Is this the real article? Is this just fake? The case for Multi-Task Learning for Fake News Detection

Azamat Omuraliev
University of Amsterdam
azamat.omu@gmail.com

Steven van de Graaf
University of Amstrerdam
steven@vandegraaf.xyz

Abstract

This paper investigates the advantages of a joint learning setup when it comes to Fake News Detection (FND). Specifically, by using Natural Language Inference (NLI) as an auxiliary task, we seek to improve performance on FND as the main task. To this end, we have designed and implemented a Multi-Task Hierarchical Attention Network (MT-HAN), where the word embeddings, as well as the attentative sentence encoder are shared between tasks. Experiments conducted show that the performance on FND is not necessarily improved by jointly training on NLI. Moreover, it shows that, due to our weighting scheme, our models focuses more on fake articles than real articles, which was to be expected and indeed favourable.

1 Introduction

In recent years, fake news has been the subject of much controversy, especially following the 2016 "Brexit" referendum, as well as the United States presidential election that same year (Rose, 2017). Indeed, fake news has become a serious concern, as, reportedly, two-thirds of Americans get their news on social media, predominantly from facebook (Matsa and Shearer, 2018). On top of this, social media have been found to be key conduits for the spread of fake news (Allcott and Gentzkow, 2017).

The definition of fake news used in this paper follows that of (Lazer et al., 2018): *Fake news* is fabricated information, which mimics the form of news media content, but not its intent. Instead. its intents contain both *misinformation* (false or misleading information), as well as *disinformation* (false information spread purposefully to deceive people).

Taking both the importance and sensitivity of the subject of fake news into consideration, as well as the intents and purposes of fake news outlets, it would be an understatement to say that there is a sincere public interest in detecting fake news, that is: Being able to automatically and accurately detect whether a given article is fake or not (real).

Modern researchers consider this problem from several perspectives, as outlined in later sections of this paper. An intuitive approach to detect untruthful content is to compare it against a verified ground truth, but while for a human fact-checker this task is rather trivial, automating this process requires an interplay of information retrieval and language modeling that scales naturally. To this day, there is no unified system that is capable of doing this, and therefore working solely with the textual content of articles provides a less challenging platform for solving this issue.

As a result, neural language models have been on the rise for tackling the problem of fake news detection. The hope is that such models can pick up on the stylistic and (trivial) semantic features that tend to systematically differ between truthful and untruthful documents, and while this assumption can be viewed as rather strong, it has some theoretical foundation. In the late 1960s, the *Undeustsch hypothesis* has been developed, stating that truth and deceptive information tend to be written differently from a stylistic perspective (Undeutsch, 1967).

And while there has already been numerous rather successful attempts to train models that can classify content as fake and real, there are still some challenges that have to be addressed more clearly. One obstacle in working with whole articles rather than short statements lies in computing an accurate representation for a document that captures the semantics and syntax of its underlying components, sentences and words accurately and fully. Another is driven by the fact that to capture the fine-grained differences between fake and

true content, accurate semantic representations of the documents need to be computed.

Another angle explored in this paper is that of multi-task learning. In recent years, as described more thoroughly in 2.2, multi-task learning in the context of deep neural networks has shown promising results. The key insight here is that jointly training a neural model on more than one task, might improve its performance on one or more of those tasks, when compared to it being trained solely on that task.

To our knowledge, no research has yet tackled the fake news detection problem in a multi-task learning setup using attention mechanisms to model document representations. Therefore, in this experiment we aim to construct a hierarchical attention network, similar to the one developed by (Yang et al., 2016), that uses an attention mechanism to construct representations of both sentences and documents, while taking word embeddings as input and passing them through a single-layer bidirectional LSTM. We also want to test whether training a fake news detection model alongside with a natural language inference model will improve the performance on the former task by developing a multi-task learning setup, as proposed by (Liu et al., 2019).

We provide the source code for our models at `https://github.com/sgraaf/JTL_NLI_and_Fake_News_Detection`.

