details

When a user enters a query in the search box (annotated with Temple Search in Figure 4(b)), the system returns a list of temples. On selecting the temple, the system provides location annotations in the **Map** tab. The system also provides list of relevant videos for the query temple in the **Videos** tab (Figure 4(c)). In our Snippet ranking demo [12], the user can select a temple from the drop-down list and view the description of extracted snippets. Additionally, a user can view snippet clusters for a temple along with the snippet ranking score (as shown in Figure 4(a)).

6 Conclusion

In this paper, we focused on the problem of ER and LD in a domain where data is scarce, mostly unstructured and user generated. We presented a novel approach to disambiguate temple names and locations. We also addressed the problem of mapping videos to temples using ER and LD techniques. We leverage evidence from user generated content to map videos to their correct temple and rank snippets. We also presented a novel CNN-based technique for snippet clustering and ranking. Furthermore, we evaluated the effective-ness of our mapping techniques. In the future, we would like to resolve attributes such as the date of establishment, main deity, etc. from the ambiguous text.

References

Takashi Awamura, Daisuke Kawahara, Eiji Aramaki, Tomohide Shibata, and Sadao Kurohashi. 2015. Location name disambiguation exploiting spatial proximity and temporal consistency. In *Proceedings of the Third International Workshop on Natural Language Processing for Social Media*, pages 1–9.

Omar Benjelloun, Hector Garcia-Molina, David Menestrina, Qi Su, Steven Euijong Whang, and Jennifer Widom. 2009. Swoosh: a generic approach to entity resolution. *The VLDB JournalThe International Journal on Very Large Data Bases*, 18(1):255–276.

Indrajit Bhattacharya and Lise Getoor. 2007. Collective entity resolution in relational data. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):5.

Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. 2002. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *Proceedings of*

[12] tinyurl.com/entityr

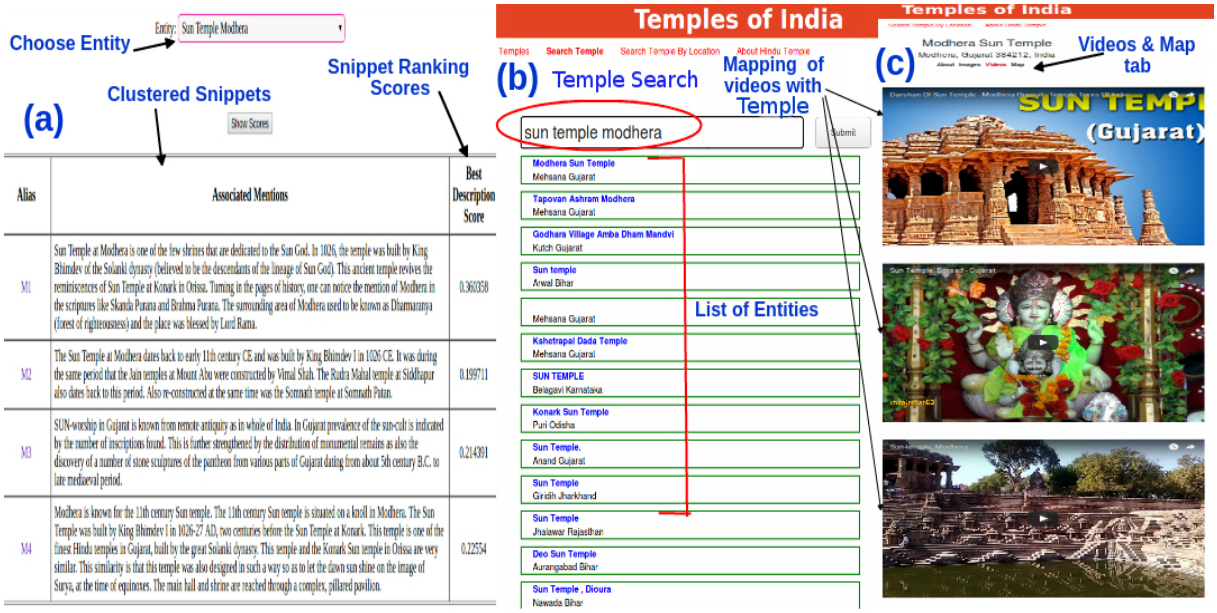

Figure 4: Snapshots of the system

the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02).

Lise Getoor and Ashwin Machanavajjhala. 2013. Entity resolution for big data. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1527–1527. ACM.

Diana Inkpen, Ji Liu, Atefeh Farzindar, Farzaneh Kazemi, and Diman Ghazi. 2017. Location detection and disambiguation from twitter messages. *Journal of Intelligent Information Systems*, 49(2):237–253.

Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 373–382. ACM.

Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics.

Jiannan Wang, Tim Kraska, Michael J Franklin, and Jianhua Feng. 2012. Crowder: Crowdsourcing entity resolution. *Proceedings of the VLDB Endowment*, 5(11):1483–1494.