A Mostly Unlexicalized Model For Recognizing Textual Entailment

Mithun Paul, Rebecca Sharp, Mihai Surdeanu
University of Arizona, Tucson, AZ, USA
{mithunpaul,bsharp,msurdeanu}@email.arizona.edu

Abstract

Many approaches to automatically recognizing entailment relations have employed classifiers over hand engineered lexicalized features, or deep learning models that implicitly capture lexicalization through word embeddings. This reliance on lexicalization may complicate the adaptation of these tools between domains. For example, such a system trained in the news domain may learn that a sentence like "Palestinians recognize Texas as part of Mexico" tends to be unsupported, but this fact (and its corresponding lexicalized cues) have no value in, say, a scientific domain. To mitigate this dependence on lexicalized information, in this paper we propose a model that reads two sentences, from any given domain, to determine entailment without using lexicalized features. Instead our model relies on features that are either unlexicalized or are domain independent such as proportion of negated verbs, antonyms, or noun overlap. In its current implementation, this model does not perform well on the FEVER dataset, due to two reasons. First, for the information retrieval portion of the task we used the baseline system provided, since this was not the aim of our project. Second, this is work in progress and we still are in the process of identifying more features and gradually increasing the accuracy of our model. In the end, we hope to build a generic end-to-end classifier, which can be used in a domain outside the one in which it was trained, with no or minimal re-training.

1 Introduction

The rampant spread of fake data recently (be it in news or scientific facts) and its impact in our day to day life has renewed interest in the topic of disambiguating between fake, or unsupported, information and real, or supported, information (Wang, 2017).

Last year, the fake news challenge (Pomerleau and Rao, 2017) was organized as a valuable first step towards creating systems to detect inaccurate claims. This year the fact verification challenge (FEVER) (Thorne et al., 2018) was organized to further this. Specifically it was organized to foster the development of systems that combine information retrieval (IR) and textual entailment recognition (RTE) together to address the fake claim problem. However, developing a system that is trained to tackle the issue only in one area (in this case for fake news detection) does not solve the problem in other domains. For example, models developed to specifically detect fake news might not work well to detect fake science articles.

An alternative will be to create systems that can be trained in broader domains and can then be used to test on specific domains with minimal modification and/or parameter tuning. Such a system should also be able to capture the underlying idiosyncratic characteristics of the human author who originally created such fake data. For example some common techniques used by writers of such fake articles include using hedging words (e.g., *possibly*), circumventing facts, avoiding mentioning direct evidence, hyperbole etc. To this end, we propose a largely *unlexicalized* approach that, when coupled with an information retrieval (IR) system that assembles relevant articles for a given claim, would serve as a cross-domain fake-data detection tool which could either stand-alone or potentially supplement other domain-specific lexicalized systems.

The goal of this paper is to present a description of such a preliminary system developed for the FEVER challenge, its performance, and our intended future work.

Proceedings of the First Workshop on Fact Extraction and VERification (FEVER), pages 166–171
Brussels, Belgium, November 1, 2018.

2 Approach

We approach the task of distinguishing between fake and real claims in a series of steps. Specifically, given a claim, we:

1. **Information retrieval (IR):** We use an IR component to gather claim-relevant texts from a large corpus of evidence (i.e., Wikipedia articles). For retrieving the Wikipedia articles which contain sentences relevant to the given claim, we reused the competition-provided information retrieval system (Thorne et al., 2018) since we are focusing here on the RTE portion of the task. To be specific, we used the DrQA (Chen et al., 2017) for document retrieval. For sentence selection the modified DrQA with binning with comparison to unigram TF-IDF implementation using NLTK (Bird and Loper, 2004) was used. The *k* and *l* parameters values, for document retrieval and sentence selection, respectively, was also left as is to be 5. Any claim which had the label of `NOT ENOUGH INFO` was removed from the Oracle setting.

2. **Evidence aggregation:** As part of the competition-provided IR system we next aggregate the top 10 documents and combine the evidence sentences into a single document.

3. **Classification:** Finally, we compare the claim to the evidence document to classify it as either `SUPPORTS` or `REFUTES`. For our learning framework, we employ a support vector machine (SVM) (Hearst et al., 1998) with a linear kernel.

