Improve temporal reasoning for Natural Language Inference using synthetic dataset

Anonymous Author(s)
Affiliation
Address
email

Abstract

Current neural models show significant accuracy drop when tested on problems annotated as quantity reasoning, time reasoning, and negation. Its a strong evidence that these models fail to capture simple reasoning. This work aims to achieve better performance with simple time reasoning problems, by training on SNLI dataset mixed with synthesized data, which is generated by replacing time reasoning keyword and by simple time reasoning, labeled with gold relation(entailment, neutral, contradiction). More generally, the models trained in this way are able to keep its performance and achieve high performance on target problems.

1 Introduction

Current neural models trained on textual entailment dataset fails to capture general reasoning(time, quantity) that appears easy to human. Naik et al. [2018] reported state-of-the-art models perform no better than random baseline on constructed numerical reasoning inference. Nangia et al. [2017] reported all systems do relatively poor on the quantity and time reasoning section. Some typical errors in this subset are as follows:

Table 1: errors reported by Nangia et al. [2017]

Premise	Hypothesis	Relation
Like one, two, three, four.	Count from one to four.	Entailment
Payment of $1,000 (or more) may be made now or at anytime before December 31, 1993.	Payment may be made anytime until December 31.	Entailment
The time was 9:38.	The time was before 10:00	Entailment
A short time later, Nawaf and Salem al Hazmi entered the same checkpoint.	Nawaf and Salem al Hazmi entered the checkpoint ten minutes later.	Neutral

Training statistical models on limited datasets have little chance of doing general reasoning needed in natural language understanding. One of the obvious reasons is that data is always sparse on many tasks, one important insight is that there is few data on common sense fact or reasoning, as there is no need for human to say it. On the other hand, human can easily come up with sentences that has never been said before. Many neural models wrongly claims "A man wearing padded arm protection is being bitten by a German shepherd dog." entails "A man bit a dog". This error could be fixed with better statistical model, but human would easily come up with new expression like "Economy is bitten by trade war".

Instead of trying to solve the general reasoning problems, we focus on solving single simple concept: time reasoning, that is the concept of early, later, before, and after, etc.

1.1 Generating sentence pair and label

In order to reason about time, models should generalize on time concepts such as before, after, between, until, and exact time. Using natural language inference Bowman et al. [2015] setup, we explore a general way to embed relations and simple reasoning into neural model training.

We first annotate language modeling corpus with NER tagger using coreNLP Manning et al. [2014] library, then change the meaning of each sentence by replacing a word or time expressions according to a set of rules. One such example is changing the keyword 'at' in 'David arrived at 10 a.m.' to 'after' would transform it to a contradiction of the origin sentence.

1.2 Training with mixed data

We then randomly replace sentence pairs in SNLI dataset with pairs generated above to get a new dataset, where 1% of the training split is of synthetic sentence pairs. We replace original dataset with this merged one and train baseline models on it.

2 Related works

Previous works on synthetic data mainly focus on attacking NLI models to show they exploit dataset artifacts, Glockner et al. [2018] shows by replacing one word without changing meanings of sentences, performance of current state-of-art NLI models drops significantly, demonstrating limited ability to generalize on simple inference and common sense knowledge. Naik et al. [2018] construct a test set targeting linguistic phenomena that account for most errors. Especially on numerical reasoning, they show all models exhibit a significant performance drop, with none achieving an accuracy better than random (33%). For introducing external knowledge during training, Chen et al. [2018] use wordnet graph structure to enhance word sense inference.
For generating paraphrase pairs, Iyyer et al. [2018] proposed an effective to generate semantically equal sentences with desired syntax and use these sentence pairs as adversarial examples.
For synthetic dataset, Weston et al. [2015] construct a dataset generation process with constraint setup. Evans et al. [2018] shows example of creating formal synthetic structural dataset without bias, as neural network is particularly good at exploiting these biases and artifacts. CLEVR[Johnson el al., 2016] created unbiased free-form questions, using question templates families.

3 Dataset

3.1 Collecting annotated corpus

We use coreNLP Manning et al. [2014] NER tagger to collect DATE, TIME, DURATION entities. Entity tags are then used in pattern searching to filter sentences at interest. The searching patterns are composed of time expression, a set of temporal entities, and a window size. One example of such patterns is 'before_DURATION/TIME/DATE_5', where 'before' should be followed by consecutive entities of either such types, within a window size of 5 relative to keyword. For the purpose of evaluation, two different kinds of corpus(language model and news commentary) are used to generate matched and mismatched data.

