Maintaining Quality in FEVER Annotation

Henri Schulte, Julie Binau, Leon Derczynski
IT University of Copenhagen
Denmark
{hens,jubi,ld}@itu.dk

Abstract

We propose two measures for measuring the quality of constructed claims in the FEVER task. Annotating data for this task involves the creation of supporting and refuting claims over a set of evidence. Automatic annotation processes often leave superficial patterns in data, which learning systems can detect instead of performing the underlying task. Humans also can leave these superficial patterns, either voluntarily or involuntarily (due to e.g. fatigue). The two measures introduced attempt to detect the impact of these superficial patterns. One is a new information-theoretic and distributionality based measure, *DCI*; and the other an extension of neural probing work over the ARCT task, *utility*. We demonstrate these measures over a recent major dataset, that from the English FEVER task in 2019.

1 Introduction

The FEVER task frames verification of claims given knowledge as a retrieval and three-class entailment problem. Given a claim, supporting or refuting text must be found, and a judgment made as to whether or not the text supports the claim.

One way in which annotation performance lapses present is with the use of shortcuts. An easy shortcut for this task would be to insert a few direct negation words into claim texts, thus making them clash with the associated evidence. A recent study of ARCT, the Argument Reasoning Comprehension Task, in which systems have to pick a warrant given a claim a premise, found that annotators were prone to inserting words such as 'not' when constructing negative examples, which later models (such as BERT) could then pick up on (Niven and Kao, 2019). These superficial shortcuts were prevalent to the extent that removing this information led to a significant drop in BERT argument reasoning performance, from 77% to 50%.

Mindful of the similar nature of the ARCT and FEVER tasks, we apply an extended version of Niven & Kao's metric to the FEVER dataset, and present an information theoretic measure over skip-grams in FEVER claims to detect candidate superficial features.

2 Annotation in FEVER

The annotation process for FEVER is involved. The FEVER dataset (Thorne et al., 2018) comprises a total of 185,445 claims created from Wikipedia articles and annotated as either `SUPPORTS`, `REFUTES` or `NOTENOUGHINFO`. Additionally, claims that are labelled `SUPPORTS` and `REFUTES` also come with the evidence against which this judgement has been made. This FEVER data was created with the help of 50 annotators and in two stages: First creating claims from Wikipedia articles, then labelling them against evidence from Wikipedia. The claim generation stage entails providing annotators with a randomly sampled sentence from the introductory section of an English Wikipedia article and asking them to create claims about the article's entity. In addition to basing their claims on the provided sentence alone, annotators were also given the choice to utilize information from hyperlinked articles to allow for more complex claims (Thorne et al., 2018). Annotators were also asked to create different variants of these claims by, for example, negating, generalizing or replacing part of the claim. This was done to introduce refutable and non-verifiable claims into the dataset. While trialing, the authors realized that "the majority of annotators had difficulty generating non-trivial negation mutations [...] beyond adding 'not' to the original" (Thorne et al., 2018). We investigate the impact of these trivial negations on the quality of the dataset later on. In the second stage, annotators labeled the previously

created claims as either SUPPORTS, REFUTES or NOTENOUGHINFO. For the first two classes, annotators also marked the sentences they used as evidence for their decision. Once again, the annotators had access to articles hyperlinked in the entity's article as well. The final dataset is segmented into multiple subsets, with the training set retaining a majority of the claims at a size of 145,449. The quality of their annotations is ensured by cross-checking labels through five-way agreement, *Super-Annotators* and even validation by the authors themselves. Yet, despite spotting the issue with non-trivial negations early in the process, they do not report on any investigations into the quality of their claims. One might argue that annotation accuracy loses its importance if the task is performed on the basis of biased data. Nevertheless, as with most complex annotation tasks over language, the complex nature of this annotation process is prone to annotation exhaustion and shortcuts (Pustejovsky and Stubbs, 2012).

3 Quality Metrics

We propose two quality metrics for FEVER. The goal of FEVER data is to help train inference/verification/entailment tools that are well-generalised. Thus, a quality metric should help detect when annotated data risks being unsuitable for that purpose. The new metrics outlined here are generic and can be applied to data for other classification tasks. They are proposed with the goal of identifying surface-level linguistic patterns that 'leak' class information, helping dataset builders improve the quality of their data.

