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-mix (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

Proceedings of The Third Workshop on Computational Approaches to Code-Switching, pages 39–50
Melbourne, Australia, July 19, 2018. ©2018 Association for Computational Linguistics

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 [1] 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 predominately 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

[1] 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 $\mathrm{s_i}$ represents a subject entity, $\mathrm{p_i}$ a predicate (also referred as relation), and $\mathrm{o_i}$ an object entity. The task of SimpleQuestions is then: Given a question represented as a sentence, i.e. a sequence of words $q = \{w_1, ..., w_n\}$, 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.041ftf** 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/

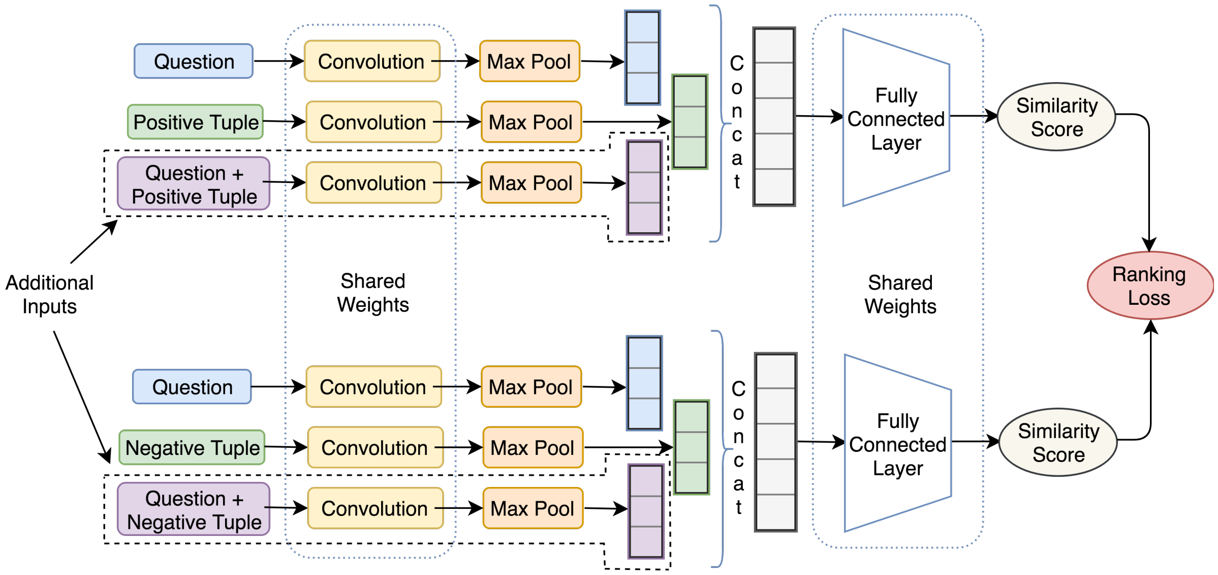

Figure 1: TSHCNN Architecture

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[3] 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.

[3] English 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 say, 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[4] 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, $\mathrm{S_{pos}}$ and $\mathrm{S_{neg}}$ are the similarity scores obtained by the question+positive tuple and question+negative tuple respectively.

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

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 embeddings[6] provided by Fasttext (Bojanowski et al., 2016) and use alignment matrices[7] 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 posses a challenge for the network and also back-transliteration.

[4] Results available on the GitHub repo: github.com/Babylonpartners/fastText_multilingual

[5] A unique ID referring to an entity in Freebase.

[6] https://fasttext.cc/

[7] https://goo.gl/Lwgu1D

Table 1: Network Parameters

Parameter	*Value*
Batch Size	100
Non-linearity	Relu
CNN Filters & Width	20 filters each of width 1, 2 and 4 resp.
Pool Type	Global Max Pooling
Stride Length	1
FC Layer 1	100 units + 0.2 Dropout
FC Layer 2	100 units + 0.2 Dropout
FC Layer 3	1 unit + No Relu
Optimizer	Adam (default params)

Table 2: End-to-End Answer Accuracy for English Questions

Model	*Acc.*
Bordes et al. (2015)	62.7
Golub and He (2016)	70.9
Lukovnikov et al. (2017)	71.2
Yin et al. (2016)	76.4
Yu et al. (2017)	77.0
Ture and Jojic (2017)	86.8
Ours: Candidate Generation	68.5
Ours: Candidate Re-Ranking	77.0

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 lexically 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)[8] 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) WebShodh[9] system, but it did not return any answers, and hence we are unable to use it for comparison.

