

# Who Blames Whom in a Crisis? Detecting Blame Ties from News Articles Using A Neural Network

Anonymous CoNLL submission

## Abstract

Blame games tend to follow major disruptions, be they financial crises, natural disasters or terrorist attacks. To study how the blame game evolves and shapes dominant crises narratives is of great significance, as sense-making processes can affect regulatory outcomes, social hierarchies and cultural norms. However, it takes tremendous time and effort for social scientists to manually examine each relevant news article and extract the blame ties (A blames B). In this study, we define a new task, named Blame Tie Extraction, and construct a new dataset related to the United States financial crisis (2007-2010) from *The New York Times*, *The Wall Street Journal* and *USA Today*. Using an end-to-end approach, we build a bi-directional Long Short-Term Memory (LSTM) network on contexts where the entities appear and learn to automatically extract such blame ties on document level. Leveraging on the large unsupervised model such as GloVe and ELMo, our best model achieves F score of 76% and 70% on development set and test set respectively for blame tie extraction, making it a useful tool for social scientists to extract blame ties more efficiently.

## 1 Introduction

Blame is an issue that has been receiving increasing attention in the social sciences in recent years (Alicke, 2000; Douglas, 1992; Farmer, 2006; Gephart, 1993; Hobolt and Tilley, 2014; Hood, 2010; Shaver, 2012; Tilly, 2009). In particular, more attention has been placed on blame dynamics following major disruptions, such as natural disasters (Boin et al., 2010; Malhotra and Kuo, 2008), financial crises (Nicol, 2016; Tourish and Hargie, 2012), or terrorist attacks (Olmeda, 2008). Studying blame is of great significance as sense-making processes inform what and who a society values, and ultimately shape lawmaking. For

1). **John B. Taylor**, a professor of economics at Stanford University and a former under secretary of the Treasury under President Bush, recently argued that the **Fed's** rate cuts after 2001 appeared to have *exaggerated both the housing boom and bust*.

2). **Homeowners** are *suing mortgage lenders*.

3). On Monday, **Mr. Schumer** invited **Angelo R. Mozilo**, chief executive of Countrywide Financial, *to testify*. It was unclear whether Mr. Mozilo would agree to appear.

Figure 1: Three sentences containing blame ties from our dataset. The red/**bold** words are entities involved in a blame tie, and the blue/*italic* words are supporting evidence that the blame tie exists.

instance, the intense blame targeting Wall Street during the financial crisis (2007-2010) helped lawmakers pass the July 2011 Dodd-Frank Wall Street Reform and Consumer Protection Act.

Important though it is, it takes tremendous time and effort for social scientists to manually examine each relevant news article and extract the blame ties (A blames B). Figure 1 shows sentences containing blame ties. Recently, the rapid development in Natural Language Processing shows how deep neural networks can be a powerful tool to solve social science problems (Rule et al., 2015; Bail, 2016). Based on the dataset annotated from several media excerpts focused on blame, we investigate automatic ways to extract blame ties from new articles.

There are challenges though. First, some patterns only have blame meanings in specific contexts. For instance, “*It was lenders that made the lenient loans, it was home buyers who sought out easy mortgages, and it was Wall Street underwriters that turned them into securities.*” (*The Wall Street Journal*, Aug 2007), only with the background of financial crisis can we identify the blame targets. Second, there are many ways to attribute blame and the structure of the sentences can

be quite complex. Third, it is common for journalists to use metaphors and irony to designate actors.

To address these challenges, we leverage a neural network to learn prior information about entities themselves, and extract features for blame tie extraction. In particular, a neural network is used to learn dense vector representations of entities, so that similar entities can be visualized close to each other in the embedding space, and the likeliness of one entity to blame another can be inferred automatically without further knowledge. In addition, a LSTM neural network is used to represent news articles and their sentences automatically, which can be used to predict blame ties between entities mentioned in the news articles using linguistic clues. Finally, a model that integrates entity knowledge and linguistic knowledge is constructed by integrating the two respective networks.

We conduct a case study on blame games for the U.S. financial crisis (2007-2010), the most important since the Great Depression, which led to at least \$6 trillion in losses (Luttrell et al., 2013). Results show that our model can effectively learn both entity knowledge and linguistic clues for blame tie. For example, it successfully mines out entity relation with regard to blame ties from the crisis, such as the fact that Wall Street and McCain tend to blame the same targets (Obama and Bernanke). In addition, the model can generalize to new cases in extracting blame patterns automatically. Our implementation and trained models are released at <https://github.com/anonymized>. A full demo is also available at <http://anonymized>.

