Transliteration Better than Translation? Answering Code-mixed Questions over a Knowledge Base

by

Vishal Gupta, Manoj Chinnakotla, Manish Shrivastava

in

56th Annual Meeting of the Association for Computational Linguistics
(ACL-2018)

Melbourne, Australia

Report No: IIT/TR/2018/-1

Centre for Language Technologies Research Centre
International Institute of Information Technology
Hyderabad - 500 032, INDIA
July 2018
Transliteration Better than Translation? Answering Code-mixed Questions over a Knowledge Base

Vishal Gupta  
IIIT Hyderabad, India  
vishal.gupta@research.iiit.ac.in

Manoj Chinnakotla  
Microsoft, Bellevue, USA  
manojc@microsoft.com

Manish Shrivastava  
IIIT Hyderabad, India  
m.shrivastava@iiit.ac.in

Abstract

Humans can learn multiple languages. If they know a fact in one language, they can answer a question in another language they understand. They can also answer Code-mixed (CM) questions: questions which contain both languages. This ability is attributed to the unique learning ability of humans. Our task aims to study if machines can achieve this. We demonstrate how effectively a machine can answer CM questions. In this work, we adopt a two-step approach: candidate generation and candidate re-ranking to answer questions. We propose a Triplet-Siamese-Hybrid CNN (TSHCNN) to re-rank candidate answers. We show experiments on the SimpleQuestions dataset. Our network is trained only on English questions provided in this dataset and noisy Hindi translations of these questions and can answer English-Hindi CM questions effectively without the need of translation into English. Back-transliterated CM questions outperform their lexical and sentence level translated counterparts by 5% & 35% respectively, highlighting the efficacy of our approach in a resource-constrained setting.

1 Introduction

Question Answering (QA) has received significant attention in the Natural Language (NLP) community. There are many variations (open-domain, knowledge bases, reading comprehension) as well as datasets (Joshi et al., 2017; Hopkins et al., 2017; Rajpurkar et al., 2016; Bordes et al., 2015) for the question answering task. However, many approaches (Lukovnikov et al., 2017; Yin et al., 2016; Fader et al., 2014; Chen et al., 2017a; Hermann et al., 2015) attempted in QA so far have been focused on monolingual questions. This is true for both methods and techniques as well as resources.

Code-mixing (referred to as CM) refers to the phenomenon of “embedding of linguistic units such as phrases, words and morphemes of one language into an utterance of another language” (Myers-Scotton, 2002). People in multilingual societies commonly use code-mixed sentences in conversations (Grover et al., 2017), to search on the web (Wang and Komlodi, 2016) and to ask questions (Raghavi et al., 2017). However, current Question Answering (QA) systems do not support CM and are only designed to work with a single language. This limitation makes it unsuitable for multilingual users to naturally interact with the QA system, specifically in scenarios wherein they do not know the right word in the target language.

CM presents serious challenges for the language processing community (Çetinoğlu et al., 2016; Vyas et al., 2014), including parsing, Machine Translation (MT), automatic speech recognition (ASR), information retrieval (IR) and extraction (IE), and semantic processing. Even for problems such as language identification, or part of speech tagging, that are considered solved for monolingual languages, performance degrades when mixed-language is present. Lack of language resources such as annotated corpora, part-of-speech taggers and parsers poses a considerable challenge for automated processing and analysis of CM languages. This further amplifies the challenge for CM QA. This CM question answering task
is challenging not just because of having multiple languages with different semantics but also because of the different word order of source language and CM, making it difficult to extract essential features from the input text.

We base our work on the premise that humans can answer CM questions easily provided they understand the languages used in the question. They require no additional training in the form of CM questions to comprehend a CM question. So, one way to tackle CM questions is to translate them into a single language and use monolingual QA systems (Lukovnikov et al., 2017; Yin et al., 2016; Fader et al., 2014). Machine Translation systems perform poorly on CM sentences. The only other viable option is lexical translation (word by word translation). Lexical translation requires language identification, which Bhat et al. (2018) show to be solved. We show that our model trained on both English and Hindi can perform better on CM question directly than its lexical translation. This removes the need to obtain a large bilingual mapping of words for lexical translation. Also, such a sizeable bilingual mapping may be hard to obtain for low-resource languages.