Table 3: Candidate generation results: Recall of top-k answer candidates for each question type

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

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-

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

[9] http://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.

TO	BE	CQT	KNB	SCNS	Codemix Question Accuracy						
					English Candidates			E2E Scores			
					cm-mt	cm-lt	cm-tl	cm-mt	cm-lt	cm-tl	cm-lt-tl
E	no	yes	no	yes	0.39	0.58	0.37	0.31	0.51	0.32	0.33
H	no	yes	no	yes	0.08	0.07	0.17	0.07	0.06	0.16	0.15
EH	no	yes	no	yes	0.40	0.53	0.53	0.31	0.50	0.48	0.47
E	yes	yes	no	yes	0.41	0.57	0.46	0.34	0.53	0.41	0.41
H	yes	yes	no	yes	0.39	0.54	0.57	0.30	0.50	0.52	0.54
EH	yes	yes	no	yes	**0.46**	**0.59**	**0.62**	0.34	**0.55**	0.55	**0.57**
E	yes	yes	k=3	yes	0.42	0.57	0.53	0.34	0.54	0.50	0.49
H	yes	yes	k=3	yes	0.44	0.53	0.59	0.33	0.46	0.54	0.55
EH	yes	yes	k=3	yes	**0.46**	**0.59**	0.61	**0.35**	**0.55**	**0.56**	0.54
E	yes	yes	no	no	0.42	0.50	0.55	0.33	0.50	0.53	0.52
EH	yes	no	no	yes	0.29	0.44	0.40	0.21	0.37	0.37	0.38

tive results on English questions with all but one of the more recent approaches for SimpleQuestions. This shows the effectiveness of our model for English QA. Our initial candidate generation step surprisingly surpasses the original Bordes et al. (2015) paper.

In Table 3, we report candidate generation results. We obtain candidates for each CM question variation using the question itself as a query. Further, cm-tl has words in Devanagari script which do not contribute to the search similarity scores when searching over an English corpus. Thus we use the candidates obtained for the lexical translation of cm-tl questions as candidates for cm-tl. This variation with candidates of cm-lt and questions of cm-tl is termed as cm-lt-tl. Additionally, we show results using the candidates obtained for the English question as the candidates for all three CM question variations (cm-mt, cm-lt and cm-tl). This ensures a fair comparison of all three CM question variations using TSHCNN.

In Table 4, we show results on the CM questions. Our model TSHCNN, trained on both English and Hindi questions gives the best scores. It is better by 3 - 8% for various CM question variations. Although, training only on English and using bilingual embeddings should offer performance that matches training on both English and Hindi. However, this does not happen since the bilingual embeddings are not perfect (see subsection KNBET). We do an ablation study of the various components and describe them in more detail further.

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-LT-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-LT Question: have wheel will travel book what subject on wrote added is? EC & E2E PA: (have wheels will travel, book written work subjects, adolescence)
CM-TL Question: have wheel will travel किताब किस विषय पर लिखी गयी है? EC, E2E & CM-LT-TL 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-LT 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)
CM-TL Question: traditional music के लिए कोनसे artist जान जाते हैं? EC, E2E & CM-LT-TL PA: (traditional music, music genre artists, sally jaye)

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.

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-

[10] We made sure that the experiments with no CQT

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 equalled 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 (cm-mt, 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.

had the same number of features as that of with CQT.

[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.

References

Arvind Agarwal, Hema Raghavan, Karthik Subbian, Prem Melville, Richard D. Lawrence, David Gondek, and James Z Fan. 2012. Learning to rank for robust question answering. In *CIKM*.

Kisuh Ahn, Beatrice Alex, Johan Bos, Tiphaine Dalmas, Jochen L. Leidner, and Matthew Smillie. 2004. Cross-lingual question answering using off-the-shelf machine translation. In *CLEF*.