3.1 Dataset-weighted Cue Information

The first metric we propose is a simple information theoretic measure of how much a pattern contributes to a classification. In this case, patterns are extracted using skip-grams. These capture a good amount of information about a corpus (Guthrie et al., 2006) while also giving a way of ignoring the typically-rare named entities that are rich in FEVER claims and focusing on the surrounding language. The metric is the weighted inverse information gain of a skip-gram relative to a pair of classes. Weighting is determined by the frequency of documents bearing the skip-gram in the corpus, which normalises skew from highly imbalanced but rare phrases. For dataset D and cue k, where cues are e.g. skip-gram features:

$$IG(D,k) = H(D) - H(D|k) \tag{1}$$

We are interested in items that cause high information gain, i.e. $1 - IG(D,f)$.

This should be weighted with the impact that a pattern can potentially have in a given dataset and split. For this reason, feature counts should be normalised by the size of each class. That is, when calculating entropy:

$$H(X) = -\Sigma_{i=1}^{n} P(x_i) log P(x_i) \tag{2}$$

Let $D_{cue=k}$ be the set of data bearing cue k, and $D_{class=y}$ be the set of data with class label y drawn from the set of class labels Y. The normalised distribution N of cue frequencies for cue k is:

$$N = \{\frac{|D_{cue=k} \cap D_{class=i}|}{|D_{cue=k}|} | i \in Y\} \tag{3}$$

Given this class-balanced dataset weighting, we can then define the information-based factor λ_h trivially thus:

$$\lambda_h = 1 - H(N) \tag{4}$$

A term is also required to correct for the rareness of features. Features that occur only for one class, but are seldom, should not receive a high value. On the other hand, knowing that features in language typically follow a Zipfian frequency distribution (Montemurro, 2001), one should still have useful resolution beyond the most-frequent items. Thus we specify a frequency-based scaling factor λ_f as a root of the scaled frequency weight:

$$\lambda_f = (|d\hat{k} : d \in D||D|^{-1})^{\frac{1}{s}} \tag{5}$$

Where s is a scaling factor corresponding to the estimated exponent of the features' power law frequency distribution. For English, $s = 3$ gives reasonable results (i.e. taking the cube root).

These two are combined taking their squared product to form DCI:

$$DCI = \sqrt{\lambda_h \times \lambda_f} \tag{6}$$

A note regarding language: in this case, we consider 1, 2, and 3-grams, with skips in the range of $[0, 2]$. This is suitable for English; other languages might benefit from broader skip ranges.

3.2 Cue Productivity and Coverage Probes

We follow the approach of Niven and Kao (2019) in determining a productivity and coverage score for each cue in the data. As the structure of their dataset is fundamentally different from the dataset presented in Thorne et al. (2018), we have made amendments to their methodology in order to attain comparable results.

As in Niven and Kao (2019), we consider any uni- or bigram a potential cue. We extract these cues from the claims in the dataset and take note of the associated label. This allows us to calculate the *applicability* of a given cue (α_k), which represents the absolute number of claims in the dataset that contain the cue irrespective of their label. Let $\mathbb{T}$ be the set of all cues and n the number of claims.

$$\alpha_k = \sum_{i=1}^{n} \mathbb{1}\left[\exists k \in \mathbb{T}\right] \tag{7}$$

The *productivity* of a cue (π_k) is the frequency of the most common label across the claims that contain the cue. In practical terms, the productivity is the chance that a model correctly labels a claim by assigning it the most common label of a given cue in the claim.

$$\pi_k = \frac{\max\left[\sum_{i=1}^{n} \mathbb{1}\left[\exists j, k \in \mathbb{T}_j\right]\right]}{\alpha_k} \tag{8}$$

From this definition productivity may be in the range $[\frac{1}{m}, 1]$ where m is the number of unique labels – three in our case. The coverage of a cue (ξ_k) is defined by Niven and Kao (2019) as $\xi_k = \alpha_k / n$. We retain this definition with the caveat that, due to the fundamentally different architecture of the data, we derive α_k differently.

This approach assumes a balanced dataset with regard to the frequency of each label. If executed on an imbalanced dataset, a given cue's productivity would be dominated by the most frequent label, not because it is actually more likely to appear in a claim with that label but purely since the label is more frequent overall. We generate a balanced sample by undersampling majority classes. In order to not discard data from the majority classes, however, we repeat the process ten times with random samples. We find that this is a better compromise than oversampling minority classes or introducing class weights when calculating productivity, as those methods inflate the productivity of rare cues that appear exclusively in the smallest class.