2.1 Features

In this section we describe the various groups of features that were used for the classification task in the last component of the approach introduced above. To create these features, the claim and evidence were tokenized and parsed using CoreNLP (Manning et al., 2014). The majority of the features are either proportions or counts so as to maintain the unlexicalized aspect. Specific lexical content was used only when the semantics were domain independent (i.e., as with certain discourse markers such as *however*, *possible*, *not*, etc).

- **Word overlap:** This set of features was based on the proportion of words that overlap between the claim and the evidence. Specifically, given a claim and a body of evidence, c and e, we compute the proportion of words in $c \cup e$ that overlap: $\frac{|c \cap e|}{|c \cup e|}$. We made similar features for verb and noun overlap as well, where we also include two sub features for the proportion of words in c and also the proportion of words in e: $\frac{|c \cap e|}{|c|}$ and $\frac{|c \cap e|}{|e|}$.

  For all these features, we used the lemma form of the words and first removed stop words (see Appendix A for the list of stop words that were removed). In all, there were 5 features in this feature set, two each for noun and verb overlap as defined above and one for word overlap.

- **Hedging words:** Hedging is the process of using cautious or vague language to vary the strengths of the argument in a given sentence. When present, it can indicate that the author is trying to circumvent facts. To capture this, we have a set of indicator features that mark the presence of any word from a given list of hedging words (see Appendix A for the list of hedging words used) in either the claim or evidence sentences This feature set has a total of 60 hedging features. While these features are lexicalized, their semantics are domain-independent and therefore in scope of our approach.

- **Refuting words and negations:** When present, refuting words can indicate that the author is unequivocally disputing a claim. To capture this, as with the hedging words above, we include a set of indicator features for refuting words (see Appendix A for the complete list of refuting words used) that are present in either the claim or evidence sentences. Also as with the hedging features, the semantics of these words are expected to be consistent across domains. This feature is a one hot vector denoting the existence (or absence) of any of the aforementioned 19 refuting words, creating a feature vector of length 19.

  Another signal of disagreement between two texts is the presence of a verb in one text which is negated in the other text, largely regardless of the identity of the verb (e.g.,

Barack Obama was *not born* in the United States). To capture this, features were created to indicate whether a verb in the claim sentence was negated in the evidence and vice versa. This polarity indicator, created through dependency parsing, thus contained 4 features, each indicating tuples (positive claim-negative evidence, negative claim-positive evidence, etc.)

- **Antonyms:** Presence of antonyms in evidence sentences may indicate contradictions with the claim (e.g.: The movie was *impressive* vs the movie was *dreadful*). This feature captures the number of nouns or adjectives in the evidence sentence that were antonyms of any noun or adjective in the claim sentence (and vice versa). Similar to the word overlap feature mentioned above, every such antonym feature has two sub features, each denoting the proportion over antonyms in claim and evidence, respectively. Thus, there are a total of 4 antonym features. The list of antonyms used were extracted from Word Net (Miller, 1995).

- **Numerical overlap:** Human authors of fake articles often exaggerate facts (e.g., claiming *Around 100 people were killed as part of the riots*, when the evidence shows a lower number). To approximately measure this, we find the intersection and difference of numerical values between claim and the evidence, making it 2 features.

- **Measures of lexical similarity:** While the use of specific lexical items or their corresponding word embeddings goes against the *unlexicalized*, domain-independent aim of this work, here we use relative distributional similarities between the texts as features. Particularly, the relative position of the words in an embedding space carries significant information for recognizing entailment (Parikh et al., 2016). To make use of this, we find the maximum, minimum, and average pairwise cosine similarities between each of the words in the claim and the evidence. We additionally include the overall similarity between the two texts, using a bag-of-words average to get a single representation for each text. We used the Glove (Pennington et al., 2014) embeddings for these features.

Model	Evidence F1	Label Accuracy	FEVER score
Baseline	0.1826	0.4884	0.2745
Our model	0.1826	0.3694	0.1900

Table 1: Performance of our submitted model on the test data.

Model	Label Accuracy
Baseline (Thorne et al., 2018)	65.13
Our model at submission	55.60
Our model post submission	56.88

Table 2: Oracle classification on claims in the development set using gold sentences as evidence

3 Experiments

3.1 Data and Tuning

We used the data from the FEVER challenge (Thorne et al., 2018), training on the 145,449 claims provided in the training set and tuning on the provided development set (19,998 claims). Since we were focusing only on the textual entailment part, we removed the claims which had the label `NOT ENOUGH INFO` during training. As a result, we trained on the remaining 109,810 claims in the training set and tuned on the remaining 13,332 in the development set.

