Non-sentential Question Resolution using Sequence to Sequence Learning

Vineet Kumar
IBM Research Labs
New Delhi, India
vineeku6@in.ibm.com

Sachindra Joshi
IBM Research Labs
New Delhi, India
jasachind@in.ibm.com

Abstract

An interactive Question Answering (QA) system frequently encounters non-sentential (incomplete) questions. These non-sentential questions may not make sense to the system when a user asks them without the context of conversation. The system thus needs to take into account the conversation context to process the incomplete question. In this work, we present a recurrent neural network (RNN) based encoder decoder network that can generate a complete (intended) question, given an incomplete question and conversation context. RNN encoder decoder networks have been show to work well when trained on a parallel corpus with millions of sentences, however it is extremely hard to obtain conversation data of this magnitude. We therefore propose to decompose the original problem into two separate simplified problems where each problem focuses on an abstraction. Specifically, we train a semantic sequence model to learn semantic patterns, and a syntactic sequence model to learn linguistic patterns. We further combine syntactic and semantic sequence models to generate an ensemble model. Our model achieves a BLEU score of 30.15 as compared to 18.54 using a standard RNN encoder decoder model.

1 Introduction

Question Answering (QA) systems (Green Jr et al., 1961; Winograd, 1971; Woods and Kaplan, 1977; Hickl et al., 2006; Gobeill et al., 2009) enable a user to obtain precise information. A natural extension is an interactive and dialogue based QA system that allows a user to ask follow up or related questions. Interactive QA system however comes with its unique set of challenges. Users ask a follow up or related question by being as terse as possible, and they implicitly refer to concepts and entities in the past conversation. Table 1 depicts a few instances of follow up questions users may ask in an ongoing conversation.

Incomplete questions are a subset of non-sentential utterances (NSU) (Fernández, 2006). NSUs are incomplete utterances which make complete sense when seen in conjunction with the utterances in conversation. Table 1 illustrates some examples of NSU questions ($Q2$) a user might ask the system given a previous question ($Q1$) and an answer ($A1$). $R1$ refers to the intended complete question. Note that (a) and (c) need the previous question $Q1$, (b) needs previous answer $A1$, whereas (d) needs both $Q1$ and $A1$ to generate $R1$. The system thus either needs to restrict how users interact (Carbonell, 1983), or needs to handle the NSU questions by considering the conversation context. Restricting how users interact with a QA system is not natural, and thus can make the system hard to use. In this work, we focus on using the incomplete question and the conversation context to generate the resolved (intended) question. In the rest of the paper, we refer to this problem as NSU question resolution.

NSU resolution is an active area of research. One set of work deals with classifying NSU (Fernández et al., 2005). Another set of work proposes a rule or grammar based approach to resolve NSU (Carbonell, 1983; Dalrymple et al., 1991). Recently, a statistical based approach has been proposed for resolving NSU question (Raghu et al., 2015). However, this approach only focuses on the simpler problem of

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 2022–2031, Osaka, Japan, December 11-17 2016.

(a)

Q1	how old was john rolfe when he died ?
A1	37
Q2	and how did he die ?
R1	how did john rolfe die ?

(b)

Q1	what animal has a 7 lettered name ?
A1	cheetah
Q2	and how fast can it run ?
R1	how fast can a cheetah run ?

(c)

Q1	what is greece 's national sport ?
A1	football
Q2	flower ?
R1	what is greece 's national flower ?

(d)

Q1	what do road runners eat ?
A1	small reptiles
Q2	how often ?
R1	how often do road runners eat small reptiles ?

Table 1: Examples of non-sentential questions in conversations:
(a) and (c) need the previous question $Q1$
(b) needs previous answer $A1$; (d) needs both $Q1$ and $A1$ to be resolved

resolving NSU based on previous questions, and thus will not be able to handle examples given in Table 1(b) and 1(d), where previous answer or a combination of previous question and answer is needed.