Knowledge Bases (KBs) like Freebase (Google, 2017) and DBpedia contain a vast wealth of information. Information is structured in the form of tuples, i.e. a combination of subject, predicate and object (s, p, o) in these KBs. Such KBs contain information predominantly in English, and low resource languages tend to lose out on having a rich information source.

We use bilingual embeddings to fill the gaps due to lack of resources. We also develop a K-Nearest Bilingual Embedding Transformation (KNBET) which exploits bilingual embeddings to outperform the performance of lexical translation.

We overcome challenges discussed above in our paper and develop a CM QA system over KB, named CMQA, using only monolingual data from individual languages. We demonstrate our system with Hinglish (Matrix language: Hindi, Embedded language: English) CM questions. Our evaluation shows promising results given that no CM data was used to train our model. This shows promise that we do not need CM data but can use monolingual data to train a CM QA system. Our results show that our system is much more useful as compared to translating a CM question.

Our contributions are as follows:

1. We show how we can answer CM questions given an English corpus, noisy Hindi supervision and imperfect bilingual embeddings.

2. We introduce a Triplet-Siamese-Hybrid Convolutional Neural Network (TSHCNN) that jointly learns to rank candidate answers.

3. We provide a test dataset of 250 Hindi-English CM questions to researchers. This dataset is mapped with Freebase tuples and English questions from the SimpleQuestions dataset.

To the best of our knowledge, we are the first to tackle the problem of End-to-End Code-Mixed Question Answering over Knowledge Bases in a resource-constrained setting. Earlier approaches for CM QA (Raghavi et al., 2017) require a bilingual dictionary to translate words to English and an existing Google like-search engine to get answers, which we do not require.

The rest of the paper is structured as follows: We survey related work in Section 2 and describe the task description in Section 3. We explain our system in Section 4. We describe experiments in Section 5 and provide a detailed analysis and discussion in Section 6 and conclude in Section 7.

2 Related Work

Question Answering and Knowledge Bases Question answering is a well studied problem over knowledge bases (KBs) (Lukovnikov et al., 2017; Yin et al., 2016; Fader et al., 2014) and in open domain (Chen et al., 2017a; Hermann et al., 2015). Learning to rank approaches have also been applied to QA successfully (Agarwal et al., 2012; Bordes et al., 2014). Many earlier works (Ture and Jojic, 2017; Yu et al., 2017; Yin et al., 2016) which tackle SimpleQuestions divide the task into two steps: mention detection and relation

\[\text{http://dbpedia.org/}\]
prediction, whereas we jointly do both using our model. Lukovnikov et al. (2017) is more similar to our approach wherein they train a neural network in an end-to-end manner.

**CodeMixing and CodeSwitching** Code-mixing and code-switching has recently gathered much attention from researchers (Bhat et al., 2018; Rijhwani et al., 2017; Raghavi et al., 2015, 2017; Banerjee et al., 2016; Dey and Fung, 2014; Bhat et al., 2017). CM research is mostly confined towards developing parsers and other language pipeline primitives (Bhat et al., 2018, 2017). There has been some work in CM sentiment analysis (Joshi et al., 2016). Raghavi et al. (2015) demonstrate question type classification for CM questions and Raghavi et al. (2017) also demonstrate a CM factoid QA system that searches for the lexically translated CM question using Google Search on a small dataset of 100 CM questions. To the best of our knowledge, there has been no work on building an end-to-end CM QA system over a KB.

**Bilingual Embeddings** Recent work has shown that it is possible to obtain bilingual embeddings using only a minimal set of parallel lexicons (Smith et al., 2017; Artetxe et al., 2017; Ammar et al., 2016; Luong et al., 2015; P et al., 2014) or without any parallel lexicons (Zhang et al., 2017; Conneau et al., 2017). Our approach, can use these bilingual embeddings and supervised corpus for a resource-rich language, to enable CM applications for resource-poor languages.