## 2 Related Work

NLP has become an increasingly popular tool in social science. O'Connor et al. [2010] aligned sentiment measured from Twitter to public opinion measured from polls, and found the two correlate well. Bamman and Smith [2015] tried to use text data to estimate the political ideologies of individuals. Preotjuc-Pietro et al. [2017] also predicted political ideologies of twitter users, in a more fine-grained form. Social scientists used NLP along with network analysis etc. to analyze social media texts (Rule et al., 2015) and State of Union addresses in United States (Bail, 2016).

The Blame Tie Extraction Task can be regarded as a special case of relation extraction (Miwa and Bansal, 2016). The differences are two-fold: 1)

Most relation extraction systems work at sentence level, while our system works at document level. 2) In relation extraction, we need to extract both entities and their relations in pipeline (Nadeau and Sekine, 2007; Rink and Harabagiu, 2010) or jointly (Zheng et al., 2017), while in the Blame Tie Extraction Task, we are given the entities we are interested in, besides the article text, and we try to predict the blame ties that exist between these entities.

To the best of our knowledge, our work is the first to study the Blame Tie Extraction Task using NLP techniques.

## 3 Dataset

The second author manually created a dataset on U.S. financial crisis. The dataset was drawn from three newspapers in the U.S., including *The New York Times*, *The Wall Street Journal* and *USA Today*, chosen for three main reasons. First, they are the most widely circulated newspapers in the United States. Second, they cover the social spectrum from elite to mass. Third, they also cover the political spectrum: from the quite liberal *New York Times*, to the centrist *USA Today*, to the conservative *Wall Street Journal* (Gentzkow and Shapiro, 2010; Groseclose and Milyo, 2005). The time period studied here spans from August 2007, after the first warning signs for the crisis appeared, to June 2010, right before the signature of the largest set of financial regulations since the Great Depression.

We use a set of key words to filter the articles gathered from Factiva<sup>1</sup> and LexisNexis<sup>2</sup>, getting articles containing blame patterns for the crisis. The key words we use are blame related (such as *attack*, *accuse*, *misconduct*, ...) or event related (such as *financial crisis*, *global recession*, *housing bubble*, ...) <sup>3</sup>. The two classes of key words are combined together to filter the articles.

Blame incidences are manually coded for each article. The identification of a blame pattern requires the presence of 1) a blame source, 2) a blame target, and 3) a causality link. An example of a sentence containing a blame instance would be the following: “*Sen. Richard Shelby, R-Ala. [Blame Source]...said the FED [Blame*

<sup>1</sup><https://www.dowjones.com/products/factiva/>

<sup>2</sup><https://www.lexisnexis.com>

<sup>3</sup>The full list of keywords are released along with the code.

	USA	NYT	WSJ
days	310	736	648
articles	132	429	438
blame ties	353	787	754

Table 1: Dataset size for the three newspapers. USA: *USA Today*. NYT: *The New York Times*. WSJ: *The Wall Street Journal*.

number of articles	998
number of samples	8562
number of entities/article	2.97
average neg/pos ratio per article	2.19
total neg/pos ratio	3.61

Table 2: Samples statistics.

*Target*] ‘kept interest rates too low for too long, encouraging a housing bubble and excessive risk taking’[causality link].” (*USA Today*, December 2009). The blame source and target form a blame tie. The statistics of the dataset with the number of blame ties are shown in Table 1.

In the training process, the extracted blame ties serve as positive samples. The negative samples are generated by considering all possible permutations of the entities involved in any blame ties, and then removing the positive entity pairs. The samples statistics about the whole dataset are shown in Table 2. Theoretically, the number of negative samples will increase quadratically with the number of entities in the article. In our dataset, the entities we considered per article are on average 3. Therefore the dataset is not severely unbalanced.

An article and its annotations are shown in Table 3. The article talks about the reactions and attitudes of a real-estate company, a health-care company and general Americans (mainly readers of *The Wall Street Journal*) towards an incoming Federal Reserve meeting on tax cut. 6 entities are mentioned and 4 blame ties are extracted from this article.