Recently, recurrent neural network (RNN) based encoder decoder networks have been applied successfully to the task of statistical machine translation (Cho et al., 2014; Bahdanau et al., 2014; Sutskever et al., 2014). RNN encoder decoder, also known as sequence to sequence learning, maps a variable length input sequence to a variable length output sequence. In this work, we approach the problem of NSU question resolution as sequence to sequence learning. We generate the input sequence by concatenating NSU question, previous question and answer. RNN encoder decoder is then used to learn a mapping of this input sequence to the resolved question.

RNN encoder decoder models have been successfully trained on huge parallel corpus of millions of sentences (Bahdanau et al., 2014; Cho et al., 2014; Sutskever et al., 2014). However, it is extremely hard to obtain conversation data of this magnitude. We have access to only 7220 conversations containing NSU questions, which were collected using Amazon Mechanical Turk (Raghu et al., 2015).

As we have a small dataset, we propose to decompose the original problem into two separate simplified problems where each problem uses an abstraction. These abstractions help the model training to focus on learning a specific aspect of the problem. Specifically, we train a syntactic sequence model to learn linguistic patterns, and a semantic sequence model to learn semantic patterns. We combine these two different models to generate an ensemble model, which can capture both linguistic and semantic patterns in NSU question conversations.

Our main contributions in this work are as follows:

1. We present a novel approach to handle non-sentential questions using the framework of sequence to sequence learning. Our approach is completely data driven, and can generate complete questions from a non-sentential question, given previous question and answer.

2. We propose a method to decompose the original NSU question resolution problem into two separate simplified abstractions that focus on learning a specific aspect of the problem. One such abstraction is semantic patterns in conversation data, that we learn with the help of a semantic sequence model.

3. We present a syntactic sequence model that focuses solely on learning linguistic patterns in conversations. Finally, we combine the semantic and syntactic sequence models to generate an ensemble model. Our ensemble model achieves a BLEU score of 30.15 as compared to 18.54 using a standard RNN encoder decoder.

Rest of this paper is organized as follows. We discuss related work in Section 2. Background needed to understand RNN encoder decoder model is discussed in Section 3. We present syntactic and semantic sequence models in Section 4 and Section 5 respectively. Finally, we discuss experiment settings and results in Section 6 and conclude in Section 7.

2 Related Work

NSUs were studied and classified into various classes by Ferna'ndez and Ginzburg (2002). One thread of work has focused on identifying and classifying NSUs into classes (Fernández et al., 2005; Rovira, 2006). Another thread of work has focused on resolving NSUs into complete intended utterances by building domain specific rules or grammar (Dalrymple et al., 1991; Carbonell, 1983). Writing rules or grammar is hard, extremely time consuming and may suffer with low recall. Therefore, we focus on a data driven and statistical approach.

Raghu et al. (2015) is the only work we know of that uses a statistical and data-driven model to resolve NSU questions. However their model cannot handle cases where previous answer or a combination of previous question and answer is needed to resolve a NSU question. For example, their approach cannot handle examples given in Table 1(b) and 1(d). Our approach does not have any such restrictions.

Sequence to sequence learning (Sutskever et al., 2014; Bahdanau et al., 2014; Cho et al., 2014) has been applied to a myriad applications. Some of the successful applications include statistical machine translation, speech translation (Duong et al., 2016), translating videos to sentences (Venugopalan et al., 2015), image captioning (Karpathy and Fei-Fei, 2015; Jia et al., 2015). Sequence to sequence learning has also been applied in modeling conversations (Li et al., 2016; Serban et al., 2016).

To the best of our knowledge, ours is the first work that approaches NSU question resolution as a sequence to sequence learning problem. Ours is also the first work that decomposes the original sequence to sequence learning problem, into separate simplified problems where each problem focuses on an abstraction.

3 Sequence to Sequence Learning

In this section, we discuss the framework of RNN encoder decoder model. This is followed by discussion on why it can be hard to train a RNN encoder decoder model using a small dataset. We finally formulate NSU question resolution as a sequence to sequence learning problem.