**Cross-lingual Question Answering** Closely related is the problem of cross-lingual QA. There have been various approaches (Ahn et al., 2004; Lin and Kuo, 2010; Ren et al., 2010; Ture and Boschee, 2016) to cross-lingual QA. Some approaches (Lin and Kuo, 2010) rely on translating the entire question. Others (Ren et al., 2010), have also explored using lexical translations for this task. Recently, Ture et al. (Ture and Boschee, 2016) proposed models that combine different translation settings. There have been some efforts (Pouran Ben Veyseh, 2016; Hakimov et al., 2017; Chen et al., 2017b) to attempt cross-lingual question answering over knowledge bases.

### 3 Task Description

The SimpleQuestions task presented by Bordes et al. (2015) can be defined as follows. Let $\mathcal{K} = \{(s_i, p_i, o_i)\}$ be a knowledge base represented as a set of tuples, where $s_i$ represents a subject entity, $p_i$ a predicate (also referred as relation), and $o_i$ an object entity. The task of SimpleQuestions is then: Given a question represented as a sentence, i.e. a sequence of words $q = \{w_1, \ldots, w_n\}$, find a tuple $(\hat{s}, \hat{p}, \hat{o}) \in \mathcal{K}$ such that $\hat{o}$ is the correct answer for question $q$. This task can be reformulated to finding the correct subject $\hat{s}$ and predicate $\hat{p}$ that question $q$ refers to and which characterise the set of triples in $\mathcal{K}$ that contains the answer to $q$.

Consider the example, given question “Which city in Canada did Ian Tyson originated from?”, the Freebase subject entity m.041fft representing the Canadian artist Ian Tyson and the relation fb:music/artist/origin, can answer it.

### 4 Our System: CMQA

In this section, we describe our system which consists of two components: (1) the Candidate Generation module for finding relevant candidates and (2) a Candidate Re-ranking model, for getting the top answer from the list of candidate answers.

#### 4.1 Candidate Generation

Any freebase tuple (specifically, the object in a tuple is the answer to the question) can be an answer to our question. We use an efficient (non-deep learning) candidate retrieval system to narrow down our search space and focus on re-ranking only the most relevant candidates. Solr\(^2\) is an open-source implementation of an inverted index search system. We use Solr to index all our freebase tuples (FB2M) and then query for the top-k relevant candidates given the question as a query. We use BM25 as the scoring metric to rank results. Since we index freebase tuples which are in English (translating the entire KB would require a very large amount of effort and we restrict ourselves to using only the provided English KB), any non-English word in the query does not contribute

\(^2\)http://lucene.apache.org/solr/
to the matching. This is a limiting factor in candidate generation for CM questions.

4.2 Candidate Re-ranking
We use Convolutional Neural Networks (CNNs) to learn the semantic representation for input text (Kim, 2014; Hu et al., 2015; Lai et al., 2015; Cho et al., 2014; Johnson and Zhang, 2015; Zhang et al., 2015). CNNs learn globally word order invariant features and at the same time pick order in short phrases. This ability of CNNs is important since different languages have different word orders.

Retrieving a semantically similar answer to a given question can be modelled as a classification problem with a large number of classes. Here, each answer is a potential class and the number of questions per class is small (Could be zero, one or more than one. Since we match only the subject and predicate, there could be multiple questions having a common subject and predicate combination). An intuitive approach to tackle this problem would be to learn a similarity metric between the question to be classified and the set of answers. We find Siamese networks have shown promising results in such distance-based learning methods (Bromley et al., 1993; Chopra et al., 2005; Das et al., 2016).

Our Candidate Re-ranking module is inspired by the success of neural models in various image and text tasks (Vo and Hays, 2016; Das et al., 2016). Our network is a Triplet-Siamese Hybrid Convolutional neural network (TSHCNN), see figure 1. Vo and Hays (2016) show that classification-siamese hybrid and triplet networks work well on image similarity tasks. Our hybrid model can jointly extract and exchange information from the question and tuple inputs.

All convolution layers share weights in TSHCNN. The fully connected layers are also Siamese and share weights. This weight sharing helps project both questions and tuples into a similar semantic space and reduces the required number of parameters to be learned.