## 4 Task

We formulate the Blame Tie Extraction task as follows. Given a news article  $d$  and a set of entities  $e$ , we have  $|e| \cdot (|e| - 1)$  possible directed links among them. We assign label 1 to a pair  $(s, t)$  when entity  $s$  blames entity  $t$  based on article  $d$ , otherwise we assign label 0 to the pair. We can use a matrix for a more intuitive illustration. For the example in Table 3, the matrix constructed is shown in Table 4.

For a given entity pair  $(s, t)$  with label  $l$ , we would like to maximize the likelihood

$$L = P(l|s, t, d)$$

In order to predict whether a blame tie exists between two entities based on the article, we have two sources of information to resort to. One is the entities themselves. For instance, we know that Democrats tend to blame Republicans so as to weaken their political opponents, and tend to blame Wall Street so as to gain popular support to impose a stringent set of financial regulation. The other one is the contexts in which the entities are mentioned. We rely on linguistic patterns of sentences to extract the blame ties. For instance, in this sentence “*Who is to blame? Hedge funds, for one, he says.*” (*The Wall Street Journal*, Sept 2007), the linguistic structure indicates that the entity appearing after the question is the blame target entity, and the narrator is the blame source entity.

## 5 Models

We first introduce the Entity Prior Model which directly extracts prior information about entities (i.e. which entities tend to blame which entities). The Entity Prior Model can learn entity information from its blame patterns, such as political standing etc., and can generalize to new events with same entities. Then, we mask out the entities mentions from text, relying purely on the contexts surrounding the mentions of the entities. The Context Model can generalize across different entities and across historical periods. Finally, a model combining the two is built to investigate the interactions.

### 5.1 Entity Prior Model

We use the fully connected feedforward neural network (FCN) to collect entity prior knowledge, namely who is likely to blame whom without further information. To this end, we represent entities by their embeddings, concatenate the embeddings of the blame source and target entity, and then stack a fully connected layer to learn the interactions between the entities. The FCN outputs the probability of whether the blame tie from the source entity to the target entity exists:

$$f_{score} = ([\mathbf{E}^e(e_s); \mathbf{E}^e(e_t)]) \cdot W^e + b^e$$

$e_s$  and  $e_t$  represent the source entity and target entity index, respectively,  $\mathbf{E}^e$  is the embedding

Articles		
... Ordinarily, <b>Americans</b> welcome lower interest rates. But many feel differently this time. Some think the economy is fine and inflation is the main danger. But a moral element is also at work: Many think a rate cut would reward foolish speculation and <b>Wall Street</b> greed at the expense of the thrifty (1). “The Federal Reserve needs to stand its ground and not bail out <b>hedge funds</b> – they should have known better to begin with!” <b>Suzanne Mitchell</b> , an administrative assistant at a Houston real-estate company, says in an email (2). In an interview, she adds: “I’m very sorry that <b>people</b> took out \$450,000 mortgages with no money down ... <b>people</b> ought to be responsible for the loans they take out.” (3) ... But Mr. <b>Brason</b> contrasts that with the far greater reliance on borrowed money that is typical nowadays. Some “of us ... weren’t buying up five or 10 properties without any money down,” he says. “ <b>People</b> took the risks and should pay the price. (4) A lot of others at the higher end of the food chain, the investment bankers and hedge-fund managers were making oodles of fee-income money and frankly, there’s a lot of public opinion that it was excessive.” ...		
Blame Source	Blame Target	Causality Link
Americans( $e_1$ )	Wall Street( $e_2$ )	(1)
Suzanne Mitchell( $e_3$ )	Hedge funds( $e_4$ )	(2)
Suzanne Mitchell( $e_3$ )	people( $e_5$ )	(3)
Brason( $e_6$ )	people( $e_5$ )	(4)

Table 3: An article titled *Rate Cut Has Foes on Main Street* (*The Wall Street Journal*, September 2007). Top: paragraphs of the article containing blame patterns. The blame entities are in **bold face**. Bottom: blame ties extracted from the article.

target source	$e_1$	$e_2$	$e_3$	$e_4$	$e_5$	$e_6$
$e_1$	-	1	0	0	0	0
$e_2$	0	-	0	0	0	0
$e_3$	0	0	-	1	1	0
$e_4$	0	0	0	-	0	0
$e_5$	0	0	0	0	-	0
$e_6$	0	0	0	0	1	-

Table 4: Matrix representation of the blame ties in Table 3.

	min	avg	max
sentences per doc	4	45	384
words per sentence	1	26	159
words per doc	133	1,209	9,064

Table 5: Sentence and words statistics.

matrix for entities, ‘;’ is the concatenation operator, and  $W^e$  and  $b^e$  are parameters. Specifically,  $\mathbf{E}^e \in \mathbf{R}^{n \times m}$ ,  $n = 707$  is the number of entities in training set,  $m$  is the entity embedding dimension, which is tuned as hyperparameters.  $W^e \in \mathbf{R}^{2m \times 2}$  and  $b^e \in \mathbf{R}^2$ .  $\mathbf{E}^e$  is initialized with standard normal distribution.

