Aligning facts in Wikidata to Sentences in Hindi Wikipedia

Vijay Vardhan Alluri * **Meghana Bommadi** **Swayatta Daw** **Shivprasad Sagare**

Project Mentor: Tushar Abhishek

Demo Video GitHub link Final Presentation

Abstract

Wikidata is a free and a community driven open knowledge base that can be read and edited by both humans and machines. It is a central storage repository containing structured data in the form of triples. All other Wikimedia projects use Wikidata as a central knowledge store of facts. Wikidata's knowledge is mostly available in a few languages, while most languages have close to no coverage. Most of the facts in Wikidata are stored in English language and there is no direct way of linking these facts to sentences present in Wikipedia articles written in other languages. In this project, we present multiple models using word overlap, vector similarity and transformers to tackle this problem of aligning the Wikidata triples to sentences in Hindi Wikipedia. Alignment of these facts (in English) to sentences in different languages will help in obtaining more accurate language labels for Wikidata entries making Wikidata entries more multi-lingual.

1 Introduction

Multilinguality is an important topic for knowledge bases, especially Wikidata because its labels are the way for humans to interact with the data (Kaffee et al., 2017). Wikidata contains linked data in the RDF format and can be queried via a SPARQL endpoint. Data is stored in the form of entities and each entity has multiple property labels attached to it. Adding one property label to an entity instantly makes thousands of statements more useful and valid. Thus, it could give various language communities access to knowledge they would not have been able to access before.

As mentioned earlier, multilinguality in Wikidata is very less and even languages spoken by large parts of the world population are not necessarily well covered. The main aim of this project is to enhance Wikidata by aligning English Wikidata facts to Hindi Wikipedia sentences. The system takes as input the data which consists of Wikidata triples as well as corresponding Hindi Wikipedia article's sentences and align them. Thus, the scope of the project is to build a model which retrieves the sentence corresponding to the queried triple. This will be the input and output of the system we aim to build.

2 Dataset

Based on our survey of the categories of articles present on Wikipedia, we have chosen to work on the 'Person' domain. The reason is that articles about people have most coverage in Hindi Wikipedia and would also contain more factual sentences with corresponding entries in Wikidata which would help us in generating more number of alignment mappings at the end. As the person domain is also huge, we further narrowed it down to sub-domains of 'Person' such as cricketers, actors and politicians. Concentrating on particular domains and subdomains not only helps us increase the performance of our models but also helps in doing a qualitative analysis on the results obtained as we can obtain useful insights. As there was no aligned data available, we had to create our own dataset.

2.1 Preprocessing

We extracted the articles from multiple subdomains using Hindi site links present in Wikidata for each entity and the triples using SPARQL query on that entity. All the entities are stored as a list and then looped upon using a Python script where a separate SPARQL query is fired which fetches all the triples of that entity. WDQS API might fail for some queries in between if there is increased load on the server or some other reasons. For that we employ an exception handling strategy in Python to skip the entity for which the API call is failed. Fortunately, not many calls failed in the process.

* All teammates have made equal contribution

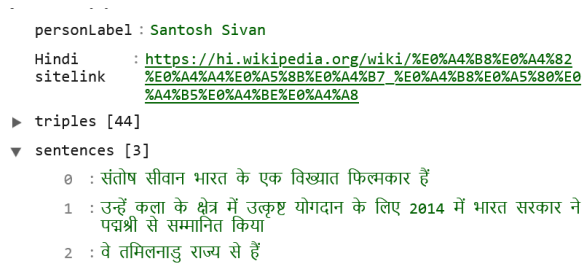

Figure 1: Snapshot of a web page in the web app

The triples are stored as JSON mapping with the Qid of that entity as a unique top level identifier. Following are some sample SPARQL queries.

```
SELECT DISTINCT ?person ?personLabel
    ?article WHERE {
  ?person wdt:P106 wd:Q82955;
        wdt:P31 wd:Q5.
  ?article schema:about ?person;
         schema:isPartOf
             <https://hi.wikipedia.org/>.

  SERVICE wikibase:label {
    bd:serviceParam wikibase:language
        "[AUTO_LANGUAGE],en".
  }
}
```

```
SELECT ?propertyLabel ?objectLabel
    WHERE {
  wd:%s ?predicate ?object.
  ?property wikibase:directClaim
      ?predicate.
  SERVICE wikibase:label {
    bd:serviceParam wikibase:language
        "[AUTO_LANGUAGE],en".
  }
}
```