2 Related work

2.1 Fake news detection

The task of fake news detection lives at the intersection of the fields of information retrieval, natural language processing and graph theory. This interdisciplinarity illustrates the presence of fundamentally different approaches that can be applied to detect fake content (online). More specifically, there are 4 general detection strategies as outlined by (Zhou et al., 2019) focusing on knowledge, style, propagation and credibility perspectives.

Combating fake news from a knowledge perspective means treating the problem as an information retrieval task, where facts are extracted both from the article of interest and from verified sources, and the entailment between the two informs the veracity of the article. Here, the main challenges are to identify claims in an article that might be factually incorrect, to find a relevant claim from the corpus of verified statements and to reliably compute whether one claim entails or contradicts the other. The main dataset that allows for experimentation of such kind is FEVER, where for every factually correct or incorrect statement there are several Wikipedia links provided and a label stating whether the link supports, refutes or is not related to the statement (Thorne et al., 2018). The state-of-the-art approach in this subfield is a model developed by (Nie et al., 2018), where the evidence retrieval task is handled by separate neural models for document extraction and sentence extraction, while the entailment task is treated as a Natural Language Inference task enhanced with WordNet-based word embeddings for more accurate semantic matching.

Another approach that does not model language finds patterns in the propagation of fake articles online. The underlying idea here is that fake news spread in social media differently than verified news. Moreover, researchers try to systematically identify users who tend to spread more unverified content. FakeNewsNet (Shu et al., 2018) is often used as a dataset to train such models. It contains a collection of news articles collected and labeled as true or false by GossipCop and PolitiFact, alongside with user interactions from Twitter. These take form in retweets and replies, and existing research models these interactions as hierarchical networks. (Shu et al., 2019) achieve high accuracy by modelling these two layers of propagation networks: on the macro-level it models the retweet patterns across documents, and on the micro-level it captures patterns in replies for a given article.

Finally, to consider the credibility and style aspects of fake news content, natural language models have been developed for identifying untruthful articles. This subfield of fake news detection is rich with available datasets focusing both on short claims, such as statements by politicians in the LIAR dataset (Wang, 2017) or tweets in the PHEME dataset (Derczynski and Bontcheva, 2014), and on entire articles, such as in FakeNewsNet. For every dataset, there are different state-of-the-art results, but the prevailing methodologies are using neural language models that incorporate word embeddings and recurrent neural networks (Karimi et al., 2018; Rashkin et al., 2017).

2.2 Multi-task learning

Multi-task learning has been used as early as 1993 (Caruana, 1993), but it has gained more prominence in recent years as it has been applied more and more in deep neural networks. The list of applications is diverse: researchers train jointly computer vision tasks alongside natural language tasks (Kiela and Bottou, 2014), and enhance model performance on complex problems using simpler examples, such as in the case of using part-of-speech tagging to improve performance on language inference (Hashimoto et al., 2016).

While the general idea of jointly training two or more models sounds rather simple, there are numerous questions that have to be considered and that have already been vigorously researched in the past. On the one hand, there is a variety of ways of implementing multi-task training in practice, each with its advantages and disadvantages. On the other hand, it is worth asking whether training more than one model at the same time should improve performance, and if so, then what is the source of these gains.

When setting up a joint learning model, from a practical side there are several major design choices to be made. A major architectural decision concerns hard parameter sharing, where several tasks share some or all layers in the architecture, or soft parameter sharing, where every tasks has its own model but in the optimization process its parameters are regularized by their distance to parameters of other tasks. While the first one is proven to reduce the risk of overfitting (Baxter, 1997), the second one ensures that models focus on their own tasks and maintain a certain level of flexibility.

This technique works overall due to five major reasons, as outlined by (Ruder, 2017): Implicit data augmentation refers to the fact that combining several tasks means combining several datasets and therefore increasing the total sample size. Focusing attention on the relevant features in noisy data, or attention focusing, also benefits the overall process. Moreover, multi-task learning helps tasks to learn features that one model is confident about while the other isn't, which is referred to in literature as eavesdropping. Finally, representation bias and regularization that are present in joint learning setups refer to the superior qualities of such models in generalizing and preventing overfitting.