Like word embedding, we hope to learn meaningful representations of the entities, i.e., the entities sharing similarity blame behavior patterns will stay close in the embedding space.

## 5.2 Context Model

The Entity Prior Model tells how likely a particular entity is to blame or be blamed without further information. In contrast, the Context Model relies only on linguistic clues from news articles, finding

blame patterns explicitly or implicitly mentioned. The Context Model is thus useful across different political settings with different entities.

To model context information about an entity, we first use string matching to locate the positions of all occurrences of the entity in article. A position is represented by a tuple  $(s\_no, w\_no)$ , where  $s\_no$  and  $w\_no$  denotes sentence number and word number, and the positions of blame source and blame target are denoted as  $pos^s$  and  $pos^t$ , respectively. Second, we replace each entity mention by a special  $\langle \text{ENTITY} \rangle$  token, so that we do not have any information regarding to the entity itself. Third, we run a bidirectional LSTM on sentences containing the entity, and use the LSTM output to represent the context of each word:

$$h_j^i = \text{LSTM}(\mathbf{E}^w(w_j^i))$$

$w_j^i$  is the  $j$ -th word of the  $i$ -th sentence,  $\mathbf{E}^w$  is the embedding matrix for words, and  $h_j^i$  is the concatenation of the LSTM outputs of the last layers from both directions. If  $(i, j) \in pos^s$  or  $(i, j) \in pos^t$ ,  $h_j^i$  will be used to represent the context of entity  $s$  or  $t$ .

Since an entity may appear at multiple positions in one article, we may have multiple representations of the entity context. Pooling is used to reduce the embeddings into one single vector. The pooling result of the contexts representations of source and target entity are denoted as  $V_s$  and  $V_t$ , respectively:

$$V_e = \text{pool}_{(i,j) \in pos^e} (h_j^i), e \in \{s, t\}$$

where pool denotes the pooling function, for



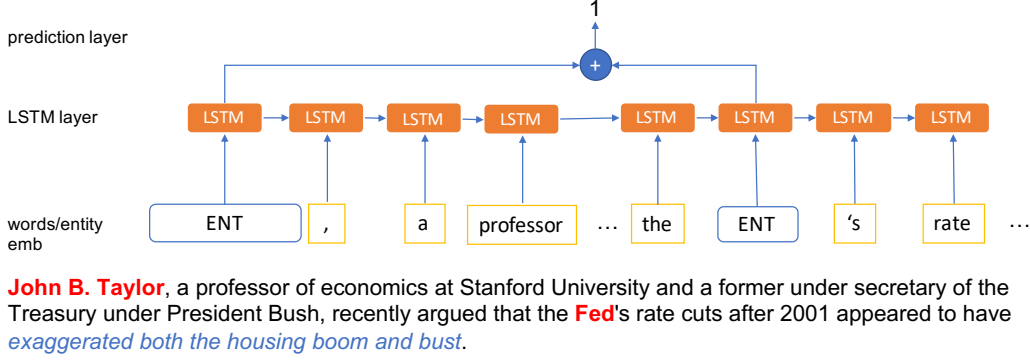


Figure 2: Context Model. LSTM is used to encode the sentences, and the hidden vectors at the positions of the entity are pooled together into a single vector to represent the context of the entity. The source entity context vector and the target entity context vector are concatenated together to be sent to the prediction layer. The prediction layer predicts 1 if the source to target blame tie exists otherwise 0.

which we try random selection, mean, max and attention. For attention pooling, we use a two layer feedforward neural network to learn a score for each vector representations, use softmax function to normalize it as weights, and sum the vectors into a single vector. Once we obtain both entities' representations, we concatenate the two vectors together, and then use a fully connected layer to learn a score of this entity pair:

$$f_{score} = ([V_s; V_t]) \cdot W^v + b^v$$

$W^v \in \mathbf{R}^{2H \times 2}$  and  $b^v \in \mathbf{R}^2$  are both parameters.  $H$  is the dimension of BiLSTM outputs. The model architecture is depicted in Figure 2.