3.2 Generating premise and hypothesis pairs

Based on sentences with time related expressions from above corpus, we create several transform templates for each keyword, denoting typical statements about time reasoning, and generate hypothesis sentence from them. Several examples for keyword. sentence longer than 100 tokens are removed.

Keyword	before	in
Pattern	before_DURATION/TIME/DATE_5	in_DATE_5
Entailment	['decrease_time']	['chg_before','increase_time']
Neutral	['increase_time']	NA
Contradiction	['chg_after', 'chg_at']	['chg_before', 'chg_after']

Table 2: tranformation rules example

3.3 Dataset partition

Training set: the first 200k sentences from 2011 [2011a] are annotated and transformed as training data. To balance between three classes, each class for each keyword is capped with 600 sentences.

Development set: the last 200k sentences from 2011 [2011a] are annotated and transformed development data. Each class for each keyword is capped with 20 sentences.

Test set: corpus released by 2011 [2011b] on news commentary is annotated and transformed as mismatched testset, and as well matched evaluation data.

3.4 Validating sentence pair

We sample 30 sentences from both development and test set, and validate them manually. There are 4 invalid sentences out of 60.

An invalid sample with Contradiction label is:
What if a government , instead of fighting the electricity industry alone , unleashed an economic " big bang , " trying to liberalize most markets at once ?
What if a government , instead of fighting the electricity industry alone , unleashed an economic " big bang , " trying to liberalize most markets after once?

A valid sample with Contradiction label is:
However , in Russia 's case two thirds of the rise in crime came before 1992 during the collapse of communism , and crime has stagnated after 1992 .
However , in Russia 's case two thirds of the rise in crime came at 1992 during the collapse of communism , and crime has stagnated after 1992 .

3.5 Dataset statistics

Table 3 shows basic corpus statistics. Unlike SNLI and other dataset with human writing hypothesis, this dataset does not have bias in sentence length among three classes or among premise and hypothesis, neither is there any semantic bias originated from human annotator and the way data is collected. The max and min token size is 69 and 9 for test set, 76 and 4 for training set, 63 and 9 for dev set.
One drawback of this method is imbalance between size of three classes, since the number of rules, or the difficulty of generating three classes is quite different. On the other hand, to improve linguistic diversity, future work should include more variety of corpus and balance between three classes.

keyword	**before**	**after**	**at**	**in**	**since**	**between**	**later**	**earlier**	**All**	**#tokens**
Entailment	20	20	20	20	20	20	0	0	120	29.6
Neutral	20	20	0	0	0	20	0	0	60	30.0
Contradiction	20	20	20	20	20	0	20	20	140	29.3
Entailment	83	88	305	600	600	238	0	0	1914	29.1
Neutral	83	88	0	0	0	238	0	0	409	30.8
Contradiction	250	232	600	600	600	0	528	207	3017	29.5

Table 3: key corpus statistics by labels. First part is for dev set and second part is for training set. Each dev set class is capped by 20 and each training set class is capped by 600 to balance between keywords. #tokens is mean token size of both premise and hypothesis sentences

3.6 Corpus format

Similar to Bowman et al. [2015], the corpus is given by two formats, txt and jsonl. Several fields are ignored and filled with exact copy of sentence1 and sentence2 on purpose, since they are not used in our training, these are `sentence{1,2}_parse`, `sentence{1,2}_binary_parse`.

Two major difference of this dataset with SNLI are **sentence length** and **coreference**. In SNLI premise sentence has a mean length of 14.1, and 8.3 for hypothesis. While sentence pair in this dataset is of same size and have both a mean length of 29.6. Coreference between premise and hypothesis is common because their data collection setup, this dataset however has no such assumption.

3.7 Merge with SNLI dataset

We merge this dataset with SNLI by randomly replacing sentences in SNLI, for training, dev, and test set. We control the number to be merged to 1% of the whole training set, that is 5340 of 550152; 3.2% of the whole dev set and test set, that is 320 of 10000 for both.

The dataset and generation code is available at [1]

4 Experiments

We run experiments on above mixed dataset to test the effectiveness of this method, and compared results to that of training on original SNLI dataset.

4.1 Experiments setup

We use baseline model code provided by MultiNLI baseline [2] to run training experiments on three models, on both original SNLI dataset and our mixed dataset, and compare performance on SNLI dev set, and MultiNLI dev set. Note that we use mixed training dataset and mixed dev dataset, the dev evaluation is done on mixed dev dataset.