3.1 Background and Model Size

Sequence to sequence learning framework uses a recurrent neural network (RNN) to encode a variable-length input sequence to a fixed length vector, and then uses another RNN to decode the vector into a variable-length target sequence (Cho et al., 2014).

The model takes a source sentence (x) as input. Each sentence is a sequence of words, and each word is encoded using a one-hot encoding:

$$x = (x_1, x_2, \cdots, x_{t_x}), x_i \in \Re^{|V|}$$

The model outputs a target sentence (y), which is a sequence of words:

$$y = (y_1, y_2, \cdots, y_{t_y}), y_i \in \Re^{|V|}$$

where t_x and t_y respectively denote length of sequence x and y, and $|V|$ denotes the vocabulary size, and t_x need not be same as t_y. Note that compared to a neural machine translation model, we do not need a separate vocabulary for source and target, as source and target are in the same language (English).

RNN encoder first computes its forward state which are fixed length vectors $\vec{h}_i$:

$$\vec{h}_i = \begin{cases} (1 - \vec{z}_i) \circ \vec{h}_{i-1} + \vec{z}_i \circ \underline{\vec{h}_i} & \text{if i} > 0 \\ 0 & \text{if } i = 0 \end{cases}$$

where

$$\underline{\vec{h}_i} = tanh(\vec{W}\overline{E}x_i + \vec{U}[\vec{r}_i \circ \vec{h}_{i-1}])$$
$$\vec{z}_i = \sigma(\vec{W}_z\overline{E}x_+\vec{U}_z\vec{h}_{i-1})$$
$$\vec{r}_i = \sigma(\vec{W}_r\overline{E}x_+\vec{U}_r\vec{h}_{i-1})$$

$\overline{E} \in \Re^{m \times |V|}$ is the word embedding matrix. $\vec{W}, \vec{W}_z, \vec{W}_r \in \Re^{n \times m}$, $\vec{U}, \vec{U}_z, \vec{U}_r \in \Re^{n \times n}$ are weight matrices. m and n are word embedding dimensionality and number of hidden units respectively. σ is the logistic sigmoid function, $\circ$ is element wise multiplication.

RNN decoder is then initialized by a context vector $\vec{c}$. Typically a context vector is some combination of RNN Encoder's forward state vectors $\vec{h}_i$. Cho et al. (2014) and Sutskever et al. (2014) assign the context vector as $\vec{h}_{t_x}$, whereas Bahdanau et al. (2014) assign the context vector as a combination of RNN encoder hidden states ($\vec{h}_1, \vec{h}_2 \cdots \vec{h}_{t_x}$). The context vector c is then used to output sequence words.

Table 2 shows model size used by various RNN encoder decoder implementations. As $n \ll |V|$ and $m \ll |V|$, we can see that $\overline{E}$ dominates over other parameters $\vec{W}, \vec{W}_z, \vec{W}_r, \vec{U}, \vec{U}_z, \vec{U}_r$. This is usually not a problem when training data is large (of order of million sentences). However, for a small dataset, training a model with so many parameters does not work. We observed the same in our experiments.

We can reduce the vocabulary size, by replacing words that occur below a minimum frequency threshold with a special unknown symbol (UNK). This however, discards lots of useful information.

We present two new models: syntactic sequence (Section 4) and semantic sequence (Section 5) which can preserve and learn linguistic and semantic patterns respectively, while keeping the vocabulary size small.

Model	**Training data**	V	n	m
(Cho et al., 2014)	12M	15,000	1000	620
(Bahdanau et al., 2014)	12M	30,000	1000	620
(Sutskever et al., 2014)	12M	160,000	1000	620

Table 2: Model size for RNN Decoder. n is hidden layer size, m is word embedding size

3.2 Modeling NSU question resolution as sequence to sequence learning

We cast the problem of NSU question resolution as sequence to sequence learning. We concatenate the non-sentential question ($Q2$) and context ($Q1$, $A1$) to generate the source sequence. We use a special end of utterance symbol (END) to create the input sequence. This source sequence is then used to generate the resolved question ($R1$). For example, Table 3 depicts parallel corpus transformation for Table 1(c).