After obtaining the triples, we have fetched the text content from Wikipedia articles and added it to the corresponding JSON objects. We have used the Hindi Wikipedia API to obtain the content of the article. Ex: Wikipedia API for "Kalpana." The Wikipedia API has provided us only the text content present in the article. We had then used sentence_tokenize() module from indic-nlp-library to tokenize the sentences from the article. Once the data is parsed, the text content would be appended to the same JSON structure. A sample json object corresponding to the entity "Santosh Sivan" could be seen in Figure 1. Each JSON object contains the person label, Hindi Sitelink, triples and sentences. In this example, the entity "Santosh Sivan" has 44 triples extracted from Wikidata and 3 sentences extracted from Hindi Wikipedia article.

We have done the data collection separately for each of the three sub-domains as it would help us later in debugging our alignment model on different sub-domains. Data for each sub-domain is stored in a separate JSON file. The entire data could be accessed from the GitHub repository.

Sub-domain	Entities fetched	After preprocessing
cricketers	2597	1912
actors	2504	1785
politicians	6337	5030

Table 1: Statistics of Wikidata triples fetched using Python SPARQL query API and later pre-processed

2.2 Exploratory data analysis

The fetched data consisted of entities with their corresponding triples and sentences. As it was fetched directly from the web, we performed an exploratory data analysis (EDA) to get insights into the data. We analyzed parameters like length of sentences, frequency of a specific triple across total entities, number of a sentence per entity, and number of triples per entity. After observing the central tendencies and spread of the above parameters, we decided to remove the data points with extreme or outlier values for the above parameters. We present our final dataset, which consists of more uniform values for the above parameters. The statistics of the dataset could be seen in Table 1.

2.3 Test data

For preparing test data to evaluate our methods, we manually annotate the data points by aligning the triples to sentences from the dataset obtained from above transformations. We decided to create the test data containing 400 sentences and their aligned triples. As this process of creating the test data needed a lot of manual efforts, we have built a web application which would help us in creating the test dataset efficiently with less scope to errors. The web app takes as input the initially created dataset in the form of json files. It then presents the user with a sentence and the list of triples corresponding to the entity with a checkbox attached to it. The annotator just needs to select the relevant triples which could be inferred from the sentence and hit the submit button. The web app would then automatically create a json object containing the sentence and the triples and store it in a file. We have deployed the web app on heroku and could be accessed by this link. We collectively

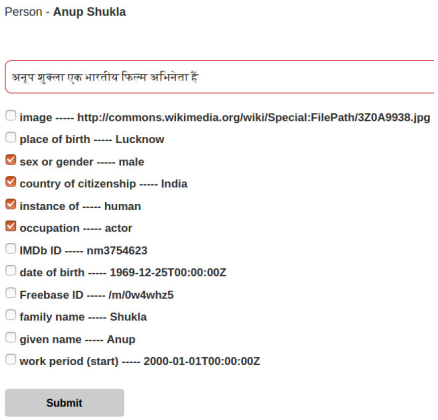

Figure 2: Snapshot of a web page in the web app

annotated 400 data points where each data point corresponds to a sentence from Wikipedia and the aligned triples from Wikidata. A snapshot of the webapp is shown in picture 2.

3 Experiments

As our baseline model, we had used static multilingual embeddings to align the English triples and Hindi sentences in common vector space. We use MUSE from Facebook Research for this. MUSE is a Python library for multilingual word embeddings, whose goal is to provide the community with: state-of-the-art multilingual word embeddings (fastText embeddings aligned in a common space) and a large-scale high-quality bilingual dictionaries for training and evaluation.

We have used :

- MUSE pre-aligned (En-Hi) embeddings made available by Facebook.
- We also train our own alignments(En-Hi) using MUSE using a train bilingual dictionary (or identical character strings as anchor points), to learn a mapping from the source to the target space.

The baseline approaches we developed are explained below.