### 5.3 Combined Model

To incorporate the prior information about entities, we concatenate two additional vectors on the basis of the Context Model, which are the embeddings of the blame source entity  $\mathbf{E}^e(s)$  and blame target entity  $\mathbf{E}^e(t)$  from section 5.1:

$$f_{score} = ([\mathbf{E}^e(e_s); V_s; \mathbf{E}^e(e_t); V_t]) \cdot W^c + b^c$$

$W^c \in \mathbf{R}^{2H+2m}$  and  $b^c \in \mathbf{R}^2$  are both parameters. We expect the model to simultaneously learn how to extract blame ties from context, and also learn the representations of the entities themselves as a byproduct.

## 6 Experiments

We conduct experiments on our dataset using the Entity Prior Model, the Context Model and the Combined Model, and compare the performance among different models. To find the best model

architecture, we also conduct development experiments to investigate the effect of different pooling functions and pretrained word embeddings.

### 6.1 Experimental Settings

Stanford CoreNLP (Manning et al., 2014) is used to tokenize articles into sentences and words. The sentence length statistics of the dataset articles are shown in Table 5. There is no further preprocessing except that words are converted to lower case. Dataset is split into train, dev and test set in the article level by the ratio of 8:1:1. We use F1 on positive class to measure the performance of the model.

To investigate whether the learned entities representations help in blame tie extraction, we do another round of evaluations on the known entities of dev and test set as shown in Table 8. KNOWN denotes entity pairs both of which appear in the training data, while ALL denotes all entity pairs. By comparing the model performance on KNOWN with that of ALL, we can evaluate the usefulness of entities representations. Conversely, we can also evaluate the robustness of the model against unknown entities. Since not all entities in dev and test set appear in the training set, a special  $\langle \text{UNK\_ENTITY} \rangle$  symbol is used to represent the entities which are not present in the training set.

Leveraging on large unsupervised data is proved to be helpful on many NLP tasks, especially on tasks with small dataset. Word2Vec (Mikolov et al., 2013) or GloVe (Pennington et al., 2014) can be used to pretrain word embeddings on larger external dataset. ELMo (Peters et al., 2018) word vectors are internal states of a deep bidirectional language models, and can effectively captures the

Model	dev F1(%)
LSTMRand	50.51
LSTMMean	50.15
LSTMMax	51.49
LSTMAtn	50.17
BiLSTMRand	56.43
BiLSTMMean	61.95
BiLSTMMax	62.26
BiLSTMAtn	61.92

Table 6: Experiment results of Context Model using different pooling functions.

syntax and semantics of words. We conduct experiments on GloVe and ELMo, and investigate the effects of these pretrained word embeddings.

Our model is implemented in PyTorch<sup>4</sup>. For the training hyperparameters, Adam (Kingma and Ba, 2014) is used as the optimizer, and the default learning rate is adopted. We use the minibatch size of 50 for all three models. Dropout (Hinton et al., 2012) of 0.5 and weight decay of  $1e-8$  are used to prevent overfitting. Dropout is used on word embeddings and RNN outputs. We set gradient clipping to 3 to stabilize training. Maximum number of epochs is 30 and if the performance does not improve on dev set for 10 epochs it is terminated. For the model hyperparameters, word embedding size is 100, LSTM hidden size is 100 for each direction, and entity embedding size is 50. A single layer of LSTM is enough, since the dataset is small.

## 6.2 Development Experiments

As stated in Section 5.2 for the BiLSTM model, we compare the model performance when different pooling method is used to aggregate different contexts representations, and we also compare the bidirection LSTM with a single forward LSTM. The results on ALL dataset are shown in Table 6. BiLSTM works better than LSTM for every pooling method, since we can take advantage of information from both directions of the sentence. Random pooling has the worst result, mainly because a lot of important information is lost. Max pooling has the highest score on most cases, therefore we use BiLSTM + max pooling in the following experiments.

To investigate effects of pretrained word embeddings, we initialize the word embedding pa-

<sup>4</sup><http://pytorch.org>

Model	dev F1(%)	test F1(%)
random	62.26	56.11
GloVe fixed	63.09	62.10
GloVe tuned	61.37	57.75
ELMo	<b>73.16</b>	<b>66.35</b>

Table 7: Experiment results of Context Model using different pretrained word vectors.