Source	what is greece 's national sport ? END football END flower ?
Target	what is greece 's national flower ?

Table 3: Parallel corpus formulation of Table 1(c)

Figure 1 depicts how RNN encoder decoder works. RNN encoder first processes the entire input sequence ($Q1$, $A1$, $Q2$) to a single fixed dimension vector (context vector $\vec{c}$). This vector is then used to initialize the RNN decoder. RNN decoder then samples output sequence by conditioning on previous sampled word, and the context vector.

4 Syntactic Sequence Model

We discussed in Section 3.1 that the parameters of RNN encoder decoder model are dominated by the size of vocabulary $|V|$. Even for a small dataset (1000s of sentences), $|V|$ may be of the order of 10,000. Thus for a small dataset, a RNN encoder decoder model has too many parameters to train it well.

We can reduce the vocabulary size by replacing out of vocabulary (OOV) with a special unknown symbol (UNK). However, we lose information by restricting vocabulary in this manner. In some cases, we just end up training the model to reproduce previous question $Q1$. For example, Table 1(c) is transformed such that $R1$ is exactly identical to $Q1$. Important information is lost that last OOV word (sport) in $Q1$ should be replaced by the OOV (flower) in $Q2$ to generate the complete question $R1$.

```
Q1: what is UNK 's national UNK ?
A1: UNK
Q2: UNK ?
R1: what is UNK 's national UNK ?
```

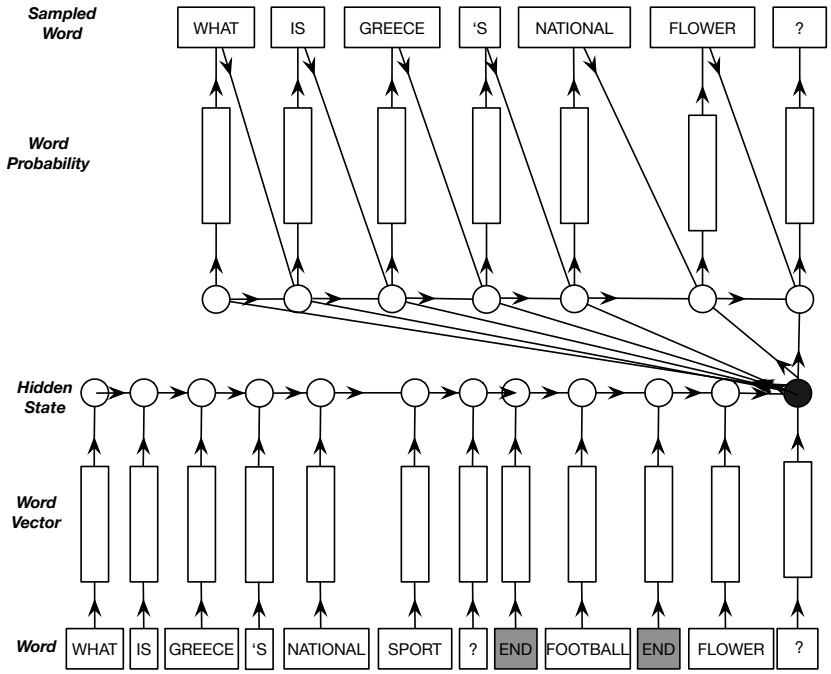

Figure 1: RNN based Encoder decoder for NSU question resolution

We can preserve linguistic structure by assigning a unique symbol to each OOV word. However, this does not help in reducing the vocabulary size. We can thus restrict assigning a new symbol only within a conversation($Q1, Q2, R1$) and reuse the symbols across conversations. Hence, it makes sense to assign symbols based on number of unknowns and its position in a single conversation. Table 4 (a) depicts how new symbols are assigned for the conversation in Table 1(c). Table 4(b) similarly shows how new symbols are assigned for the conversation in Table 1(b). Note how symbols(UNK1, UNK2, UNK3, UNK4) are shared across these two conversations. This allows the model to preserve (and learn) linguistic structure across different conversations.