3 Methodology

In this section, the model architecture as designed and implemented is presented, which is composed of (1) word embeddings (3.1), (2) an attentative sentence encoder (3.2), (3) an attentative document encoder (3.3) and (4) a classifier (3.4). A summary of our methods is presented in Fig. 1.

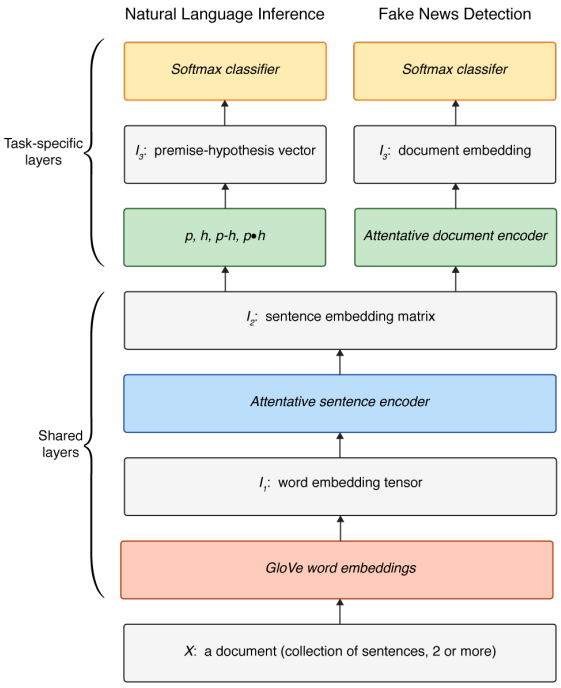

Figure 1: Overview of our model architecture

3.1 Word embeddings

In our model, at the lowest level, the words present in the documents (and sentences) in the input are represented by their *embeddings* (dense, real-valued vectors). For this, we use a pre-trained word embedding model, namely: GloVe (Pennington et al., 2014).

The pre-trained GloVe model used was trained on $840B$ tokens (collected from web crawl data, as provided by CommonCrawl), and produces $300-$dimensional vectors.

3.2 Attentative sentence encoder

Treating a sentence S of length T as a sequence of words w_i, such that:

$$S = [w_1, w_2, \ldots, w_T],$$

one can begin to consider constructing a sentence representation. While the most straightforward method of doing so would be to do some

kind of pooling (e.g. max- or average-pooling) of the word vectors, this would not only ignore the structure of the sentence (word order), but also any sentence semantics.

Instead, we use a single-layer bi-directional Long Short-Term Memory (BiLSTM) (Hochreiter and Schmidhuber, 1997), with an attention mechanism to form a sentence representation vector out of the most informative words present in the sentence. The BiLSTM consists of the forward $\overrightarrow{LSTM}$, which reads the sentence S from w_1 to w_T, and the backward $\overleftarrow{LSTM}$, which reads the sentence S from w_T to w_1:

$$\overrightarrow{h_{wi}} = \overrightarrow{LSTM}(w_i), \forall i \in [1, \ldots, T] \quad (1)$$

$$\overleftarrow{h_{wi}} = \overleftarrow{LSTM}(w_i), \forall i \in [T, \ldots, 1], \quad (2)$$

where the hidden state for a word w_i is given by the concatenation of the forward hidden state $\overrightarrow{h_{wi}}$ and backward hidden state $\overleftarrow{h_{wi}}$:

$$h_{wi} = [\overrightarrow{h_{wi}}, \overleftarrow{h_{wi}}] \quad (3)$$

As mentioned previously, an attention mechanism is then used to aggregate the hidden states h_{wi} of the most informative words in a sentence to form a sentence representation vector. First, the hidden states h_{wi} are passed forward through a one-layer MLP to obtain the hidden representations u_{wi} of the hidden states. The informativeness of a word is then determined as the similarity between the hidden representation u_{wi} and a context vector v_w, which operates at the word-level. These informativeness weights are then passed through a softmax to obtain the normalized informativeness weights α_{wi}. Finally, the sentence representation vector s is computed as a weighted sum of the hidden states h_{wi}, weighted by their normalized informativeness weights α_{wi}:

$$u_{wi} = \tanh\left(W_w h_{wi} + b_w\right) \quad (4)$$

$$\alpha_{wi} = \frac{\exp\left(u_{wi}^\top v_w\right)}{\sum_{j=1}^{T} \exp\left(u_{wj}^\top v_w\right)} \quad (5)$$

$$s = \sum_{j=1}^{T} \alpha_{wj} h_{wj}, \quad (6)$$

where W_w and b_w are the weights and biases of the one-layer MLP, respectively. The context vector v_w is randomly initialized and jointly learned during training. The subscript w denotes that these variables are used at the word-level.

