Author's Emotion as a Feature in Event Factuality Prediction

Samuel Haymann, Lizzie Liang, Erik Andersen
{shaymann, yliang, erikandersen}@brandeis.edu
COSI 140

1. Introduction

Factuality in the scope of this project refers to whether or not a specific event has happened or will occur in the future. Events are the main predicates in a sentence that convey a change in status. For example, in a sentence such as (1) *It rained.* the event is *rained* and the change in status is from not raining to having rained. Regarding this event, we can tell that its factuality is positive, as we can see that *rained* is something that did happen. Conversely, for a sentence such as (2) *It didn't rain.* the factuality of the event *rain* is negative, i.e. that it did not happen.

The topic of factuality annotation and prediction has been addressed several times over the past decade, ranging from FactBank (Saurí and Pustejovsky, 2009), which was published in 2009, to more recent projects such as the Universal Decompositional Semantics It Happened v2 (UDS-IH2; Rudinger et al., 2018), which was published in 2018. All of these annotations capture whether or not a given event happened and assign each event a score. In some annotations, like FactBank, the score reflects how certain the source reporting the event is that the event happened or not, while other annotations, like UDS-IH2, capture how confident the annotator is of their factuality annotation.

In this project, we aim to build on this previous work by annotating the same concepts (the event, the factuality, a score of confidence, the source reporting the event) and by additionally capturing the emotion of the source toward the reported event. We will then use this additional annotation to test whether or not it improves our ability to predict the factuality of events.

2. Annotation

The corpus for this project consists of 200 tweets from Donald Trump's twitter account (@realDonaldTrump) selected randomly from 2018, with the only criterion being that each tweet must be longer than 60 characters. Rather than querying the Twitter API directly, these tweets were obtained from the Trump Twitter Archive[1], which collects every tweet posted from this account (including any that may have been deleted) and makes them available publically in JSON format.

The annotation guidelines[2] consist of three main tasks, each requiring annotators to select text, tag it appropriately, and add additional information as necessary. The topic of those three tasks are:

Source The source can be identified as the person or entity that is reporting about the event. This could either be the source of the text (i.e. the author), or it could be a source

[1] http://www.trumptwitterarchive.com/

[2] See Appendix for guidelines

mentioned in the text via hearsay, often paired with a verb such as: “reported that”, “said that”, “according to”, etc.

Event As mentioned above, the event is the main predicate that conveys a change in status. In addition, each event is assigned a factuality score from -2 to +2. Here, -2 indicates that the source has high confidence the event has not happened or will not happen. Conversely, +2 indicates that the source has high confidence the event has happened or will happen. Scores of +1 and -1 are used when the certainty of the event is not as strong. Finally, 0 is used if the factuality cannot be discerned.

Emotion Any word that overtly expresses the source’s emotion toward an event. In addition, each emotion is assigned a sentiment of either positive or negative.

Three annotators with, at minimum, a core linguistics background worked on this project. Not all three were native English speakers, but all were fluent. At the onset, each annotator was given 20 tweets to annotate. Due to inconsistencies between each annotator, as well as between the annotations and the intent of the guidelines, the guidelines were revised, some of the instructions were simplified, and feedback was provided before moving forward with the project. Subsequently, each annotator was given 120 tweets from the remaining 180 so that there would be an overlap of two annotators per tweet.

Next we calculated the inter-annotator agreement (IAA) scores with script. The IAA.py file takes text files processed from the annotations using another script parse_data.py, and computes the IAA using Cohen’s Kappa (since there is an overlap of two annotator per file.) The script IAA.py exposes the following methods:

- get_data(file, coder)
  Reads the text files parsed from the xml files.
  Returns three dictionaries. Each dictionary contains annotations from each annotator.
- get_agreement()
  Computes Cohen’s Kappa score between annotators 1 and 2, 2 and 3, 1 and 3 using cohen_kappa_score from sklearn.metrics package.

After the second round of annotations, the IAA scores improved, but they nevertheless remained too low to automatically build a gold standard. The Cohen’s Kappa score across all of the tags was at an average of .27 (see Table 1), while the event only was at an average of .21 (see Table 2). It is likely that the complexity of annotating the event, as well as the time constraint in finishing the project were major contributions to these scores. As a comparison, TimeML[3], which is used for event annotation in FactBank, had an agreement of about .80[4] across two experienced annotators.