*Additional Input: Concatenate question + tuple* Our initial network only had two inputs (question and tuple) to each corresponding branch. We further modify our network to provide a third input in the form of the concatenation of question and tuple. This additional input helps our network learn much better feature representations. We discuss this in the results section.

As shown in figure 1, questions and candidate tuples are provided to our system. Our experiments vary in the input questions (English and CM variations of questions), but the candidates (tuples or answers) are always in monolingual English. Thus our final answer is always in English.

---

3English is SVO whereas Hindi is free word order. SVO means Subject Verb Object.
4.2.1 K-Nearest Bilingual Embedding Transformation (KNBET)

The standard approach given bilingual (say English-French) embeddings (Plank, 2017; Da San Martino et al., 2017; Klementiev et al., 2012) has been to use the English word vector corresponding to the English word and the French word vector for the French word. Also, the network is trained only on the English corpora, i.e. trained using English word vectors only. When the input is a French sentence, they use French word vectors. Bilingual embeddings try and project both the English and French word vectors in the same semantic space, but these vectors are not perfectly aligned and might lead to errors in the networks’ prediction.

We propose to obtain the average of the nearest k-english-word-vectors for the given French word and use it as the embedding for the French word. For k=1, this reduces to a bilingual lexical dictionary using bilingual embeddings (Vulic and Moens, 2015; Madhyastha and España-Bonet, 2017). Since the bilingual embeddings are not perfectly aligned, Smith et al. (2017) show that precision@k increases as k increases (e.g. for Hindi P@1 is 0.39, P@3 is 0.58 and P@10 is 0.63), when we obtain French (or any other language) translations for an English word. Thus, we conduct experiments with varying values of k and report the best results for the optimal k. Our experiments confirm the efficacy of KNBET. Further, we believe this KNBET can be used to improve the performance of any multilingual system that uses bilingual embeddings.

4.2.2 Loss Function

We use the distance based logistic triplet loss (Vo and Hays, 2016) which gave better results than a contrastive loss (Bordes et al., 2014). This has also been reported by Vo and Hays (2016) to exhibit better performance in image similarity tasks as well. Here, $S_{pos}$ and $S_{neg}$ are the similarity scores obtained by the question+positive tuple and question+negative tuple respectively.

$$\text{Loss} = \log(1 + e^{(S_{neg} - S_{pos})})$$

5 Experiments

5.1 Dataset

We use the SimpleQuestions (Bordes et al., 2015) dataset which comprises 75.9k/10.8k/21.7k training/validation/test questions. Each question is associated with an answer, i.e. a tuple (subject, predicate, object) from a Freebase (Google, 2017) subset (FB2M or FB5M). The subject is given as a MID 5 and we obtain its corresponding entity name by processing the Freebase data dumps. We were unable to obtain entity name mappings for some MIDs, and these were removed from our final set. We also obtain Hindi translations for all questions in SimpleQuestions using Google Translate. Note, these translations are not perfect and serve as a noisy input to the network. Also, we only translate the questions, and the answers remain in English. As with previous work, we show results over the 2M-subset of Freebase (FB2M).

We use pre-trained word embeddings6 provided by Fasttext (Bojanowski et al., 2016) and use alignment matrices7 provided by Smith et al. (2017) to obtain English-Hindi bilingual embeddings. Smith et al. (2017) use a small set of 5000 words to obtain the alignment matrices. The provided Hindi embeddings are in Devanagari script. We use randomly initialised embeddings between [-0.25, 0.25] for words without embeddings.

We have prepared a dataset of Hindi-English CM questions for a smaller set of 250 tuples obtained from the test split of SimpleQuestions dataset. We gathered these questions from Hindi-English speakers, who were asked to form a natural language CM question, shown a tuple. Further, for every tuple we obtained CM questions from 5 different annotators and pick one at random for the final test set, to ensure multiple variations. Each CM question is in Roman script, and annotators anglicise (or transliterate) Hindi words (Devanagari script) to Roman script, to the best of their ability. This introduces variations in spellings and poses a challenge for the network and also back-transliteration.