For all experiments, we use GloVePennington et al. [2014] 840B.300d. as word vector input, and use a sequence length limit of 35, dropout rate set to 0.5, and word embedding is trainable. We experiemnt three baseline models:

CBOW: A bag-of-words sentence representation from word embeddings.

BiLSTM: The simple BiLSTM baseline model described by Nangia et al. [2017]

ESIM: This is an Enhanced Sequential Inference Model proposed by Chen et al. [2017], implemented by Nangia et al. [2017] without ensembling with a TreeLSTM.

4.2 Model performance on SNLI

Table 4 shows that mixed training keep performance on SNLI metrics as well as MultiNLI metrics, and achieve significantly higher performance on target problems. Figure 1. show that progress of CBOW training accuracy and validation accuracy for original dataset and mixed dataset are synchronized. Figure 3 in supplementary shows the results for BiLSTM and ESIM models.

Note that figures in Table 4 on mix training of BiLSTM and ESIM is tested before they reached the same step with original training due to limited time. Numbers in parenthesis show the step when these figures are tested.

[1] `https://github.com/josherich/Temporal-NLU`
[2] `https://github.com/nyu-mll/multiNLI`

Model/Dataset	SNLI train	SNLI dev	SNLI test	MultiNLI dev matched	MultiNLI dev mismatched	time matched	time mismatched
CBOW/origin	94.8	81.1	71.0	49.5	51.6	34.8	38.9
CBOW/mix	93.1	81.8	72.9	49.6	51.7	**70.8**	**69.0**
BiLSTM/origin(54200)	91.6	82.4	81.6	49.4	50.3	22.9	24.1
BiLSTM/mix(23200)	82.1	79.4	78.8	46.4	46.5	**72.7**	**71.2**
ESIM/origin(18350)	91.0	84.5	84.5	52.3	54.5	19.4	21.6
ESIM/mix(7100)	78.7	79.2	80.7	52.1	53.6	**66.5**	**65.5**
Decomp Parikh et al. [2016][3]	89.5		86.3				33.3

Table 4: Training results for origin dataset and mixed dataset. (time matched) is our dev set and (time mismatched) is our out-of-domain test set. numbers after model names are the step when models are tested.

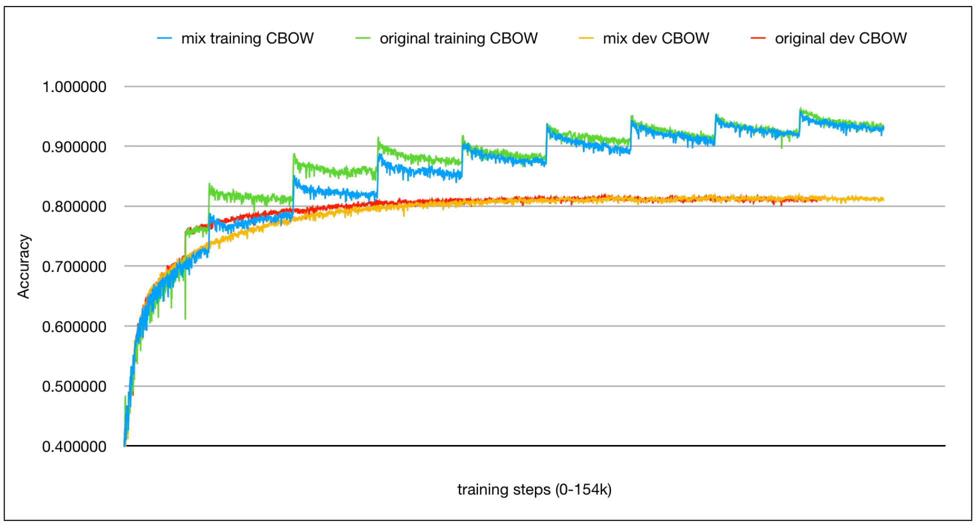

Figure 1: training progress comparison between CBOW on original dataset and mixed dataset

4.3 Error Analysis

The result on target validation set is significantly higher than that of original model, however lower than expected, mostly because limiting sentence length to 35 remove information for many longer sentences. The cutting is for shorter training time.
We examine errors made by both training approach on Nangia et al. [2017]'s annotation subset, both make most errors on contradiction golden label, that is 79%(19 out of 24) for origin training, and 60%(12 out of 20) for mix training. We show here one example which both predict entailment while the gold label is contradiction, since mix training do not include antonym information:
`Premise`: It was replaced in 1910 by the famous old pontoon bridge with its seafood restaurants, which served until the present bridge was opened in 1992.
`Hypothesis`: The famous old pontoon bridge was erected in 1920.