3.1 Word overlap

For this method, we have used the MUSE pre-trained aligned multilingual embeddings made publicly available by Facebook. We obtain the multilingual word embeddings in Hindi for every word in Hindi sentences. Then, using k-nearest neighbours algorithm(kNN), we find the most semantically similar English words to these Hindi words.

```
Nearest neighbors of "अभिनेता":
0.4929 - actor
0.4046 - actors
0.3443 - actress
0.3031 - actresses
0.2843 - film
```

Figure 3: Top 5 similar words are used for finding the word overlap. This improves the flexibility as compared with direct string matching of only one particular word.

We use k = 5 to get 5 most similar English words for each word in the Hindi sentence. For out of vocab words, which are mostly in English, we use google translate for direct transliteration. We keep all these words in a list and then compare these English words with the words in entity triples consisting of both predicate and object. We obtain a score for the amount of overlap of the words with each triple. We set a threshold score, and return the triples which have a greater score than the threshold.

3.2 Vector similarity

For this method, we have used our own trained aligned Multilingual embeddings using the MUSE bilingual dictionary. Using these, we obtain word embeddings for every word in a Hindi sentence. To get sentence level embeddings, we simply take the average of the embeddings of all the words in the given sentence. For the triples, we average the word embeddings for every word in the triple. Then we find the cosine similarity between the sentences and the triples for each entity. We set a threshold of 0.5 and return the triples which have a greater score than the threshold, for each sentence. The results obtained for both the baseline models are show in Table 2

	Word Overlap		Vector Similarity	
Sub-domain	Prec	Recall	Prec	Recall
actors	0.432	0.825	0.321	0.508
cricketers	0.385	0.821	0.563	0.608
politicians	0.378	0.606	0.322	0.263
Average	0.3985	0.75	0.3825	0.48

Table 2: Precision and recall values corresponding to each sub-domain for both methods.

3.3 Transformers

After completing the baseline models, we went on to explore dynamic transformer based models in order to utilize the contextual information to obtain semantic similarity between Hindi sentences and the Wikidata triples. Similar to the baseline vector similarity approach, we treat the (predicate + object) in the entity triple as an English sentence and compare the similarity with the Hindi sentence. We have set a threshold to classify a triple aligning with a sentence. We further explore two types of approaches to create sentence level representation - Mean pooling and Sentence-BERT.

3.3.1 Mean pooling

We get the mean of the embeddings of all the tokens using "BERT-base-multilingual-cased" pre-trained model. It is pre-trained on the top 104 languages with the largest Wikipedia using a masked language modelling(MLM) objective. However, the result obtained for multilingual semantic similarity is sub-par as investigated in (Reimers and Gurevych, 2019)

3.3.2 Sentence-BERT

The sentence BERT framework (Figure 4) uses pre-trained transformer models like BERT/XLM-R/distilBERT and obtains the pooled output for two input sentences, and then fine-tunes it via a Siamese architecture, basically concatenating the two sentences and passing it through a soft-max classifier to get the classification loss function. The aim is to train the model on pairs of sentences to obtain the semantic similarity between two sentences. So the result is, similar sentences are mapped closer to each other and dissimilar sentences and mapped farther.

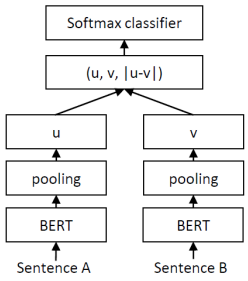

Figure 4: S-BERT architecture with a classification objective function for fine-tuning on a labeled dataset. The two BERT networks have tied weights

A high-level overview of the training process which was used to achieve this is shown in Figure 5. Parallel sentences in different languages with the same meaning is fed as the training data. The Teacher Model is Sentence-BERT which creates sentence representation of the English (source) sentence. The student model is any multilingual model. The objective is that the student's sentence representations for both languages in the pair to be close to the teacher's embedding in the source language.

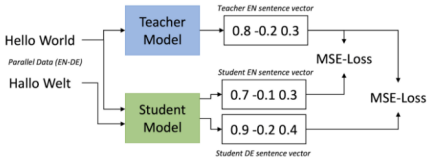

Figure 5: Schematic overview of knowledge distillation training process proposed by Reimers et al.

The entire frame-work is available as an open source library and we have used the same to create our multilingual sentence representations.