---

5 A unique ID referring to an entity in Freebase.
6 https://fasttext.cc/
7 https://goo.gl/Lwgu1D
Table 1: Network Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch Size</td>
<td>100</td>
</tr>
<tr>
<td>Non-linearity</td>
<td>ReLU</td>
</tr>
<tr>
<td>CNN Filters &amp; Width</td>
<td>20 filters each of width 1, 2 and 4 resp.</td>
</tr>
<tr>
<td>Pool Type</td>
<td>Global Max Pooling</td>
</tr>
<tr>
<td>Stride Length</td>
<td>1</td>
</tr>
<tr>
<td>FC Layer 1</td>
<td>100 units + 0.2 Dropout</td>
</tr>
<tr>
<td>FC Layer 2</td>
<td>100 units + 0.2 Dropout</td>
</tr>
<tr>
<td>FC Layer 3</td>
<td>1 unit + No Relu</td>
</tr>
<tr>
<td>Optimizer</td>
<td>Adam (default params)</td>
</tr>
</tbody>
</table>

Table 2: Candidate generation results: Recall of top-k answer candidates for each question type

<table>
<thead>
<tr>
<th>K</th>
<th>English</th>
<th>CM-MT</th>
<th>CM-LT</th>
<th>CM-TL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>68.4</td>
<td>39.1</td>
<td>54.6</td>
<td>58.4</td>
</tr>
<tr>
<td>2</td>
<td>75.7</td>
<td>42.3</td>
<td>59.3</td>
<td>63.8</td>
</tr>
<tr>
<td>5</td>
<td>82.3</td>
<td>49.4</td>
<td>67.2</td>
<td>70.0</td>
</tr>
<tr>
<td>10</td>
<td>85.5</td>
<td>53.4</td>
<td>71.5</td>
<td>73.4</td>
</tr>
<tr>
<td>50</td>
<td>91.4</td>
<td>56.9</td>
<td>78.7</td>
<td>78.6</td>
</tr>
<tr>
<td>100</td>
<td>92.9</td>
<td>59.7</td>
<td>80.6</td>
<td>81.0</td>
</tr>
<tr>
<td>200</td>
<td>94.3</td>
<td>62.1</td>
<td>83.0</td>
<td>83.1</td>
</tr>
</tbody>
</table>

5.2 Generating negative samples

We generate 10 negative samples for each training sample. We follow Bordes et al. (2014) to generate 5 negative samples. These candidates are samples picked at random and then corrupted following Bordes et al. (2014). We further use 5 more negative samples obtained by querying the Solr index. This gives us negative samples which are very similar to the actual answer and further the discriminatory ability of our network. This second policy is unique, and our experiments show that it gives us better performance.

5.3 Evaluation and Baselines

We report results using the standard evaluation criteria (Bordes et al., 2015), in terms of path-level accuracy, which is the percentage of questions for which the top-ranked candidate fact is correct. A prediction is correct if the system correctly retrieved the subject and the relationship.

Since there is no earlier work on CM QA over KBs, we compare the different ways a CM question can be answered using our QA system. We translate the entire CM question to English using Google translate (cm-mt). We also lexicographically translate the CM question to English (cm-lt). Further, since the CM question is in Roman script, we apply language identification (LI), and back-transliteration (BTL) for Hindi words using Bhat et al. (2018) and obtain a CM question which has English words in Roman script and Hindi words in Devanagari (cm-tl). We report results for all these different CM question variations using our TSHCNN. We tried Raghavi et al. (2017) WebShodh9 system, but it did not return any answers, and hence we are unable to use it for comparison.

We also report results for the English questions in SimpleQuestions on our model trained only on English. This serves as a benchmark for our model as compared to other work on SimpleQuestions (Ture and Jojic, 2017; Yu et al., 2017; Yin et al., 2016; Lukovnikov et al., 2017; Golub and He, 2016; Bordes et al., 2015).

Network parameters and decisions are presented in Table 1. We train our model until the validation loss on the validation set stops improving further for 3 epochs. We report the results on the epoch with the best validation loss. We use K = 200 for the initial candidate generation step.