Model	KNOWN		ALL	
	dev F1(%)	test F1(%)	dev F1(%)	test F1(%)
entity	73.97	<b>70.97</b>	61.07	60.06
context	74.29	63.11	73.16	66.35
combined	<b>81.75</b>	68.67	<b>76.13</b>	<b>69.92</b>

Table 8: Experiment results of Entity Model, Context Model and Combined Model on KNOWN data and ALL data.

rameters with random initialization, GloVe pretrained word embeddings and ELMo models. Since the official release of pretrained ELMo model has the output dimension of 1024, we do a linear transformation to reduce the output dimension to 100. The transformation matrix are part of the model parameters and are tuned during training. For GloVe we use fixed version and tuned version. For ELMo model, since it slows down the training significantly, we do not tune the model parameters. The results using BiLSTM+Max pooling are shown in Table 7. Fixed GloVe vectors improves the dev and test F1 by 0.83% and 4.99%, respectively. Tuned version of GloVe does not improve that much, maybe due to the reason that the dataset is small and too many parameters will overfit the model. Pretrained ELMo model improves dev and test F1 by 10.90% and 10.24% compared with random initialization, proving the powerfulness of ELMo model.

## 6.3 Results

Table 8 exposes the final result of the Entity Model, Context Model and Combined Model. For the Context Model, we use the BiLSTM+Max pooling, and use ELMo model to get words representations.

For the Entity Model, the performance on KNOWN entities is better than that on ALL entities. This result illustrates that the model learns prior information about the entities in the train set. From a visualization of entity embeddings using tSNE (Maaten and Hinton, 2008), we found that

	CTX Wrong	CTX Correct
<b>CMB Wrong</b>	0/0	31/19 (II)
<b>CMB Correct</b>	98/127 (I)	413/406

Table 9: Model comparison between Context Model (CTX) and Combined Model (CMB) on dev/test sets.

entities with similar political backgrounds tend to be close to each other in the embedding vector space. For example, Wall Street and McCain are close to each other, which is intuitive since they were both blamed by Obama, and they both blamed Obama and Bernanke, according to the training data. In addition, Obama, Congress and Bernanke also form a cluster, since they were all blamed by McCain, and blamed the Fed. Part of the visualization is shown in Figure 3.

For the Context Model, we can see that it can effectively extract blame ties from news articles without prior entity knowledge. For known entities, the F1 is 63.11%; for all entities in the test data, the F1 does not decrease like Entity Prior Model, it even increases to 66.35%. Unlike knowledge about entities, linguistic knowledge generalizes robustly to unseen test data, where most entities do not exist in the training data. This implies that our model can be used to extract blame ties in other occasions where the political settings are highly different. For example, when the President of the United States changes, we can still use our model to predict how the new President will play the blame game.

For the KNOWN entities, the Combined Model does not perform better than Entity Model. This shows that entities information alone may be useful in extracting blame ties than using contexts. However, such information could not be used when the entities are new, for instance, our ALL data. On ALL data, our Combined Model achieves best result on dev and test set, showing that the model can integrate context information as well as the entity prior information to make better predictions.

Therefore, the Context Model is the most useful one, applicable even when we would like to use the model to extract blame tie among new entities. Entity prior information is helpful. If available, it can be used to boost the model performance.

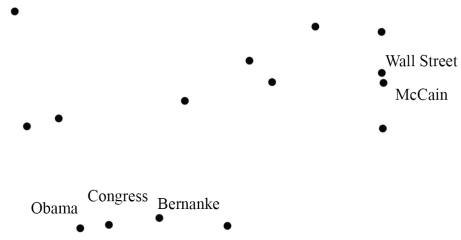


Figure 3: Entity embedding visualizations (only a subset of the entities are shown for legibility). Wall Street and McCain are close. Obama, Congress and Bernanke also form a group.

## 7 Analysis

To verify our hypothesis about why the Combined Model is better than the Context Model, we take several examples on which the Combined Model succeeds while the Context Model fails, and vice versa. Then we use our trained model to extract blame ties from several recent news articles, to evaluate the practicality and generalization of the model.

### 7.1 Case Study

To analyze fine-grained differences between behaviors of the Context Model and the Combined Model, we evaluate the two models on samples of the test data, and divide the samples into four classes, as shown in Table 9.