3.3 Attentative document encoder

Treating a document D of length N as a sequence of sentences s_i, such that:

$$D = [s_1, s_2, \ldots, s_N],$$

one can form a document representation vector out of sentence representation vectors similarly to the attentative sentence encoder introduced and described previously. Again, a single-layer BiLSTM with an attention mechanism is used to form a document representation vector out of the most informative sentences present in the document:

$$\overrightarrow{h_{si}} = \overrightarrow{LSTM}(s_i), \forall i \in [1, \ldots, N] \quad (7)$$

$$\overleftarrow{h_{si}} = \overleftarrow{LSTM}(s_i), \forall i \in [N, \ldots, 1] \quad (8)$$

$$h_{si} = [\overrightarrow{h_{si}}, \overleftarrow{h_{si}}] \quad (9)$$

$$u_{si} = \tanh\left(W_s h_{si} + b_s\right) \quad (10)$$

$$\alpha_{si} = \frac{\exp\left(u_{si}^\top v_s\right)}{\sum_{j=1}^{T} \exp\left(u_{sj}^\top v_s\right)} \quad (11)$$

$$d = \sum_{j=1}^{T} \alpha_{sj} h_{sj}, \quad (12)$$

where W_s and b_s are the weights and biases of the one-layer MLP, respectively. The context vector v_s is randomly initialized and jointly learned during training. The subscript s denotes that these variables are used at the sentence-level.

3.4 Classifier

In the case of FND, the document representation vector serves as input to our classifier in order to classify is as either fake or not (real). The classifier concerned is a *softmax* classifier, meaning: a fully-connected layer combined with the softmax function in order to output class probabilities. All the same holds for NLI: The combined premise-hypothesis representation vector serves as input to a fully-connected layer with a softmax function to output the class probabilities.

4 Experiments

4.1 Tasks

As mentioned in 1, the two tasks considered in this paper are (1) Fake News Detection (FND) and (2) Natural Language Inference (NLI). The former entails classifying an article as either fake or not (real), while the latter entails classifying the

entailment type between a premise and hypothesis sentence pair (e.g. *entailment*, *contradiction* or *neutral*).

4.2 Datasets

For our FND task, we use the FakeNewsNet (FNN) dataset (Shu et al., 2018), as mentioned previously in 2.1, the statistics of which can be found in Tab. 1.

Platform	Fake/real	# of articles
GossipCop	Fake	5323
	Real	16817
PolitiFact	Fake	432
	Real	624
	Total:	**23196**

Table 1: Statistics of the FakeNewsNet dataset

Due to privacy policies and copy rights, however, the authors did not publish the full dataset itself. Instead, they published `CSV`-files containing: (1) `id` (unique identifider for each news article), (2) `url` (URL of the news article), (3) `title` (the title of the news article) and (4) `label` (the label of the news article: fake or real) (and `tweet_ids`, but we don't use them here), as well as working code to scrape these articles from the web.