6 Results

6.1 Quantitative Analysis

In Table 2, we present end-to-end results using our CMQA system. It shows competi-

---

8The LI system is trained on CM data and the BTL system is trained with parallel transliteration pairs.

9http://tts.speech.cs.cmu.edu/webshodh/cmqa.php
Table 4: End-to-End Results. We train on different inputs: only English questions, only Hindi questions and both English and Hindi questions. The answers are in English for all training scenarios. TO: Train on, E: English questions, H: Hindi questions, EH: English and Hindi questions, BE: Bilingual Embeddings, CQT: Concatenate question + tuple, KNB: KNBET (K-Nearest Bilingual Embedding Transformation), SCNS: Solr Candidates as Negative Samples, E2E Scores: Candidates obtained using the same CM question variation, cm-lt-tl: Candidates obtained using lexical translation of cm-tl questions and input question was cm-tl.

<table>
<thead>
<tr>
<th>TO</th>
<th>BE</th>
<th>CQT</th>
<th>KNB</th>
<th>SCNS</th>
<th>English Candidates</th>
<th>Codemix Question Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>cm-mt</td>
<td>cm-lt</td>
</tr>
<tr>
<td>E</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>0.39</td>
<td>0.58</td>
</tr>
<tr>
<td>H</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>0.08</td>
<td>0.07</td>
</tr>
<tr>
<td>EH</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>0.40</td>
<td>0.53</td>
</tr>
<tr>
<td>E</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>0.41</td>
<td>0.57</td>
</tr>
<tr>
<td>H</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>0.39</td>
<td>0.54</td>
</tr>
<tr>
<td>EH</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>0.46</td>
<td>0.59</td>
</tr>
<tr>
<td>E</td>
<td>yes</td>
<td>yes</td>
<td>k=3</td>
<td>yes</td>
<td>0.42</td>
<td>0.57</td>
</tr>
<tr>
<td>H</td>
<td>yes</td>
<td>yes</td>
<td>k=3</td>
<td>yes</td>
<td>0.44</td>
<td>0.53</td>
</tr>
<tr>
<td>EH</td>
<td>yes</td>
<td>yes</td>
<td>k=3</td>
<td>yes</td>
<td>0.46</td>
<td>0.59</td>
</tr>
<tr>
<td>E</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>0.42</td>
<td>0.50</td>
</tr>
<tr>
<td>EH</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>0.29</td>
<td>0.44</td>
</tr>
</tbody>
</table>

Monolingual vs Bilingual Embeddings
Results clearly show that improvements are obtained when we use bilingual embeddings. There is an improvement of 17% for cm-tl questions when the network is trained on English and Hindi using bilingual embeddings versus using monolingual embeddings. This is because bilingual embeddings project words with similar semantics more closely. This difference is much more pronounced when we train the network only on Hindi questions. The tuples were still in English, and the misaligned semantic space (when using monolingual embeddings) for English and Hindi made it difficult for the Siamese network to learn anything meaningful. We can also observe an improvement of 11% for cm-lt questions (when trained on English and Hindi questions and using bilingual embeddings). We attribute this to the fact that CM questions have a different word order than English questions. Moreover,
Table 5: Qualitative Analysis. CA: Correct Answer, EC: Candidates obtained for English question used as candidates for CM questions, E2E: Candidates obtained in an end-to-end manner i.e., the same question variation was used to obtain candidates, PA: Predicted Answer, EC, E2E & CM-L T-TL PA: Predicted answers grouped if same.

Examples

**Example 1:** CA (have wheels will travel, book written work subjects, family)

English Question: what is the have wheels will travel book about?

Predicted Answer: (have wheels will travel, book written work subjects, adolescence)

CM Question: have wheel will travel kitaab kis vishaya par likhi gyi hai? PA: NA

CM-MT Question: howe wheel will travel book written on?

EC & E2E PA: (travel, media common literary genre books in this genre, michael palin)

CM-L T Question: have wheel will travel book what subject on wrote added is?

EC & E2E PA: (have wheels will travel, book written work subjects, adolescence)

**Example 2:** CA (traditional music, music genre artists, the henrys)

English Question: which quartet is known for traditional music?