[3] http://timeml.org/
[4] http://www.timeml.org/timebank/documentation-1.2.html

annotators	κ
1 and 2	0.2149
1 and 3	0.2757
2 and 3	0.3265
avg	0.2724

Table 1: Cohen's Kappa of IAA across all annotations

annotators	κ
1 and 2	0.1809
1 and 3	0.2049
2 and 3	0.2544
avg	0.2134

Table 2: Cohen's Kappa of IAA of event annotations only

In order to train our event factuality prediction model, we created a gold standard corpus by manually adjudicating the annotations and adding in any missing information. The distribution of annotations in the gold standard corpus can be seen in Table 3.

annotation	subtotal	total
events		**408**
+2	297	
+1	30	
0	17	
-1	9	
-2	55	
emotions		**73**
pos	60	
neg	13	
sources		**197**
author	148	
other	49	

Table 3: Number of annotations in gold standard corpus

3. Feature Engineering

What and Why

MAE[5] is a great tool for annotation purposes, but its output still needs processing before it can be fed into our model. The output from is in XML files. The actual tweets are in the tag <TEXT> </TEXT>, while the annotated data (event, source, emotion, and the links between source and event, emotion and event) are stored in another tag called <TAGS>. In order for our model to ingest the data, we needed to be able to associate the tags back to their context. The XML has a field called *spans* for each tag if text associated with it was highlighted in the context. This field provides the text position in the entire document. In order to use the field to associate the event back to each tweet, or sentence, we found the information according the *spans* based on the position relative to the entire document. Similarly for the source and emotion, since the annotators were asked to highlight the source and emotion, the *spans* field was also used to find the position of each in each sentence. In addition to associating the source and emotion back to the sentence, it was also important to link the source and emotion to an event, as there may be more than one event per sentence and each event should have a source. For events that do not have any associated emotion words, we kept a blank space in the parsed file to keep the format consistent.

Implementation

In order to generate the text files to input into the model, the XML files from MAE were parsed into pipe delimited text files. These files contain the tweets followed by a row containing the event with the source, confidence, emotion, and emotion value. The data parser was built as a separate script. The input are the XML files and output of the data parser are the delimited text files. The input XML files are explored using xml.etree.ElementTree built-in python module. There is a class called Parse in the parse_data.py script and it supports following methods:

- parse(file)
  Loads the xml file. It extracts information such as event, event id, spans of the event, tweet context, event value, confidence, source, source id, spans of source, source text, emotion, emotion id, spans of emotion, emotion value, and the linkages of event and source, event and emotion.
  Creates a dictionary that contains the fields mentioned above.
  Calls the method write_file(annotation, file) that takes the dictionary and writes the fields into a outputting file.

[5] https://github.com/keighrim/mae-annotation

- write_file(annotation, file)
  Calls _get_tweet(text, events) to line up fields such as source, emotion to each event.
  Writes fields from parse() to a pipe delimited file.
- _get_tweet(text, events)
  It is a private method that used internally to align event, source, emotion etc. to each sentence.
  Returns a dictionary with source, emotion aligned with each event.

Sample Output

annotation_id|tweet_content|event_span|event_text|event_confidence|source|emotion|emotion_span|emotion_value

969523385016471552|969523385016471552
Fri Mar 02 10:42:42 +0000 2018
[Trump]
Alex Baldwin, whose dieing mediocre career was saved by his impersonation of me on SNL, now says playing DJT was agony for him.
Alex, it was also agony for those who were forced to watch.
You were terrible.
Bring back Darrell Hammond, much funnier and a far greater talent!|107~112|saved|+2|Trump|mediocre|87~95|negative

969523385016471552|969523385016471552
Fri Mar 02 10:42:42 +0000 2018
[Trump]
Alex Baldwin, whose dieing mediocre career was saved by his impersonation of me on SNL, now says playing DJT was agony for him.
Alex, it was also agony for those who were forced to watch.
You were terrible.
Bring back Darrell Hammond, much funnier and a far greater talent!|241~246|watch|+2|Trump|agony|206~211|negative