(a)

Q1	what is UNK1 's national UNK2 ?	greece	UNK1
A1	UNK3	sport	UNK2
Q2	UNK4 ?	football	UNK3
R1	what is UNK1 's national UNK4 ?	flower	UNK4

(b)

Q1	what UNK1 has a UNK2 UNK3 name ?	animal	UNK1
A1	UNK4	7	UNK2
Q2	and how fast can it run ?	lettered	UNK3
R1	how fast can a UNK4 run ?	cheetah	UNK4

Table 4: Syntactic sequence training data for Table 1(c) and Table 1(b). Note how new symbols are assigned for each conversation, but shared across conversations.

Syntactic sequence model uses NSU question ($Q2$), conversation context ($Q1, A1$) and a symbol map, to generate the resolved question ($R1$). This symbol map helps in two important ways: it helps preserve the linguistic structure, and at the time of prediction it helps replace unknown symbol with the original word. We can also compare syntactic sequence model to a standard RNN encoder decoder model, where vocabulary is restricted and all OOV words are replaced with a single UNK. A standard RNN encoder decoder model will end up having UNK symbols as output. However, it is not possible to determine which word does this symbol correspond to, as there will be typically many UNK words in an input sequence. Syntactic sequence model addresses this problem by having a symbol map for the current conversation.

Syntactic sequence model however focuses solely on the position of OOV word in the sequence to assign a new unknown symbol and completely discards similarity between OOV words. In the next section (Section 5), we introduce a semantic sequence model that directly addresses this issue. Semantic sequence model focuses on learning semantic patterns from conversations.

5 Semantic Sequence Model

Syntactic sequence model focuses on learning linguistic patterns in conversations. However, it completely ignores the similarity of OOV words. This can lead to two different input sequences($Q1, A1, Q2$) to appear completely identical, even when they resolve to a different question $R1$. For example, Table 5 depicts two different input sequences with different $R1$, that look completely identical after assigning new unknown symbols based solely on position. Syntactic sequence model discards the information that OOV words 'Greece' and 'India' are similar, and 'flower' and 'sport' are similar (as compared to other tokens).

(a)

Q1	What is Greece 's national sport ?
A1	football
Q2	flower ?
R1	What is Greece 's national flower ?

Q1	What is UNK1 's national UNK2 ?
A1	UNK3
Q2	UNK4 ?
R1	What is UNK1 's national UNK4 ?

Greece	UNK1
sport	UNK2
football	UNK3
flower	UNK4

(b)

Q1	What is Greece 's national sport ?
A1	football
Q2	India ?
R1	What is India 's national sport ?

Q1	What is UNK1 's national UNK2 ?
A1	UNK3
Q2	UNK4 ?
R1	What is UNK4 's national UNK2 ?

Greece	UNK1
sport	UNK2
football	UNK3
India	UNK4

Table 5: Syntactic sequence input and output for two conversations with same $Q1$ but different $Q2$. Note how this model ends up assigning the same input sequence to (a) and (b)

We thus need a model that can exploit the similarity between OOV words and learn a higher level of abstraction. We can assign each OOV word a category number, based on a pre-learnt word category assignment. Each OOV word can then be assigned a new symbol based upon its word category index.

Semantic sequence model assigns a new symbol to each OOV word, based on the word category index. We learn the word category assignments by using a k-means algorithm (MacQueen and others, 1967), where pre-trained word vectors (Mikolov et al., 2013) are used as features.