Cue	Productivity	Coverage
a	0.36	0.34
is	0.38	0.32
in	0.37	0.30
the	0.36	0.26
was	0.35	0.25

Table 1: Top five cues by coverage

Cue	Productivity	Coverage
not	0.86	0.04
only	0.90	0.04

Table 2: High-productivity cues

Productivity values alone are not necessarily comparable across datasets. Niven and Kao (2019) acknowledge that a cue is only useful to a machine learning model if $\pi_k > 1/m$. In their case, every claim can have two possible labels, i.e. $m = 2$. For the FEVER dataset three labels exist. This means that the productivity threshold at which cues start becoming useful to a model is higher in the ARCT task. We should therefore actually consider the *utility* of a cue to the model (ρ_k).

$$\rho_k = \pi_k - \frac{1}{m} \tag{9}$$

4 Running the metrics

4.1 Neural Probe Results

We apply the described methodology to the FEVER training dataset presented in Thorne et al. (2018) and thereby determine productivity and coverage for 14,320 cues. Considering the cues with a productivity of 1, i.e. cues that could predict the label with a 100% accuracy, is not particularly relevant as none of them have a coverage over 0.01, meaning that they only appear in $\leq 1\%$ of claims. In fact, there are 12,126 cues that only ever appear with one label ($\approx 85\%$).

Table 1 shows the cues with the highest coverage. It is dominated by common English stop words with productivity near the minimum of $\frac{1}{3}$. This means that to a machine learning model these cues provide very little utility in finding a shortcut. Some of the more common cues do still provide some utility though. The cues "an", "to" and "and" each appear in 6-8% of all claims and provide 0.44, 0.53 and 0.49 productivity respectively.

Cue	Utility	Coverage	Harmonic Mean
to	0.19	0.07	0.10
an	0.10	0.08	0.09
and	0.15	0.06	0.09
is	0.04	0.32	0.08
not	0.53	0.04	0.07
only	0.56	0.04	0.07

Table 3: Top seven cues by harmonic mean of utility and coverage

These values pale, however, in comparison to the slightly less common but considerably more productive cues "not" and "only" (see table 2). While these only have a coverage of 0.04 each (Table 2, they provide productivity of 0.86 and 0.90 respectively. Even though Thorne et al. (2018) explicitly mention that they attempted to minimize the use of "not" for the creation of refuted claims, we find that in our sample claims containing "not" were labelled `REFUTES` 86% of the time. We find no other cues with comparable coverage to reach such high productivity.

Niven and Kao (2019) find that in the Argument Reasoning Comprehension Task (ARCT) dataset (Habernal et al., 2018) the cue "not" has a productivity of 61% and coverage of 64%. In the FEVER training data "not" to has a higher productivity but lower coverage.

For "not" this provides a utility value of ≈ 0.11 in ARCT and ≈ 0.53 in the train set of FEVER, meaning that in the FEVER data the cue provides a significantly higher utility to a ML model.

This conclusion is only drawn from the utility alone though. For the sake of comparability across both utility and coverage, we condense these values to one metric by taking their harmonic mean. We choose the harmonic mean as it assigns higher values to cues that are **both** utilisable and covering. For "not" this results in ≈ 0.19 in ARCT and ≈ 0.07 in the FEVER training data.

Considering cues by their harmonic mean of utility and coverage suggests that despite their high productivity, "not" and "only" might not be the most relevant cues in the data, being preceded by common stop words that yet provide noticeable utility (see Table 3).

Besides "not", some relatively neutral, such as "to" and "and", also appear in a somewhat imbalanced manner. In fact, in our samples 53% of claims containing "to" are labelled as `REFUTES` and 49% of claims containing "and" are labelled as `SUPPORTS`. These distributions are hard to predict. We therefore encourage analyses of this during dataset construction.

DCI	Classes	Skipgram
	unigrams	
0.5830	support/refute	only
0.5684	refute/not enough	not
0.4953	support/refute	not
0.4860	refute/not enough	only
0.4564	support/refute	incapable
0.4486	support/not enough	person
	skip-2-bigrams	
0.3278	refute/not enough	('is', 'not')
0.3226	support/refute	('only', '.')
0.3212	refute/not enough	('not', 'a')
0.3103	support/refute	('There', 'a')
0.3100	refute/not enough	('not', '.')
0.3052	support/refute	('only', 'a')
	skip-2-trigrams	
0.2511	refute/not enough	('is', 'not', '.')
0.2503	refute/not enough	('is', 'not', 'a')
0.2488	refute/not enough	('not', 'a', '.')
0.2466	support/refute	('There', 'is', 'a')
0.2396	support/refute	('is', 'not', '.')
0.2347	support/refute	('only', 'a', '.')