3.4 T-REx

T-REx (Text/Triple Relation Extraction) is a dataset of large scale alignments of Wikipedia Abstracts with Knowledge Base facts represented in Wikidata Triples. (Elsahar et al., 2019) It consists of three types of alignments made by three automatic alignment hypotheses. A crowd-sourcing experiment on 2,600 alignments exhibits a 97.8% accuracy with a high inter annotator agreement (ranging from 0.854 to 0.962 depending on the triple aligners used in the process). T-REx also provides an extensive evaluation through a crowd-sourcing experiment, through which T-REx showed to have high quality alignments reaching 97.8% accuracy.

3.4.1 Dataset

It consists of 11 million triples aligned with 3.09 million Wikipedia abstracts (6.2 million sentences). T-REx is two orders of magnitude larger than the largest available alignments dataset and covers 2.5 times more predicates. Figure 6 shows the alignment accuracy of top occurring predicates along side with inter annotator agreement. T-REx data set contains the abstracts from Wikipedia and the Wikidata URLs tagged to them including sentence boundaries with the sentence triples and each word with it's surface form, corresponding Wikidata URL, and the corresponding annotator.

Property Label	AllEnt	SPO	NoSub	Inter ann.
located in	0.95	1.00	1.00	0.90
member of sports team	1.00	0.99	0.99	0.97
date of birth	1.00	1.00	1.00	0.97
date of death	1.00	1.00	0.99	0.98
country of citizenship	0.91	1.00	0.95	0.92
educated at	0.88	0.92	1.00	0.92
occupation	0.90	0.94	1.00	0.93
spouse	0.75	0.94	1.00	0.92
capital	**0.40**	**1.00**	n.a.	0.82
shares border with	**0.14**	**1.00**	n.a.	**0.69**

Figure 6: Accuracy of top properties for each annotation methodology in T-REx

3.4.2 Process

We pre-processed the T-REx dataset and extracted the sentences with three triples tagged (subject, predicate, object). We found the common Wikidata articles in both the datasets (the dataset we created and the T-REx dataset) by using the "doc-id", "entity_id" in Hindi json data and T-REx data respectively. We then created a transformer based cross lingual sentence embedding system which can take a sentence in a language and create an embedding in a multilingual space. Based on the document id, we collected the respective sentences in Hindi (from our data) and English (from T-REx). Using the embedding model, we found the similarities of the extracted English and Hindi sentences with common Wikidata document ids. We matched sentences that have cosine similarity of more than 70% in the multilingual vector space. Using the English sentences we found the triples that matched it and aligned it with the corresponding Hindi sentences. We can then align the Hindi sentence to the T-REx triple of the English Sentence.

4 Other approaches

Apart from the experiments mentioned above, we had thought of a few other approaches to tackle the problem. A few of the approaches and why we chose not to implement them are explained below.

1. **Encoding triplets using Knowledge Graph Embeddings** - This approach involves preparing embeddings for the entities and relationships for each triple in a vector space using the methods which leverage the knowledge graph structure. Due to time constraint and other better approaches, we decided to not implement this.

2. **Named Entity Recognition (NER) matching** - This approach involves Named Entity Recognition from the Hindi sentences. After we extract the subject and object from sentences, we translate the words to English and then perform word overlap with Wikidata triples for finding most similar mappings. Implementing NER felt like a little digression from the problem statement and hence, we did not consider this.

3. **Translation** - This approach involves translation of Hindi sentences to English using English embeddings or translation of English triples to Hindi and then aligning them without using multilingual embeddings. Though this approach seems easy, it did not promise good results because the translation models are not very accurate. Also, the problem would then reduce to aligning English sentences to English triples which would take away the multilinguality aspect. So, we chose to not implement this approach.

5 Evaluation Metrics

The output of our alignment model is a set of triples corresponding to the queried sentence. We decided to use the evaluation metrics of precision and recall to compare the predicted triples with those present in test data. Precision and Recall are standard metrics in the task of information retrieval and preferred over accuracy in general. Precision is the measure of how many triples our model correctly predicted over the number of correct and incorrect predictions of triples. Recall is the measure of how many triples our model correctly predicted over the total number of correct triples. Out of these two, we concentrated on increasing the recall of the model as we wanted to prioritise not missing a correct triple eventhough we have a few false positives.