Assigning new unknown symbols based on word similarity helps the model to focus on a powerful abstraction. The model learns that if a word of a particular category appears in a conversation, output will have words of a specific category. For example, Table 6 demonstrates how the model is trained to retain same output structure even with different NSU question ($Q2$). This is helpful as model can correctly be trained to preserve output structure at the level of word category, even with variations in input sequence.

Semantic sequence model takes as input a NSU question $Q2$, conversation context ($Q1, A1$) and a cluster symbol map. As compared to syntactic sequence model, we can have multiple OOV words assigned to the same cluster symbol token. We can replace the cluster symbol token (such as CL3) by replacing it with $Q2$ OOV word that was assigned to this cluster. We replace it with $Q2$ OOV word, as there is a greater chance that words in $Q2$ will appear in resolved utterance. For example, in Table 6(a), we can replace CL3 with flower.

(a)

Q1	What is CL1 's national CL3 ?
A1	CL3
Q2	CL3 ?
R1	What is CL1 's national CL3 ?

Greece	CL1
sport, football, flower	CL3

(b)

Q1	What is CL1 's national CL3 ?
A1	CL3
Q2	CL1 ?
R1	What is CL1 's national CL3 ?

Greece, India	CL1
sport, football	CL3

Table 6: Semantic sequence input and output for Table 5

6 Experiments and Results

6.1 Dataset

We evaluate our models on NSU question conversation data which was collected using Amazon Mechanical Turk (Raghu et al., 2015). NSU question conversation data has 7220 conversations. Each conversation consists of a previous question ($Q1$), previous answer ($A1$), NSU question ($Q2$), and a

resolved question ($R1$). Table 1 highlights a few examples from the dataset. This dataset however has many spelling mistakes, that were fixed manually using a spell checker. 6820 conversations were used for training and the remaining 400 were used as a validation set. We further lower case and then tokenize the text. RNN encoder decoder needs a parallel corpus of an input and output sequence for training. Input sequence is generated by concatenating question ($Q1$), answer ($A1$) and NSU question ($Q2$) with a special end of utterance symbol (END). We use resolved question ($R1$) as the output sequence. Table 3 lists a sample input and output sequence. There are a total of 12,603 word types, 134K words in input sequence text and 65K words in output sequence text.

6.2 Training and Model details

For all experiments, Bidirectional RNN encoder decoder with attention mechanism (Bahdanau et al., 2014) is used. Gated Recurrent unit (GRU) (Cho et al., 2014) is used as the hidden unit for RNN. We used Adam (Kingma and Ba, 2014) as the optimization algorithm with a learning rate of 0.005 and mini-batch size of 128. Although GRU does not suffer from the vanishing gradient problem, it can still suffer from exploding gradient (Graves, 2013; Pascanu et al., 2013). Thus, a hard constraint on norm of the gradient was enforced by scaling it when norm exceeds a threshold.

Word embedding matrix was initialized using pre-trained word vectors (Mikolov et al., 2013). We use an open source Theano (Theano Development Team, 2016) based implementation[1] for training all our models. Word embedding size m, hidden unit size n and regularization parameters are treated as hyper-parameters. We train with different configurations based on a combination of these hyper-parameters and select the model that gives the best BLEU score on held out set of 400 conversations.

6.3 Evaluation Metric

One possible method to evaluate our models is to manually compare the generated output sequence to gold standard (collected from a held out set). However, this method is slow, human intensive and prone to errors. We wanted a method that could automatically assign a score to the generated output sequence based on how similar it is to the gold standard. BLEU (Papineni et al., 2002) is a popular metric for evaluating statistical machine translation systems and fits our needs well. A corpus level BLEU score (based on average of four grams) on a held out dataset of 400 was computed to evaluate all our models. We use the standard evaluation script [2] used by machine translation community.