3.2 Baseline

We compare against the provided FEVER baseline. The IR component of the baseline is identical to ours as we reuse their component, but for the textual entailment they use the fully lexicalized state of the art decomposable attention model of (Parikh et al., 2016).

4 Results

The performance of our submitted domain-independent model on the test data (using the baseline IR system) can be found in Table 1, along with the performance of the fully lexicalized baseline. The current performance of our model is below that of the baseline, presumably due to the lack of domain-specific lexicalized features.

Since here we focus on the RTE component only, we also provide the model's performance in an oracle setting on the development set, where we use the gold sentences as evidence in Table 2. Included in the table are the results both at the time of submission and post-submission. At the time of submission, the model included only the word overlap, negated and refuting words, hedg-

Feature group removed	Accuracy
With all features	56.89 %
– Word overlap	50.89 %
– Hedging words	50.96 %
– Antonyms	52.17 %
– Measures of lexical similarity	55.60 %
– Refuting words and negations	55.82 %

Table 3: Ablation results: performance of our model on development after removing each feature group, one at a time. Performance is given in the oracle setting, using the gold sentences as evidence.

ing words and antonym features. Post-submission, we added the lexical similarity features.

Making use of the relative interpretability of our feature-based approach, we performed an ablation test on the development data (again, in the oracle setting using gold sentences as evidence) to test the contribution of each feature group. The results are shown in Table 3 . The word-overlap and hedging features had the largest contribution. The relatively small contribution of the refuting words and negation features, on the other hand, could be due to the limited word set or the lack of explicit refuting in the evidence documents.

5 Analysis

To find the importance of each of the features as assigned by the classifier we printed the weights for the top five features for each class, shown in Table 4. As can be seen in this table, the feature that was given the highest weight for the class `REFUTES` is the polarity feature that indicates a conflict in the polarity of the claim and evidence (as determined by finding a verb which occurs in both, but is negated in the claim and not negated in the evidence). The feature with the second highest weight for the `REFUTES` class is the proportion of nouns that were common between claim and evidence. Another feature that the classifier has given a high importance for belonging to this class, is the count of numbers that were present in the claim but not in evidence (numbers are defined as tokens having *CD* as their part of speech tag).

Similarly, the feature which had the highest weights for the class `SUPPORTS` is that of the word overlap (which denotes the proportion of unique words that appear both in claim and evidence). Notably, the existence of some of the hedging words were found to be indicative of the `REFUTES` class (e.g., *question* and *predict*) while others were indicative of the `SUPPORTS` class (*argue*, *hint*, *prediction* and *suggest*).

While most of the weights as generated by the classifier are intuitive, these features are clearly insufficient, as demonstrated by the low accuracy of the classifier. To address this we manually analyzed 30 data points from the development data set that were wrongly classified by our model. A particular focus was to try to understand and trace back which features contributed (or did not contribute) to the `SUPPORTS` and `REFUTES` classes.

Several of the data points demonstrated ways in which straightforward extensions of the approach (i.e., additional features) could help. For example consider this data point below, which belongs to the class `REFUTES` but was classified to be in the class `SUPPORTS` by our model:

Claim:*Vedam was written and directed by Christopher Nolan .*
Evidence: *Vedam is a 2010 Telugu language Indian drama film written and directed by Radhakrishna Jagarlamudi....*

We conjecture that this error occurred due to the lack of syntactic information in our system. For example, a simple extension to our approach that could address this example would look for similar (and dissimilar) syntactic dependencies between the claim and evidence.

On the other hand, a few of the data points contained more complex phenomenon that would be difficult to capture in the current approach. Consider the following example which belongs to the class `REFUTES` but was wrongly classified as `SUPPORTS` by our model:

Claim:*Sean Penn is only ever a stage actor.*
Evidence:*Following his film debut in the drama Taps and a diverse range of film roles in the 1980s, ... Penn garnered critical attention for his roles in the crime dramas At Close Range, State of Grace, and Carlito's Way .*

This example shows the difficulty involved in capturing the underlying complexities of words that indirectly capture negation such as *only*, which our features do not capture presently.

Lastly, we found that certain aspects of our approach, even with minimal dependence on lexicalization, are still not as domain-independent as desired. Consider the example below, whose gold label is `SUPPORTS`, but was classified as `REFUTES` by our model.