4. Model

Due to the small quantity of training data, we decided to limit our factuality prediction to only positive {+2,+1} or negative {-1,-2}. In order to do so, we trained a logistic regression model to predict the factuality of a given event using the scikit-learn library. Because of how skewed the data was (we had many more positive instances than negative instances), we made sure to

partition the data such that both the train and test data included an appropriate number of positive and negative instances. We split our data into 77% training and 23% test for the negative data, and 73% training and 27% test for the positive data. Note the small differences in proportion between the two: we found that splitting the two based on *slightly* different proportions resulted in slightly better results. However, if these proportions are skewed too much, the model receives an erroneous idea as to the distribution of positive and negative values; as we used a balanced logistic regression model (i.e. *class_weight='balanced'*), skewing the split too much would have disastrous effects on our model. The model was trained with the following features:

1. A continuous bag of words using a window from 6 tokens before the first word of the event up to (but not including) the second token after the event. We tried experimenting with removing stopwords, but this had no visible effect.
2. Negatives that appeared within the same window around the event. We tried to weight this higher than other features by adding 10 to each occurrence.
3. Auxiliaries that appeared within the same window around the event. We tried to weight this higher than other features by adding 10 to each occurrence.
4. Use of a dummy auxiliary in case no negatives or auxiliaries appeared in this window. We tried to weight this higher than other features by adding 10 to each occurrence.
5. The last two characters of the unstemmed event.
6. All bigrams within the same window around the event
7. Whether a word was present in the emotion column (unstemmed or stemmed), and if so, its corresponding value. The model was run both with and without the emotions as features to test our hypothesis as to whether the emotion of the source towards an event could help predict its factuality.

5. Results

Due to the low number of negative factuality events in the training data, the model had difficulty in predicting those events, and by the same token did much better at predicting positive factuality events because of the overwhelming number of positive events, and the overall low number of negative factuality events in our corpus overall. However, we did test both with and without the emotion as features in our model, and the results can be seen in Table 4 and Table 5. Neither the precision, recall, nor f1-score of either the positive or negative categories changed at all when the emotion features were added. Although not conclusive, as our model needs work, this leads us to conjecture that emotion probably does not play a role in predicting factuality.

	precision	recall	f1-score	support
neg	0.50	0.23	0.32	30
pos	0.74	0.90	0.81	72
micro avg	0.71	0.71	0.71	102
macro avg	0.62	0.57	0.57	102
weighted avg	0.67	0.71	0.67	102

Table 4: Results without emotion as a feature

	precision	recall	f1-score	support
neg	0.50	0.23	0.32	30
pos	0.74	0.90	0.81	72
micro avg	0.71	0.71	0.71	102
macro avg	0.62	0.57	0.57	102
weighted avg	0.67	0.71	0.67	102

Table 5: Results with emotion as a feature

As can be seen in the above tables, removing the emotion features resulted in no change in results for our model.

As an experiment, we also tried to use "no_emotion" as a feature, to see if it might have some effect, but this ended up having no effect. Our final model leaves the *no_emotion* feature commented out, as we felt that it should not be a good predictor for positive/negative factuality.

6. Challenges

Some of the biggest challenges were the size of the data and the building of the model. As we did not have a big training dataset to make the results statistically significant, it was difficult to determine which aspects of the small training set would be beneficial for a logistic regression model. With regard to our data size, although we asked the annotators to work on 200 tweets, a size of 200 is not ideal in order to generalize the result. Even though we had more than 200 events, some of the tweets were just fragments, so they did not necessarily have an event at all. Additionally, we needed to split the data up for testing, which resulted in the test number that can be seen above (98). Furthermore, we had very few emotion annotations, so it is possible that the lack of any correlation found between factuality and emotion was simply because we had so few emotion annotations in the test set.

As can be seen from the results of our model, fitting the data into an effective model was a very challenging aspect of this project. While we knew that negatives, such as *'no', 'not',* etc.

were likely to appear near events with negative factuality, there were some exceptions to this rule. For example, the event *'NO TAPES'* in the August 14 tweet collection has a factuality of +2, even though the word *'no'* appears in the event. The presence of *'no*' may help classify the next event, *'used'* as negative, but the model will most likely erroneously classify events such as *'NO TAPES'* as positive.