We find several class I samples (i.e. samples for which the Context Model fails while the Combined Model works correctly) for analysis. In one example, we want to figure out whether *Obama* blames *Republicans* based on the article titled *Obama Issues Sharp Call for Reforms on Wall Street* (*The New York Times*, April 2010). From sentences such as “*The president and his allies have eagerly portrayed Republicans as handmaidens of Wall Street... Obama avoided incendiary language attacking Republicans...*” we can see that the blame tie holds. The Context Model predicts that the blame tie exists with a confidence score of 0.45, while the Combined Model gives a score of 0.97. The reason may be that the Combined Model learned prior information about Obama and Republicans: Obama blamed Republicans before in the training data, which makes sense since he is a Democrat.

For class II samples, the Combined Model fails, while the Context Model works correctly. In the *USA Today* article *Obama tells Wall Street to join in* (April 2010), the context is “*...The president said the financial crisis, which has cost more*

Articles	
<b>Trump</b> Accuses <b>Russia</b> of Helping <b>North Korea</b> Evade Sanctions (1)	
President <b>Donald Trump</b> accused <b>Russia</b> in unusually harsh terms of helping <b>North Korea</b> evade United Nations sanctions intended to press the country to give up its nuclear and ballistic missile programs. (2)	
“ <b>Russia</b> is not helping us at all with <b>North Korea</b> ,” <b>Trump</b> said in an interview with Reuters on Wednesday (3). “What China is helping us with, <b>Russia</b> is denting. In other words, <b>Russia</b> is making up for some of what China is doing.”	
<b>Trump</b> has leaned on China to curb its support for <b>North Korean</b> leader Kim Jong Un’s regime, and in exchange has so far laid off the punishing trade measures he promised against the U.S.’s largest creditor during his campaign.	
North Korea’s weapons programs are <b>Trump</b> ’s most urgent foreign crisis (4). He has vowed not to allow the country to develop a missile capable of carrying a nuclear warhead to the U.S. mainland, threatening war to prevent it if necessary. But Kim has plunged ahead, and his government made rapid advances with both its missile and nuclear technology after <b>Trump</b> took office.	
<b>Trump</b> ’s criticism of <b>Russia</b> (5) is striking because members of Congress have said in the past that he was too reluctant to criticize <b>Russia</b> ’s foreign policy and too eager to establish good relations with President Vladimir Putin.	
Major Entities Mentioned	
Trump ( $e_1$ ), Russia ( $e_2$ ), North Korea ( $e_3$ )	

Table 10: An article from *Bloomberg* published on January 18, 2018. Top: paragraphs of the article containing blame patterns. The blame entities are in **bold face**. Bottom: major entities appearing in the article.

source \ target			
	$e_1$	$e_2$	$e_3$
$e_1$	-	0.82	0.91
$e_2$	0.02	-	0.40
$e_3$	0.00	0.01	-

Table 11: Prediction result of the article in Table 10 using the Context Model. The number represents the probability that the blame ties exists between the two corresponding entities.

than 8 million jobs so far, was ‘born of a failure of responsibility from Wall Street all the way to Washington.’...”. The Context Model predicts that Washington does not blame Wall Street, at the confidence level of 0.62, which is true. However, the Combined Model mistakenly predicts a probability of 0.88 that the blame exists. This is because the Combined Model overly relies on entity information. Nevertheless, from Table 9 this class of samples are much less than the number of class I samples.

## 7.2 Generalization to New Cases

To validate the generalization of our model, we conduct further analysis on news articles beyond the time frame of the financial crisis. In particular, we manually annotate 13 recent articles containing 14 blame ties from Google News, mostly on January 2018, and use our pretrained Context Model to extract blame ties from the articles. The F1 on individual blame ties is 72.00% on this new test data, and 8 out of 13 articles are labeled correctly. The result is consistent with the result of our financial crisis test set. This further demonstrates the generalisability of our Context Model. The articles on which the model fails mainly contain blame patterns that have not be seen in our

financial crisis dataset.

Table 10 shows one of these new articles from Bloomberg Politics<sup>5</sup>. In this news, United States Presidents accused Russia of helping North Korea evade United Nations sanctions. The blame tie is reflected in the sentences (1) (2) (3) and (5). From (4) we can infer that Trump has a negative attitude towards North Korea because of the nuclear weapons. The results of the Context Model are shown in Table 11. The model successfully identifies the blame ties from Trump to Russia and Trump to North Korea, with a high confidence score, while the probabilities for other entity pairs are low.