Table 4: Highest DCI skip-grams, i.e. most class-informative superficial features, in the English FEVER dataset

4.2 DCI Results

DCI enables ranking of superficial n-grams. Table 4 presents the most informative superficial patterns in the FEVER data. We can see that "not" plays a prolific role, especially as part of a trigram. This might be what one would expect given the high utility of this word (Table 3). Both support/refute and refute/not-enough-data partitions give the most highly-ranked skip-grams; support/not-enough-data doesn't generate annotation artefacts as frequently.

5 Discussion

Applying productivity, utility and coverage indicates a dearth of the sort of superficial features in FEVER that were present in previous tasks (namely the ARCT dataset). This is somewhat at odds with other work over FEVER. Schuster et al. (2019) find that local mutual information (LMI) reveals some

n-grams that are strongly-associated with negative examples, and are able to predict claim veracity based on claims alone. The phrases that Schuster et al. find match those top-ranked by our DCI metric.

We can therefore see that mutual information-based measures (LMI, DCI) find different biases to frequency-associative measures, such as those use to find cues in the ARCT task. It may be worth applying e.g. LMI or DCI to the ARCT data to see if complementary results emerge.

Note that we examine all n- and skip-grams in the dataset, without smoothing. Suntwal et al. (2019) experiment with removing named entities and rare noun-phrases from their dataset when training models. While this is likely to reduce variances in the data representation, enhancing the signal, the goal of this work is to find the strongest signals, and go down from there, rather than remove noise in a "bottom-up" fashion.

This is not the first investigation into biases related to crowdsourcing and human annotation: Belinkov et al. (2019) find patterns in corpora for inference. Sabou et al. (2014) and Bontcheva et al. (2017) discuss best practices in crowdsourcing for corpus creation. Notably, the number of annotations created by a single annotator should be capped strongly, to avoid nuances of a single worker's style disrupting the data significantly – rather, many annotators should contribute to the data. We propose further controlling quality by looking for superficial patterns during the annotation process, and asking annotators to consider re-formulating their input choices if such patterns are present.

6 Conclusion

Annotators are prone to introducing artefacts, certainly in the construction of datasets involving synthesis of claims and counterclaims. This paper presented metrics and an analysis of the English FEVER dataset with three previously-used measures: productivity, coverage and utility; and a new measure, dataset-weighted cue information. We find that the FEVER dataset is somewhat free of superficial artefacts, and present a truncated set of its most-informative (or most distracting) patterns.

Acknowledgments

This research was partly supported by the Danmarks Frie Forskningsfond project Verif-AI.

References

Yonatan Belinkov, Adam Poliak, Stuart M Shieber, Benjamin Van Durme, and Alexander M Rush. 2019. On adversarial removal of hypothesis-only bias in natural language inference. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 256–262.

Kalina Bontcheva, Leon Derczynski, and Ian Roberts. 2017. Crowdsourcing named entity recognition and entity linking corpora. In *Handbook of Linguistic Annotation*, pages 875–892. Springer.

David Guthrie, Ben Allison, Wei Liu, Louise Guthrie, and Yorick Wilks. 2006. A closer look at skip-gram modelling. In *LREC*, pages 1222–1225.

Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2018. The argument reasoning comprehension task: Identification and reconstruction of implicit warrants. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1930–1940.

Marcelo A Montemurro. 2001. Beyond the zipf–mandelbrot law in quantitative linguistics. *Physica A: Statistical Mechanics and its Applications*, 300(3-4):567–578.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664.

James Pustejovsky and Amber Stubbs. 2012. *Natural Language Annotation for Machine Learning: A guide to corpus-building for applications*. O'Reilly Media, Inc.

Marta Sabou, Kalina Bontcheva, Leon Derczynski, and Arno Scharl. 2014. Corpus annotation through crowdsourcing: Towards best practice guidelines. In *LREC*, pages 859–866.

Tal Schuster, Darsh Shah, Yun Jie Serene Yeo, Daniel Roberto Filizzola Ortiz, Enrico Santus, and Regina Barzilay. 2019. Towards debiasing fact verification models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3410–3416.

Sandeep Suntwal, Mithun Paul, Rebecca Sharp, and Mihai Surdeanu. 2019. On the importance of delexicalization for fact verification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3404–3409.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: A Large-scale Dataset for Fact Extraction and VERification. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pages 809–819.