Experiment	V	**BLEU4**
All-Vocab	12,603	8.24
Freq-10	1519	17.76
Freq-20	808	**18.54**
semantic-seq-20	818	21.20
syntactic-seq-20	823	29.11
ensemble-20	823	**30.15**

Table 7: BLEU score on a held out set of 400. V refers to vocabulary size

6.4 Experiments

Section 3.1 highlighted that RNN encoder decoder model parameters are dominated by the size of vocabulary $|V|$, which can make the model difficult to train on a small dataset. To evaluate the effect of a large vocabulary on a small dataset, standard RNN encoder decoder is trained. We obtain low BLEU score for this model which has 12,603 words in vocabulary (All-Vocab). For further experiments, size of vocabulary is reduced by selecting only words that occur above a minimum threshold. All the remaining out of vocabulary words (OOV) are marked as UNK. We found that restricting vocabulary further leads to a drop in BLEU score as we generate many UNK words in the output sequence. Thus, we consider this standard RNN encoder decoder model with reduced vocabulary of 808 words as our baseline model for comparison with semantic and syntactic sequence models.

[1] https://github.com/nyu-dl/dl4mt-tutorial/tree/master/session2

[2] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl

To train semantic sequence model, first all words (12,603) in original vocabulary are assigned clusters using k-means algorithm. Pre-trained word vectors (Mikolov et al., 2013) are used as word features. We use default parameters of scikit-learn (Pedregosa et al., 2011) with $k = 8$ clusters to assign the word clusters. We experimented with different cluster size, and found $k = 8$ to give the best results. Words with no word vectors are assigned a new UNK cluster, and words that are numbers are assigned a new NUM cluster. We thus have a total of 10 word clusters. Semantic sequence model shows improvement over the baseline.

Syntactic sequence model is trained by replacing OOV words in input sequence with unique UNK symbols based on its position, as described in Section 4. Maximum length of sequence symbol map for a conversation was found to be 15. Syntactic sequence model achieves significant gain in BLEU score over the baseline. We summarize all the results in Table 7.

We finally combine the best semantic and syntactic model to create an ensemble model. Ensemble model picks the output sequence which has maximum keywords overlap with NSU question $Q2$. The intuition behind this criteria is that keywords that appear in Q2 are likely to occur in the resolved question, and therefor a higher overlap of a candidate resolution with Q2 is likely to lead to a better resolution. Table 8 highlights model output for the same input sequence as generated by the best syntactic sequence and semantic sequence model. Ensemble model picks up the better among the two candidate resolutions.

Q1	A1	Q2	Gold	Syntactic	Semantic
who is the founder of usa today ?	al neuharth	and the new york times ?	who is the founder of the new york times ?	who is the founder of brazil ?	**who is the founder for the new york times ?**
where do zorse live ?	africa	and hulu ?	where do hulu live ?	what do hulu live ?	**where do hulu live ?**
who is the richest sport personality in south africa ?	ernie els	and in india ?	who is the richest sport personality in india ?	who is the richest sport in south africa ?	**who is the richest sport in india ?**
how many pounds in 125 kilograms ?	275.57375	and how many ounces ?	how many ounces are in 125 kilograms ?	how many pounds in UNK ounces ?	**how many ounces in kilograms ?**
what is the eye color of coyotes ?	yellow-brown	and that of wolves ?	what is the eye color of wolves ?	**what is the eye color of wolves ?**	what is the wolves color of ?
what does socio means ?	sociological	and what echo ?	what does echo means ?	**what does echo means ?**	what does socio start with ?
how many sides does a octagon have ?	eight	and a pentagon ?	how many sides does a pentagon have ?	**how many sides does a pentagon have ?**	how many times does a sides have ?
what is another word for portrait ?	represent	for ignorant ?	what is another word word for ignorant ?	**what is another word for ignorant ?**	what is another word for?
what is the posterior part of the brain called ?	cerebellum	and the anterior ?	what is the anterior part of the brain called ?	**what is the anterior part of the brain called ?**	what is the definition of the brain called ?
what sport originated from africa ?	wrestling	and in the united states ?	what sport originated in the united states ?	**what sport originated in the united states ?**	what sport is in a united states ?