Another example where the mix trained model get right label contradiction while the SNLI trianed model predict neutral:
`Premise`: The call lasted about two minutes, after which Policastro and a colleague tried unsuccessfully to contact the flight.
`Hypothesis`: The call only lasted 5 seconds before it was dropped.

This might be due to more exposure to time expression in mixed training, although it certainly could due to just being lucky, as there is no such specific examples crafted in it.

4.4 Experiment on mixed percentage

We make two training set where respectively 1% and 2% of sentences is synthetic, and train CBOW model on both datasets. Table 5 shows that 2% of synthetic sentence pairs still doesn't break the training progress. It is a strong evidence that synthetic data comply with overall distribution of SNLI dataset, assuming these models generalize well in terms of inference on SNLI, or that CBOW model overfits on both parts of the dataset.

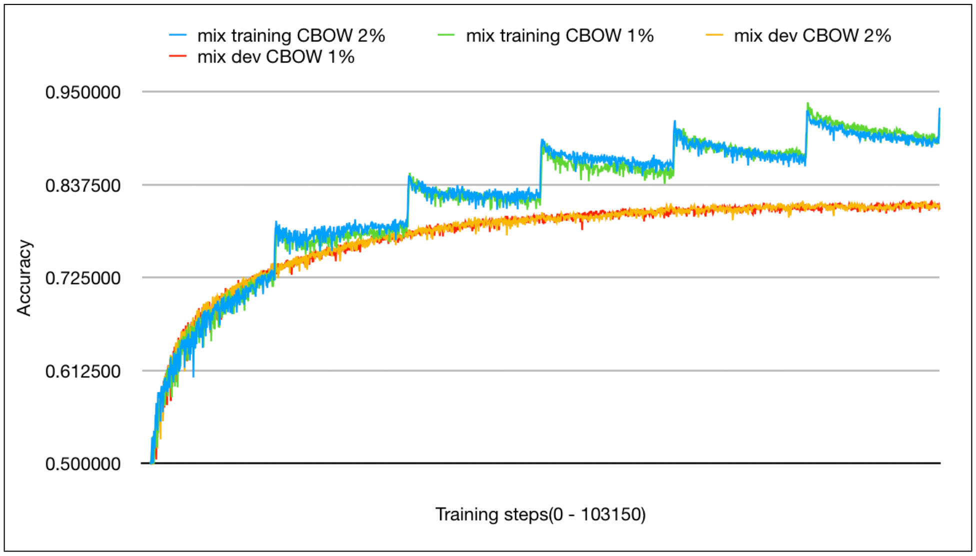

Figure 2: training progress comparison between CBOW on original dataset and mixed dataset

5 Evaluation

We evaluate the model on our test set, result is in Table 4. We then evaluate on annotated subset(matched and mismatched) of MultiNLI[Williams el al., 2017], provided by RepEval 2017 Share Task.

To show that the model potentially learn inference on cases that current neural models fail, we compare performance on difficult subset mentioned above with three baselines: 1. CBOW 2. BiLSTM. 3. ESIM.

5.1 Model performance on annotated subset

We test six below trained models on difficult annotated subset released by Nangia et al. [2017]. Although some figure favours the mixed training method, these examples are beyond scope of this method, both in terms of linguistic complexity and reasoning difficulty.

Model/Dataset	QUANTITY TIME REASONING matched	QUANTITY TIME REASONING mismatched	All matched	all mismatched
CBOW/origin	26.7	43.7	44.5	41.0
CBOW/mix	**40.0**	**53.8**	**49.3**	**50.7**
BiLSTM/origin	26.7	41.0	49.2	50.2
BiLSTM/mix	33.3	28.2	48.8	49.6
ESIM/origin	33.3	25.6	52.1	53.6
ESIM/mix	46.7	43.6	51.8	53.4

Table 5: testing results on annotated subset provided by Nangia et al. [2017], for origin dataset training and mixed dataset training.