It was also difficult to select the event to use for determining our window for features. Though rare, as tweets are short, there are some tweets that have multiple instances of the same event word - which in some cases might throw the model off. For example, the Mar 2 tweet collection has a sentence as follows: *I have decided that sections of the Wall that California wants built NOW will not be built until the whole Wall is approved.* Notice how *built* appears twice, though our gold data identifies the event as the second instance. We tried working with the spans to determine the exact place of the event, but this proved difficult as the spans are based on the position of the characters in the original xml files, and this includes tweets that were thrown out for having no events in them. We tried both selecting all instances of the first token of the event (in this instance *built*) within the tweet as well as selecting only the first, but ultimately there appeared to be no visible difference between these, most likely because the former distorts the model by including additional features that have nothing to do with the event, whereas the latter may erroneously include only features that do not relate to a given event. Extending the window to 4, which would include *not* in the event window for the above sentence, did not help either, resulting in lower scores for the model.

Another challenge is the inconsistency of the annotation. Our annotation task was relatively complex, and it required a number of steps. We asked our annotators to tag the event, source associated with the event, a confidence score indicating how confident that the annotator felt the event has happened according to the source, and emotion. Although we have written a very detailed set of guidelines, some of the annotations are subjective. Therefore, the IAA score is low between each annotator pair. These inconsistencies made coming up with the gold standard annotation even more challenging, and it also made it difficult for our model, which initially performed better on a single annotator's data.

7. Improvements

Increasing the IAA score would be vital to annotating more content in order to train a model on a larger corpus. One way this might be achieved may be by separating the task of event annotation and factuality. Event annotation could be done manually by following another guideline such as TimeML, or it could potentially be done automatically by algorithms trained on a TimeML corpus, or could also be done by trying to parse sentences into Universal Dependency treebanks and using these structures to focus on specific predicates such as was done in UDS-IH2. This would leave the task of evaluating factuality which by itself should prove to be more consistent across annotators.

A similar approach could be taken with words representing the author's emotion. That is, a sentiment dictionary could be used to highlight potential words that the annotator could confirm as being the author's emotion towards an event, and could further confirm its positive or negative sentiment. Alternatively, rather than trying to annotate all emotional words, the task could be narrowed (and automated) to specific words that the author uses frequently. For example, in the case of Trump, expressive words such as "loser", "terrible", "stupid", "weak" are common across many tweets.

As mentioned in the Results section, the uneven distribution of positive and negative factuality data has impacts on the ability to train a model. In order to mitigate this issue the data could perhaps be preprocessed for specific patterns such as negated verbs to try and even the data to create a balanced corpus.

In terms of automated selection of the event in the context of the tweet, which we found was particularly difficult, we may benefit in the future from using a dependency parser to identify the uppermost heads of the sentence as event candidates, though this would not be without its own share of problems.

Furthermore, as a last resort, we may want to employ a different model style. We used logistic regression because we knew it was a good place to start when creating models for machine learning. We also tried to use a Support Vector Machine and a logistic regression cross-validation model, both using sklearn, but these both resulted in worse results than those presented above. Given more time, we would like to experiment with new models outside of the scope of the class, such as the Random Forest Classifier.

8. Conclusions

We have proposed a method to extend the work that has been done on factuality annotation in order to test additional features in the training of an event factuality prediction model. Our results at this point seem to discourage a correlation between the two, but more data will be needed to determine the viability of this approach.

Future work could extend this project to multiple authors to understand whether the relationship between an author's emotion and event factuality is indicative of writing style or if broader patterns exist. Additionally, this annotation could also be used to analyze if factuality can be used to predict the emotion of the author.

9. References

Saurí, R. and Pustejovsky, J., "FactBank: a corpus annotated with event factuality," Lang. Resour. Eval., vol. 43, no. 3, p. 227, May 2009.

Rudinger, R., White, A.S., & B. Van Durme. 2018. Neural models of factuality. Proceedings of NAACL-HLT 2018, pages 731–744. New Orleans, Louisiana, June 1-6, 2018.

10. Appendix

See the Factuality Annotation Guidelines PDF for full guidelines on annotating factuality and emotion.