Phillip Merritt
CSCE 5900
Summer 2021

NLI With Tables

Intro

Natural Language Inference is an NLP task where the model must classify if a hypothesis is supported or refuted by a premise. It is a popular subject of study due to the level of reasoning in combination with information extraction required. State-of-the-art models (usually some descendant of BERT) have achieved extremely high levels of performance on NLI tests datasets.

Tables

Table data such as sport scores are a common way to communicate information. For humans interpreting the data is fairly trivial, but there are several reasons that it isn't so easy for machine learning models. Imagine a table of sports scores with columns like (year, team, head coach, games won). Take a statement like "Team x won less games than team y until they hired head coach John Smith.". Your average human could verify this without much effort while a model would have to be able to filter information by several columns of differing types at once and then apply some kind reasoning to compare the filtered information. This requires high levels of spatial, language, and numerical reasoning to accomplish.

Table NLI

InfoTabS (Gupta et al., 2020) is a new NLI dataset that uses table data as premises. Their results show that the best NLI models struggle when working with this type of information. For instance, RoBERTa achieved a 90.8% accuracy score on one of the common NLI test datasets MultiNLI yet it only achieved an average accuracy of 69.66% across the three InfoTabS test sets.

Google research (Eisenschlos et al., 2020) improved on a BERT model designed for NLI on tables called TAPAS by introducing an intermediate training step that trains on automatically generated statements based on wikipedia tables. The first of two methods for generating statements works by finding sentences referencing wikipedia tables using those as positive examples and then the same statement with a value from the table replaced by another as the negative example. The second samples from a CFG (context free grammar) that creates sql like sentences that sound somewhat natural. This resulted in a near 10% jump in performance over the state-of-the-art on TabFact and 3% on SQA.

Replication

Between the two papers linked above I chose to pursue (Eisenschlos et al., 2020) because I can't replicate InfoTabs contribution of a human annotated dataset. However, I did test each of the models they tested on their dataset and got similar scores across the board.

I implemented the method of synthetic statement generation through sampling a CFG from (Eisenschlos et al., 2020). Here is their figure of the CFG:

⟨statement⟩ → ⟨expr⟩⟨compare⟩⟨expr⟩
⟨expr⟩ → ⟨select⟩ when ⟨where⟩ |
⟨select⟩
⟨select⟩ → ⟨column⟩ |
the ⟨aggr⟩ of ⟨column⟩ |
the count
⟨where⟩ → ⟨column⟩⟨compare⟩⟨value⟩ |
⟨where⟩ and ⟨where⟩
⟨aggr⟩ → first | last |
lowest | greatest |
sum | average | range
⟨compare⟩ → is |
is greater than |
is less than
⟨value⟩ → ⟨string⟩ | ⟨number⟩

Figure 2: Grammar of synthetic phrases. ⟨column⟩ is the set of column names in the table. We also generate constant expressions by replacing expressions with their values. Aggregations are defined in Table 1.

The paper was fairly deceptive about the underlying complexity of this process. They devoted a figure and three short paragraphs to what amounts to almost 1000 lines of code (mostly uncommented) in their public repository. Even with their code available for inspiration I had to fill in a lot of gaps.

In a nutshell the entire statement is two expressions compared to each other. Each expression is either a column name, the aggregation of a column, or the count of rows followed by 0 to an unlimited number of where statements. Each expression must be capable of being evaluated into a single number so that the comparison can be made to see if it meets the desired result for positive and negative examples. If the desired result is met one of the expressions is used in its string format and the other as its evaluated version. The sampling is made randomly and anytime an issue is encountered such as the where statements filtering out all of the rows or the desired result not being met the process is started over again.

Implementation

My implementation created a class for most of the clauses you see in the CFG figure. Each being capable of being evaluated to a single value or cast a string to use in the final statement. It manages to output statements ranging from decent to questionable depending on the nature of the table it is working with. Here are some examples:

	Type	Manufacturer	Numbers	Year built	Quantity built	Power (horsepower)	Max Speed (km/h)	Note
0	RHN	Hitachi	1011-1048 (power cars) 11-48 (trailer cars)	1967	38+38	220	90	Now use as a Northeastern line commuter train.
1	RTS	Tokyu	D9-D16 (power cars) TS4-TS7 (center/trailer cars)	1971	8+4	220	70	Ex-Mahachai railways, to be refurbished. Simil...
2	THN	Tokyu, Hitachi and Nippon Sharyo	1101-1140	1983	40	235	105	Similar to NKF.
3	NKF	Nippon Sharyo, Hitachi, Fuji Heavy Industries,...	1201-1264, (center) 2101-2112	1985	64+12	235	105	Similar to THN, but with plastic chairs.
4	ASR (Class 158 Express Sprinter)	BREL, Derby Works	2501-2512, (center) 2113-2120	1991	12+8	285	120	Metre gauge version of British Rail Class 158,...
5	APD .20	Daewoo Heavy Industries	2513-2524 (center) 2121-2128	1995	10+8	298	120	First batch, narrow body.
6	APD .60	Daewoo Heavy Industries	2525-2544	1996	20+40	298	120	Second batch, wide body.