Waleed Ammar, George Mulcaire, Yulia Tsvetkov, Guillaume Lample, Chris Dyer, and Noah A. Smith. 2016. Massively multilingual word embeddings. *CoRR*, abs/1602.01925.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. Learning bilingual word embeddings with (almost) no bilingual data. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462.

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

Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Manish Shrivastava, and Dipti Misra Sharma. 2018. Universal dependency parsing for hindi-english code-switching.

Riyaz Ahmad Bhat, Manish Shrivastava, Irshad Ahmad Bhat, and Dipti Misra Sharma. 2017. Joining hands: Exploiting monolingual treebanks for parsing of code-mixing data. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain,* *April 3-7, 2017, Volume 2: Short Papers*, pages 324–330.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

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

Antoine Bordes, Jason Weston, and Nicolas Usunier. 2014. Open Question Answering with Weakly Supervised Embedding Models. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 8724 LNAI, pages 165–180.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1993. Signature verification using a "siamese" time delay neural network. In *Proceedings of the 6th International Conference on Neural Information Processing Systems*, NIPS'93, pages 737–744, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Özlem Çetinoğlu, Sarah Schulz, and Ngoc Thang Vu. 2016. Challenges of computational processing of code-switching. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 1–11. Association for Computational Linguistics.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017a. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879. Association for Computational Linguistics.

Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. 2017b. Multilingual Knowledge Graph Embeddings for Cross-lingual Knowledge Alignment. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 1511–1517, California. International Joint Conferences on Artificial Intelligence Organization.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. In *SSST@EMNLP*.

Sumit Chopra, Raia Hadsell, and Yann LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, pages 539–546, Washington, DC, USA. IEEE Computer Society.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word translation without parallel data. *CoRR*, abs/1710.04087.

Giovanni Da San Martino, Salvatore Romeo, Alberto Barroón-Cedeño, Shafiq Joty, Lluís Maàrquez, Alessandro Moschitti, and Preslav Nakov. 2017. Cross-language question re-ranking. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, pages 1145–1148, New York, NY, USA. ACM.

Arpita Das, Harish Yenala, Manoj Kumar Chinnakotla, and Manish Shrivastava. 2016. Together we stand: Siamese networks for similar question retrieval. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Anik Dey and Pascale Fung. 2014. A hindi-english code-switching corpus. In *LREC*.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2014. Open question answering over curated and extracted knowledge bases. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 1156–1165, New York, NY, USA. ACM.

David Golub and Xiaodong He. 2016. Character-Level Question Answering with Attention.

Google. 2017. Freebase data dumps. `https://developers.google.com/freebase/data`.

Jeenu Grover, Prabhat Agarwal, Ashish Sharma, Mayank Sikka, Koustav Rudra, and Monojit Choudhury. 2017. I may talk in english but gaali toh hindi mein hi denge: A study of english-hindi code-switching and swearing pattern on social networks. In *Proceedings of the Social Networking Workshop, COMSNETS 2017*. IEEE.

Sherzod Hakimov, Soufian Jebbara, and Philipp Cimiano. 2017. AMUSE: Multilingual semantic parsing for question answering over linked data. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10587 LNCS:329–346.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 1693–1701, Cambridge, MA, USA. MIT Press.

Mark Hopkins, Cristian Petrescu-Prahova, Roie Levin, Ronan Le Bras, Alvaro Herrasti, and Vidur Joshi. 2017. Beyond sentential semantic parsing: Tackling the math sat with a cascade of tree transducers. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 795–804. Association for Computational Linguistics.

Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2015. Convolutional Neural Network Architectures for Matching Natural Language Sentences. *NIPS*, page 2009.

Rie Johnson and Tong Zhang. 2015. Effective use of word order for text categorization with convolutional neural networks. In *HLT-NAACL*.

Aditya Joshi, Ameya Prabhu, Manish Shrivastava, and Vasudeva Varma. 2016. Towards subword level compositions for sentiment analysis of hindi-english code mixed text. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 2482–2491.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. pages 1746–1751.

Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. 2012. Inducing Crosslingual Distributed Representations of Words. *24th International Conference on Computational Linguistics - Proceedings of COLING 2012: Technical Papers (2012)*, (December):1459–1474.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *AAAI*.