Table 8: Model output for syntactic and semantic sequence models. Ensemble model picks the ones highlighted in **bold**

7 Conclusion

In this work we approach non-sentential question resolution in conversations as a sequence to sequence learning problem. Sequence to sequence learning models have been shown to work well when trained on a parallel corpus with millions of sentences. However, dataset of this magnitude is extremely hard to get for NSU question conversations.

We thus propose to decompose the original problem of NSU question resolution into two separate simplified problems. Each of these simpler problems focuses on an abstraction. Specifically we train a semantic sequence model that learns semantic patterns in conversations, and a syntactic sequence model that learns linguistic patterns in conversations. We finally combine the syntactic and semantic sequence model to generate an ensemble model. Our ensemble model achieves a BLEU score of 30.15 when compared to 18.54 on a standard RNN encoder decoder model with same vocabulary size.

As future work we wish to explore learning much simpler abstractions such as entity and concepts. Ensemble model is created using simple rules that pick the output sequence which has maximum overlap with NSU question. One can learn a statistical ensemble model too that uses other richer features from the simpler abstract models.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Jaime G Carbonell. 1983. Discourse pragmatics and ellipsis resolution in task-oriented natural language interfaces. In *Proceedings of the 21st annual meeting on Association for Computational Linguistics*, pages 164–168. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merrienboer, aglar Gulehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*.

Mary Dalrymple, Stuart M. Shieber, and Fernando Pereira. 1991. Ellipsis and higher-order unification. *CoRR*, cmp-lg/9503008.

Long Duong, David Chiang, Steven Bird, and Trevor Cohn. 2016. An attentional model for speech translation without transcription.

Raquel Fernández and Jonathan Ginzburg. 2002. Non-sentential utterances in dialogue: A: Corpus-based study.

Raquel Fernández, Jonathan Ginzburg, and Shalom Lappin. 2005. Using machine learning for non-sentential utterance classification. In *6th SIGdial Workshop on Discourse and Dialogue*.

Raquel Fernández. 2006. Non-sentential utterances in dialogue: Classification, resolution and use. *Unpublished Ph. D. thesis, University of London*.

Julien Gobeill, E Patsche, D Theodoro, A-L Veuthey, C Lovis, and P Ruch. 2009. Question answering for biology and medicine. In *2009 9th International Conference on Information Technology and Applications in Biomedicine*, pages 1–5. IEEE.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Bert F Green Jr, Alice K Wolf, Carol Chomsky, and Kenneth Laughery. 1961. Baseball: an automatic question-answerer. In *Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference*, pages 219–224. ACM.

Andrew Hickl, Patrick Wang, John Lehmann, and Sanda M. Harabagiu. 2006. Ferret: Interactive question-answering for real-world environments. In *ACL*.

Xu Jia, Efstratios Gavves, Basura Fernando, and Tinne Tuytelaars. 2015. Guiding the long-short term memory model for image caption generation. In *ICCV*.

Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *CVPR*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and William B. Dolan. 2016. A persona-based neural conversation model. *CoRR*, abs/1603.06155.

James MacQueen et al. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. *ICML (3)*, 28:1310–1318.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Dinesh Raghu, Sathish Indurthi, Jitendra Ajmera, and Sachindra Joshi. 2015. A statistical approach for non-sentential utterance resolution for interactive qa system. In *16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 335.

Raquel Fernández Rovira. 2006. *Non-sentential utterances in dialogue: Classification, resolution and use*. University of London.

Iulian Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Theano Development Team. 2016. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May.

Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond J. Mooney, and Kate Saenko. 2015. Translating videos to natural language using deep recurrent neural networks. In *NAACL*.

Terry Winograd. 1971. Procedures as a representation for data in a computer program for understanding natural language. Technical report, DTIC Document.

William A Woods and R Kaplan. 1977. Lunar rocks in natural english: Explorations in natural language question answering. *Linguistic structures processing*, 5:521–569.