6 Future works

6.1 Paraphrase transformation and coreference

Although this work does not implement paraphrase transformation, paraphrase is important to make sure target inference ability generalize. Glockner et al. [2018] has showed current NLI systems don't generalize well by breaking them with replacing one word with its synonym. Iyyer et al. [2018] proposed an efficient way to get controllable paraphrase. Hypothesis should also include coreference to comply with the way human communicate.

7 Conclusion

In this work, we introduct a first try to create synthetic data using simple sentential semantic transformation. We show that training NLI models on such mixed dataset improve prediction accuracy on problems at interest, while keeping performance on general NLI dataset metrics. We argue that such approach, although limited by the difficulty of semantic transforming sentences, is still promising if the problem at interest is well defined and manageable in linguistic complexity.
In another way, this method could also be used to test neural model's response to training dataset. By evaluating on synthetic dataset, this approach could serve as a sanity check for general NLI neural models. By introducing knowledge into training dataset and observing response from change of original metrics and changes of domaim specific metrics, such method could be extended to solving problems where dataset is sparse or not available.

References

M. T. S. T. 2011. language model corpus from machine translation shared task, 2011a. URL `http://www.statmt.org/lm-benchmark/1-billion-word-language-modeling-benchmark-r13output.tar.gz`.

M. T. S. T. 2011. News commentary corpus from machine translation shared task, 2011b. URL `http://www.statmt.org/wmt11/translation-task.html#download`.

S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. A large annotated corpus for learning natural language inference, 2015.

Q. Chen, X. Zhu, Z.-H. Ling, S. Wei, H. Jiang, and D. Inkpen. Enhanced lstm for natural language inference. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017. doi: 10.18653/v1/p17-1152. URL `http://dx.doi.org/10.18653/v1/P17-1152`.

Q. Chen, X. Zhu, Z.-H. Ling, D. Inkpen, and S. Wei. Neural natural language inference models enhanced with external knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*, Melbourne, Australia, July 2018. ACL.

R. Evans, D. Saxton, D. Amos, P. Kohli, and E. Grefenstette. Can neural networks understand logical entailment?, 2018.

M. Glockner, V. Shwartz, and Y. Goldberg. Breaking nli systems with sentences that require simple lexical inferences, 2018.

M. Iyyer, J. Wieting, K. Gimpel, and L. Zettlemoyer. Adversarial example generation with syntactically controlled paraphrase networks, 2018.

C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The Stanford CoreNLP natural language processing toolkit, 2014. URL `http://www.aclweb.org/anthology/P/P14/P14-5010`.

A. Naik, A. Ravichander, N. Sadeh, C. Rose, and G. Neubig. Stress test evaluation for natural language inference, 2018.

N. Nangia, A. Williams, A. Lazaridou, and S. R. Bowman. The repeval 2017 shared task: Multi-genre natural language inference with sentence representations, 2017.

A. P. Parikh, O. Täckström, D. Das, and J. Uszkoreit. A decomposable attention model for natural language inference, 2016.

J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL `http://www.aclweb.org/anthology/D14-1162`.

J. Weston, A. Bordes, S. Chopra, A. M. Rush, B. van Merriënboer, A. Joulin, and T. Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks, 2015.

8 Supplementary

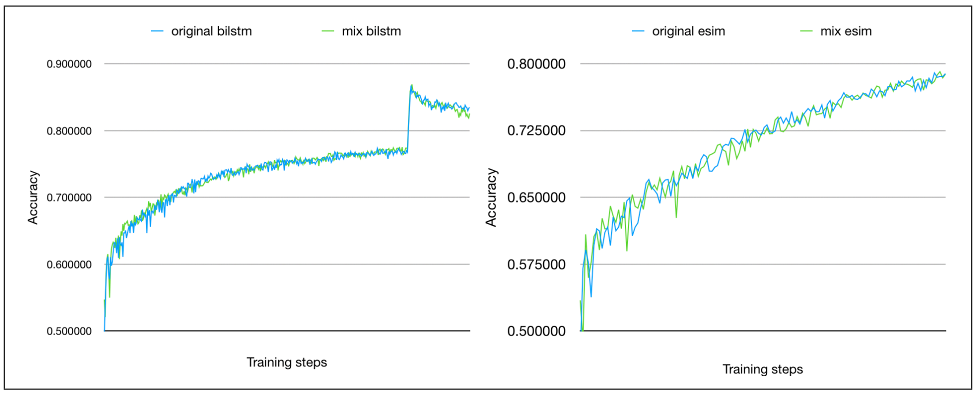

Figure 3: training progress comparison between CBOW on original dataset and mixed dataset