Chuan-Jie Lin and Yu-Min Kuo. 2010. Description of the ntou complex qa system. In *NTCIR*.

Denis Lukovnikov, Asja Fischer, Jens Lehmann, and Sören Auer. 2017. Neural Network-based Question Answering over Knowledge Graphs on Word and Character Level. *Proceedings of the 26th International Conference on World Wide Web - WWW '17*, pages 1211–1220.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Bilingual word representations with monolingual quality in mind. In *VS@HLT-NAACL*.

Pranava Swaroop Madhyastha and Cristina España-Bonet. 2017. Learning bilingual projections of embeddings for vocabulary expansion in machine translation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 139–145. Association for Computational Linguistics.

Carol Myers-Scotton. 2002. *Contact linguistics: Bilingual encounters and grammatical outcomes.* Oxford University Press on Demand.

Sarath Chandar A P, Stanislas Lauly, Hugo Larochelle, Mitesh M. Khapra, Balaraman Ravindran, Vikas Raykar, and Amrita Saha. 2014. An Autoencoder Approach to Learning Bilingual Word Representations.

Barbara Plank. 2017. All-in-1 at ijcnlp-2017 task 4: Short text classification with one model for all languages.

Amir Pouran Ben Veyseh. 2016. Cross-Lingual Question Answering Using Common Semantic Space. *Workshop on Graph-based Methods for Natural Language Processing, NAACL-HLT 2016 ,*, pages 15–19.

Khyathi Chandu Raghavi, Manoj Kumar Chinnakotla, Alan W. Black, and Manish Shrivastava. 2017. Webshodh: A code mixed factoid question answering system for web. In *CLEF*.

Khyathi Chandu Raghavi, Manoj Kumar Chinnakotla, and Manish Shrivastava. 2015. "Answer ka type kya he?". In *Proceedings of the 24th International Conference on World Wide Web - WWW '15 Companion*, pages 853–858, New York, New York, USA. ACM Press.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392. Association for Computational Linguistics.

Han Ren, Dong-Hong Ji, and Jing Wan. 2010. Whu question answering system at ntcir-8 aclia task. In *NTCIR*.

Shruti Rijhwani, Royal Sequiera, Monojit Choudhury, Kalika Bali, and Chandra Sekhar Maddila. 2017. Estimating code-switching on twitter with a novel generalized word-level language detection technique. In *Proc. of ACL 2017*. ACL.

Samuel L. Smith, David H. P. Turban, Steven Hamblin, and Nils Y. Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. pages 1–10.

Ferhan Ture and Elizabeth Boschee. 2016. Learning to translate for multilingual question answering. In *EMNLP*.

Ferhan Ture and Oliver Jojic. 2017. No Need to Pay Attention: Simple Recurrent Neural Networks Work! (for Answering "Simple" Questions). *Empirical Methods in Natural Language Processing (EMNLP)*, pages 2866–2872.

Nam N. Vo and James Hays. 2016. Localizing and orienting street views using overhead imagery. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9905 LNCS:494–509.

Ivan Vulic and Marie-Francine Moens. 2015. Bilingual word embeddings from non-parallel document-aligned data applied to bilingual lexicon induction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015)*, pages 719–725. ACL.

Yogarshi Vyas, Spandana Gella, Jatin Sharma, Kalika Bali, and Monojit Choudhury. 2014. Pos tagging of english-hindi code-mixed social media content. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 974–979. Association for Computational Linguistics.

Jieyu Wang and Anita Komlodi. 2016. Understanding users' language selection: Code-switching in online searches. In *Proceedings of the 2016 ACM on Conference on Human Information Interaction and Retrieval*, CHIIR '16, pages 377–379, New York, NY, USA. ACM.

Wenpeng Yin, Mo Yu, Bing Xiang, Bowen Zhou, and Hinrich Schütze. 2016. Simple question answering by attentive convolutional neural network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1746–1756. The COLING 2016 Organizing Committee.

Mo Yu, Wenpeng Yin, Kazi Saidul Hasan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2017. Improved Neural Relation Detection for Knowledge Base Question Answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 571–581, Stroudsburg, PA, USA. Association for Computational Linguistics.

Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. 2017. Adversarial training for unsupervised bilingual lexicon induction. In *ACL*.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *NIPS*.