Weight	Feature Name	Description
1.30	polarity_neg_claim_pos_ev	Presence of verb negated in the claim but not in the evidence
0.537	noun_overlap	Proportion of nouns in claim and evidence that overlap
0.518	hedging_evidence_question	Presence of the hedging word *question* in the evidence
0.455	num_overlap_diff	Count of numbers present in claim but not in the evidence
0.385	hedging_claim_predict	Presence of the hedging word *predict* in the claim
-0.454	hedging_evidence_suggest	Presence of the hedging word *suggest* in the evidence
-0.477	hedging_evidence_prediction	Presence of the hedging word *prediction* in the evidence
-0.584	hedging_claim_hint	Presence of the hedging word *hint* in the claim
-0.585	hedging_claim_argue	Presence of the hedging word *argue* in the claim
-1.59	word_overlap	Proportion of words in the claim and evidence that overlap

Table 4: Top five features with the highest weight in each class, where the positive class is `REFUTES` and the negative class is `SUPPORTS`.

Claim: *The Gettysburg Address is a speech.*
Evidence: *Despite the speech's prominent place in the history and popular culture of the United States, the exact wording and location of the speech are disputed.*

We believe this error occurred because we have more argumentative features (for example, in this case the presence of the word *despite*), and fewer features to capture the type of *neutral* sentences common in data sources like Wikipedia pages, which have more informative, objective content. On the other hand, fake news articles contain more subjective language, for which argumentative features are well-suited.

Keeping all these errors in mind our future goal is to enhance the performance of the system by adding more potent unlexicalized/domain independent features, including features that take advantage of dependency syntax and discourse information. Also another possibility we would like to explore is replacing the current classifier with other non-linear classifiers including a simple feed-forward neural network. Through these steps, we hope to improve the accuracy of the classifier predictions, pushing the performance closer to that of a fully lexicalized systems, and yet able to transfer between domains.

6 Conclusion

Despite our current low performance in the FEVER challenge, we would like to propose this system as a precursor to an effort towards building a cross-domain fake data detection model, especially considering its basic implementation. The added benefit for our simple system, when compared to other complicated neural network/deep learning architectures (which are harder to interpret), is that this also provides an opportunity to peek into the what features contribute (or do not contribute) to the development of such a cross-domain system.

Acknowledgements

We would like to thank Ajay Nagesh and Marco A. Valenzuela-Escárcega for their timely help.

References

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Steven Bird and Edward Loper. 2004. Nltk: the natural language toolkit. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, page 31. Association for Computational Linguistics.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*.

Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. 1998. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Dean Pomerleau and Delip Rao. 2017. Fake news challenge.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*.

William Yang Wang. 2017. " liar, liar pants on fire": A new benchmark dataset for fake news detection. *arXiv preprint arXiv:1705.00648*.

A Supplemental Material

A.1 Hedging words

allegedly, argument, belief, believe, conjecture, consider, hint, hypotheses, hypothesis, hypothesize, implication, imply, indicate, predict, prediction, previous, previously, proposal, propose, question, reportedly, speculate, speculation, suggest, suspect, theorize, theory, think, whether

A.2 Stop words

We used a subset of the stop words (and partial words) that come from the python Natural Language Toolkit (NLTK) (Bird et al., 2009):

a, about, ain, all, am, an, and, any, are, aren, aren't, as, at, be, been, being, by, can, couldn, couldn't, did, did n't, didn, do, does, doesn, doesn't, doing, don't, few, for, from, further, had, hadn, hadn't, has, hasn, hasn't, have, haven, haven't, having, he, her, here, hers, herself, him, himself, his, how, i, if, in, into, is, isn, isn't, it, it's, its, itself, just, ll, me, mightn, mightn't, more, most, mustn, mustn't, my, myself, needn, needn't, nor, of, on, or, our, ours, ourselves, own, shan, shan't, she, she's, should, should've, shouldn, shouldn't, so, some, such, than, that, that'll, the, their, theirs, them, themselves, then, there, these, they, this, those, through, to, too, until, ve, very, was, wasn, wasn't, we, were, weren, weren't, what, when, where, which, while, who, whom, why, will, with, won't, wouldn, wouldn't, y, you, you'd, you'll, you're, you've, your, yours, yourself, yourselves

A.3 Refuting words

bogus, debunk, denies, deny, despite, doubt, doubts, fake, false, fraud, hoax, neither, no, nope, nor, not, pranks, refute, retract