## 8 Conclusion

We investigated blame analysis, for which previous research looked at the evolution of blame frames, without systematically connecting the frames to the actors who produce them. Experiments show that neural network is effective for identifying blame ties from news articles. Our approach can enable researchers to quantify the importance of a frame, and to understand how and why it became prevalent. It can also enable researchers to study actors’ position alignments over time. We release our code and model for automatic blame tie extraction to facilitate such research.

## References

Mark D Alicke. 2000. Culpable control and the psychology of blame. *Psychological bulletin*, 126(4):556.

<sup>5</sup><https://www.bloomberg.com/news/articles/2018-01-17/trump-accuses-russia-of-helping-north-korea-evade-un-sanctions>



- Christopher Andrew Bail. 2016. Combining natural language processing and network analysis to examine how advocacy organizations stimulate conversation on social media. *Proceedings of the National Academy of Sciences*, 113(42):11823–11828.
- David Bamman and Noah A Smith. 2015. Open extraction of fine-grained political statements. In *EMNLP*, pages 76–85.
- Arjen Boin, Paul‘T Hart, Allan McConnell, and Thomas Preston. 2010. Leadership style, crisis response and blame management: The case of hurricane katrina. *Public Administration*, 88(3):706–723.
- Mary Douglas. 1992. Risk and blame. essays in cultural theory.
- Paul Farmer. 2006. *AIDS and Accusation: Haiti and the Geography of Blame*. Univ of California Press.
- Matthew Gentzkow and Jesse M Shapiro. 2010. What drives media slant? evidence from us daily newspapers. *Econometrica*, 78(1):35–71.
- Robert P Gephart. 1993. The textual approach: Risk and blame in disaster sensemaking. *Academy of Management Journal*, 36(6):1465–1514.
- Tim Groseclose and Jeffrey Milyo. 2005. A measure of media bias. *The Quarterly Journal of Economics*, 120(4):1191–1237.
- Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.
- Sara B Hobolt and James Tilley. 2014. *Blaming europe?: Responsibility without accountability in the european union*. Oxford University Press.
- Christopher Hood. 2010. *The blame game: Spin, bureaucracy, and self-preservation in government*. Princeton University Press.
- Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- David Luttrell, Tyler Atkinson, Harvey Rosenblum, et al. 2013. Assessing the costs and consequences of the 2007–09 financial crisis and its aftermath. *Economic Letter*, 8.
- Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605.
- Neil Malhotra and Alexander G Kuo. 2008. Attributing blame: The public’s response to hurricane katrina. *The Journal of Politics*, 70(1):120–135.
- Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.
- Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. *arXiv preprint arXiv:1601.00770*.
- David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26.
- Olivia Nicol. 2016. No body to kick, no soul to damn: Responsibility and accountability for the financial crisis (2007–2010). *Journal of Business Ethics*, pages 1–14.
- Brendan O’Connor, Ramnath Balasubramanyan, Bryan R Routledge, and Noah A Smith. 2010. From tweets to polls: Linking text sentiment to public opinion time series. *ICWSM*, 11(122-129):1–2.
- José A Olmeda. 2008. A reversal of fortune: blame games and framing contests after the 3/11 terrorist attacks in madrid. by A. Boin, A. McConnell, P. Hart. Cambridge: Cambridge University Press.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.
- Daniel Preotiu-Pietro, Ye Liu, Daniel Hopkins, and Lyle Ungar. 2017. Beyond binary labels: political ideology prediction of twitter users. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 729–740.
- Bryan Rink and Sanda Harabagiu. 2010. Utd: Classifying semantic relations by combining lexical and semantic resources. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 256–259. Association for Computational Linguistics.
- Alix Rule, Jean-Philippe Cointet, and Peter S Bearman. 2015. Lexical shifts, substantive changes, and continuity in state of the union discourse, 1790–2014. *Proceedings of the National Academy of Sciences*, 112(35):10837–10844.

- Kelly G Shaver. 2012. *The attribution of blame: Causality, responsibility, and blameworthiness*. Springer Science & Business Media.
- Charles Tilly. 2009. *Credit and blame*. Princeton University Press.
- Dennis Tourish and Owen Hargie. 2012. Metaphors of failure and the failures of metaphor: A critical study of root metaphors used by bankers in explaining the banking crisis. *Organization Studies*, 33(8):1045–1069.
- Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. 2017. Joint extraction of entities and relations based on a novel tagging scheme. *arXiv preprint arXiv:1706.05075*.

950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999