Predicted Answer: (traditional music, music genre albums, music and friends)

CM Question: traditional music ke liye konse artist jaane jate hain? PA: NA

CM-MT Question: which artists are known for traditional music?

EC & E2E PA: (traditional music, music genre artists, sally jaye)

CM-L T Question: traditional music of for which one artist life goes are there?

EC PA: (bbc music volume 19 number 6 chamber music elias string quartet, music album artist, franz schubert)

E2E PA: (the way life goes, music album artist, tom keifer)

**SCNS: Using Solr Candidates as Negative Samples** We ran experiments using 10 negative samples generated as per Bordes et al. (2014). However, the scores obtained when using a combination of both negative sample generation policies: corrupted tuples and Solr candidates, was 12.7% higher. This is a significant improvement in scores.

**CQT: Additional Input, Concatenate question + tuple** 10 We obtain an improve-

with the use of bilingual embeddings, our network can project both Hindi and English questions into the same semantic space, which in turn helps CM questions. The effect of monolingual embeddings is visible when we train only on Hindi. We notice accuracies for all CM question variations drop significantly.

**K-Nearest Bilingual Embedding Transformation (KNBET)** With k = 3, the results obtained with KNBET are higher by 16% for cm-tl trained only on English compared to no KNBET. This demonstrates that our transformation increases the effectiveness of bilingual embeddings. This is attributed to the fact that our transformation reduces the errors that bilingual embeddings may otherwise possess due to imperfect alignment.

**Training on Hindi Questions** Training with Hindi questions helps the network learn the different word orders that are present in Hindi questions. This improves scores for cm-tl questions when trained only on Hindi. Further, joint training on both English and Hindi questions gives us the best results.
ment of 34% - 62% in our scores when we provide additional input in the form of concatenated question and tuple. One plausible explanation for this improvement is the 50% more features for the network. To verify this, we added more filters to our convolution layer such that total features equaled that when additional input was provided. However, the improvement in results was only marginal. Another, more likely explanation would be that the max pooling layer picks out the dominant features from this additional input, and these features increase the discriminatory ability of our network.

**EC: English candidates** We perform experiments wherein we use the same set of candidates obtained for English questions as the candidates for all CM question variations (cmmt, cm-lt and cm-tl). Results show that cm-tl questions give the highest scores on a network trained on both English and Hindi questions using bilingual embeddings. This result shows that lexical translation might not be the best strategy to tackle CM questions. Further, more techniques should be devised to handle the CM question in its original form rather than translating it at the sentence or lexical level.

### 6.2 Qualitative Analysis

In Table 5, some examples are shown to depict how results of transliterated CM question fare better than their translated counterparts. Example 1 shows that machine translation fails to translate the CM question correctly. The predicted answer is henceforth incorrect. Example 2 highlights limitations for lexical translation. Lexically translated questions lose their intended meaning if a word has multiple possible translations and it results in an incorrect prediction.

### 7 Conclusion

This paper proposes techniques for Code-Mixed Question Answering over a Knowledge Base in the absence of direct supervision of CM questions for training neural models. We use only monolingual data \(^{11}\) and bilingual embeddings to achieve promising results. Our TSHCNN model shows impressive results for English QA. It outperforms many other complicated architectures that use Bi-LSTMs and Attention mechanisms. We also introduce two techniques which significantly enhance results. KNBET reduces the errors that may exist in bilingual embeddings and could be used by any system working with bilingual embeddings. Additionally, negative samples obtained through Solr are useful for the network to learn to differentiate between fine-grained inputs. Despite imperfect bilingual embeddings, our model shows impressive results for CM QA. Our experiments highlight the need for CM QA system, since CM questions in their original form outperforms translated CM questions.

### References

- Somnath Banerjee, Sudip Kumar Naskar, Paolo Rosso, and Sivaji Bandyopadhyay. 2016. The first cross-script code-mixed question answering corpus. In *MultiLingMine@ECIR*.

\(^{11}\)The language identification system uses CM data. We could instead use a rule-based system using no CM data without much loss in performance.


Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale Simple Question Answering with Memory Networks.