6 Results

We use multiple pre-trained multilingual models (XLM-R(paraphrase), XLM-R(STS), quora-distilBERT, multilingual Universal Sentence Encoder) and compare the performances on our dataset. We use two variants of pre-trained XLM-R, trained on paraphrase and on STS dataset. The results are presented in Table 3

	mBERT (mean-pooling)		ML distilBERT-base		ML universal SE		XLMR (STS)		XLMR paraphrase	
	P	R	P	R	P	R	P	R	P	R
actors	0.123	0.648	0.139	0.496	0.554	0.488	0.384	0.501	0.407	0.624
cricketers	0.240	0.628	0.225	0.493	0.321	0.416	0.441	0.679	0.474	0.770
politicians	0.137	0.7	0.207	0.55	0.592	0.383	0.381	0.616	0.334	0.875
average	0.167	0.658	0.19	0.515	**0.489**	0.429	0.402	0.599	0.485	**0.796**

Table 3: Precision and recall values corresponding to each sub-domain with corresponding models

7 Qualitative analysis

Beyond the quantitative results of precision and recall for each of the model, we also perform the qualitative analysis based on the output alignments based on our two models, word overlap and XLM-R vector similarity. We look at the individual examples in test data and analyze the quality of output triples. We got to know some meaningful insights about the data through it and also the several edge cases which occur in this generic alignment model. We note down the observations so that it can be useful in future work for anyone working on this problem statement. Some insights worth noting are illustrated below.

One observation was that the number of output triples for given sentence was higher than usually number of triples annotated by us. It can be mitigated by increasing the threshold, so that less number of triples are outputted. But it would adversely affect the recall of our system as less number of matching triples would be outputted. This needs to be taken care of in future iterations. Another insight we found in data was that single predicate has multiple values as object. The decision as to which to include from them can be tricky even for human annotators.For e.g. occupation predicate has multiple triples corresponding to values like film actor, television actor, actor. This point is worth investigating too in further iterations. One more issue which exists at data level is occurrence of some mundane and less useful triples in dataset. For e.g. id, surname, creative commons, twitter username etc. This needs to be carefully eliminated at the time of pre-processing itself.

An important observation regarding the structure of Hindi sentences is that it may not necessarily contain the words in triples in a neatly verbalized form. For e.g. the nationality-Indian predicate-value pair can be simply verbalized as 'X is Indian'. It skips the word 'nationality' in the sentence. Hence, it might not be captured by word overlap methods we implement. Similarly, we come across words in Hindi sentence which are morphological compositions of root word. Hence, the possibility of those words existing in vocabulary of our multilingual embedding models is very low. A solution on this can be to stem those words to root form in Hindi language. Also, an important decision is to decide whether to translate or transliterate a particular word which is out of vocabulary. Because, some words in English have become so common in India that they are used as it is just in Devanagari script. At such instances, it is required to transliterate them back in English for valid comparison. These points of analysis will serve as a direction for future work to output better aligned mappings.

8 Conclusion

We worked on the challenging research problem of cross-lingual triple-to-sentence alignment. We formulated the task for English wikidata triples and Hindi sentences. After fetching and pre-processing of the data, we experimented with several approaches to determine the alignment between english triple and Hindi sentence. We employed static multilingual embeddings MUSE as well as transformer based contextual models for encoding the triples and sentences. We evaluate the baseline models of word-overlap and vector similarity using the measures of precision and recall. Finally, we do some qualitative analysis by looking closely at our output results and propose the directions for future work.

Acknowledgments

We would like to thank Tushar Abhishek for his constant mentorship and support throughout the course of the project. We also thank Prof. Vasudeva Varma for giving us an opportunity to work on this project and other batchmates for giving us their valuable feedback.

References

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Elena Simperl, and Frederique Laforest. 2019. T-rex: A large scale alignment of natural language with knowledge base triples.

Lucie-Aimée Kaffee, Alessandro Piscopo, Pavlos Vougiouklis, Elena Simperl, Leslie Carr, and Lydia Pintscher. 2017. A glimpse into babel: an analysis of multilinguality in wikidata. In *Proceedings of the 13th International Symposium on Open Collaboration*, pages 1–5.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.