True:

1971 is less than Year built when Numbers is 2525-2544

1571 is greater than Power (horsepower) when Year built is 1971 and Max Speed (km/h) is less than 90

False:

90 is Max Speed (km/h) when Power (horsepower) is less than 285

120 is less than Max Speed (km/h) when Year built is greater than 1995

	Rank	Heat	Nation	Competitors	Time	Notes
0	1.0	2	United States	Kelly Willie, Derrick Brew, Andrew Rock, Darol...	2:59.30	Q
1	2.0	2	Nigeria	James Godday, Musa Audu, Saul Weigopwa, Enefio...	3:01.60	Q, SB
2	3.0	2	Bahamas	Andrae Williams, Dennis Darling, Nathaniel McK...	3:01.74	Q, SB
3	4.0	1	Great Britain	Timothy Benjamin, Sean Baldock, Malachi Davis,...	3:02.40	Q, SB
4	5.0	1	Japan	Yuki Yamaguchi, Jun Osakada, Tomohiro Ito, Mit...	3:02.71	Q
5	6.0	1	Germany	Ingo Schultz, Kamghe Gaba, Ruwen Faller, Basti...	3:02.77	Q
6	7.0	1	Australia	John Steffensen, Clinton Hill, Patrick Dwyer, ...	3:03.06	q
7	8.0	1	Botswana	Oganeditse Moseki, Johnson Kubisa, California ...	3:03.32	q, SB
8	9.0	2	Russia	Aleksandr Larin, Andrey Rudnitskiy, Oleg Mishu...	3:03.35	NaN
9	10.0	2	Poland	Piotr Rysiukiewicz, Piotr Klimczak, Marcin Mar...	3:03.69	NaN
10	11.0	2	Ukraine	Volodymyr Demchenko, Yevgeniy Zyukov, Myhaylo ...	3:04.01	NaN
11	12.0	1	Greece	Stilianos Dimotsios, Anastasios Gousis, Panagi...	3:04.27	SB
12	13.0	1	France	Ahmed Douhou, Ibrahima Wade, Abderrahim El Hao...	3:04.39	NaN
13	14.0	2	Spain	Eduardo Ivan Rodriguez, David Canal, Luis Flor...	3:05.03	SB
14	NaN	2	South Africa	Marcus la Grange, Hendrick Mokganyetsi, Ockert...	DNF	NaN
15	NaN	1	Jamaica	Michael Campbell, Michael Blackwood, Jermaine ...	DSQ	NaN

True:

1 is Heat when Competitors is Yuki Yamaguchi, Jun Osakada, Tomohiro Ito, Mitsuhiro Sato

the greatest of Heat when Heat is greater than 1 is less than 16

False:

1 is greater than the count when Heat is greater than 1

the first of Heat when Heat is 2 is less than 1

	Pos	No	Rider	Manufacturer	Laps	Time	Grid	Points
0	1	46	Valentino Rossi	Yamaha	21	43:06.007	2	25.0
1	2	2	Dani Pedrosa	Honda	21	4.008	1	20.0
2	3	4	Andrea Dovizioso	Honda	21	8.536	6	16.0
3	4	69	Nicky Hayden	Honda	21	8.858	4	13.0
4	5	56	Shinya Nakano	Honda	21	10.583	15	11.0
5	6	1	Casey Stoner	Ducati	21	13.64	7	10.0
6	7	65	Loris Capirossi	Suzuki	21	15.936	8	9.0
7	8	5	Colin Edwards	Yamaha	21	18.802	5	8.0
8	9	7	Chris Vermeulen	Suzuki	21	23.174	11	7.0
9	10	14	Randy de Puniet	Honda	21	25.516	9	6.0
10	11	21	John Hopkins	Kawasaki	21	27.609	10	5.0
11	12	13	Anthony West	Kawasaki	21	41.399	13	4.0
12	13	50	Sylvain Guintoli	Ducati	21	45.617	16	3.0
13	14	15	Alex de Angelis	Honda	21	49.003	17	2.0
14	15	24	Toni Elias	Ducati	21	59.139	19	1.0
15	16	33	Marco Melandri	Ducati	21	+1:03.328	14	NaN
16	17	9	Nobuatsu Aoki	Suzuki	21	+1:48.363	18	NaN
17	Ret	48	Jorge Lorenzo	Yamaha	12	Accident	3	NaN
18	Ret	52	James Toseland	Yamaha	2	Accident	12	NaN

True:

18 is greater than the lowest of Grid when Grid is greater than 11 and Laps is greater than 12

1 is the count when Time is 18.802

Grid when Grid is greater than 17 is greater than 16

False:

Laps when Laps is less than 21 and No is greater than 33 is less than 12

the greatest of Grid when Laps is less than 21 and No is greater than 24 is greater than 16

Grid when Laps is less than 21 is greater than 6

While some of the statements are clearly quite silly and can be solved without even looking at the table there is a decent amount of complexity in some of them. I think the results are suffering from the bias towards numerical values. 2 of the 3 where filters are numerical only so each where filter is more than twice as likely to filter a numerical column than a non-numerical column. If I was to work on this more I would adjust some of the probabilities to account for issues like this. It's possible some of the downsides are overcome with the sheer quantity of data you can produce with this method.

Results

Unfortunately, the dataset that they used to create the intermediate training data is technically not publicly available. They do make available data that in combination with a DataFlow pipeline on GCP can create this dataset. I did manage to accomplish this using trial credits, but it took 3 days and over a week of CPU time to complete. The result was a 175gb tfrecord file and not enough time to figure out how to process it using my implementation.

So instead I used my implementation on the tables from the SQA dataset to create a preliminary training dataset for SQA roughly 4x the size of the original dataset. Using GCP TPUs I fine-tuned a BERT medium and base for SQA with and without an intermediate training step using my data. The models were TAPAS checkpoints (called Mask-LM) and the same ones used as starting points as the researchers in the paper.

In the end the intermediate training step did not improve the BERT-medium However, the BERT-base performance improved by about 1.5 points which compared to the reported improvement of 3.4 points is not bad! Especially given the timeframe I was working with.