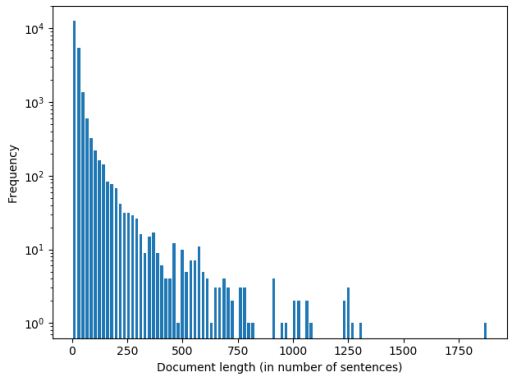

Figure 2: Plot of document length frequencies of FNN after cleaning

As such, we scraped the news articles with the code provided (from the US, using a VPN). What we noticed immediately, however, was that not all scrapes succeeded, as we only managed to obtain 22141 articles instead of 23196. This could be due to the article having been taken down since the time of dataset publication, for example. Another thing we noticed was that the obtained news article scrapes were quite noisy: for quite a few news articles, instead of obtaining the actual news articles themselves, we only scraped the paywalls, privacy notices, sign-up notices, etc. As such, through manual inspection of any such offending string patterns, we removed any samples from the dataset for which the article itself was not scraped successfully, leaving us with 21439 news articles after cleaning. Another issue we found with the dataset, was that some news articles were extremely long compared to others (see Fig. 2). In order to make the training process as smooth and fast as possible, we made the decision to furthermore exclude any articles longer than 40 sentences in length. This ultimately left us with 20428 news articles, instead of the original 23196.

For our NLI task, we used the Stanford NLI (SNLI) corpus, as introduced by Bowman et al. (2015), which contains 570k sentence-pairs with entailment labels.

4.3 Training

For the purposes of this paper, we trained 2 models: (1) our Single-Task Learning (STL) model (only trained on FNN) and (2) our Multi-Task Learning (MTL) model (trained on both FNN and SNLI). These models were trained on the train set(s) of FNN and/or SNLI. During training, the following parameters were used:

```
LEARNING_RATE = 0.1
MAX_EPOCHS = 10
BATCH_SIZE_FND = 32
BATCH_SIZE_NLI = 1024
GLOVE_EMBED_DIM = 300
WORD_HIDDEN_DIM = 100
SENT_HIDDEN_DIM = 100
```

with the learning rate decreasing by factor 2 after every epoch.

For both tasks, we define the cross-entropy loss as the objective function:

$$\mathcal{L}(x, class) = -\log\frac{\exp(x(class))}{\Sigma_i \exp(x(j))}. \qquad (13)$$

To account for the class imbalance between fake and real articles, the loss function in both the STL and MTL setups for the fake news detection task

has been adapted appropriately:

$$\theta_{real} = \frac{1}{\text{\# of real articles}}, \quad (14)$$

$$\theta_{fake} = \frac{1}{\text{\# of fake articles}}. \quad (15)$$

Moreover, to ensure balance between the fake news detection and language inference tasks during the joint learning process, the losses for the two tasks have been combined with the following weights:

$$\lambda_{FND} = \frac{1}{\text{\# of samples in FNN}}, \quad (16)$$

$$\lambda_{NLI} = \frac{1}{\text{\# of samples in SNLI}}. \quad (17)$$

Finally, following (Liu et al., 2018), we introduce dynamic weighted averages to weigh the losses for the FND and NLI task, which has shown to reduce overfitting to a single task in a joint learning setup.

$$\lambda_k(t) = \frac{K \exp(w_k(t-1)/T}{\Sigma_i \exp(w_i(t-1)/T}, \quad (18)$$

where:

$$w_k(t-1) = \frac{\mathcal{L}_k(t-1)}{\mathcal{L}_k(t-2)} \quad (19)$$

For the MTL objective, in any given epoch, the task to be trained was sampled uniformly at random, with a probability proportional to the number of batches in a dataset, to ensure a balanced training process.

4.4 Evaluation

During training, after every epoch, our models (STL and MTL) are evaluated on FND and/or NLI using a validation set of their respective dataset(s). The models are evaluated both by their classification accuracy on this validation set, as well as their loss.

Finally, after training has converged, our models are evaluated on FND only using a kept-aside test set from FNN. The models are evaluated by their classification accuracy, their precision, their recall, and their F1-score.

5 Results

In this section, first the results demonstrating the learning process of the single-task model compared to multi-task model will be discussed, and later the performance of both models on the FNN dataset will be investigated.

5.1 Training process

As can be seen in Fig. 3, it may seem that the model does not converge during the training process of the Single-Task Learning model. The accuracy on the validation set increases after the first epoch, and then consistently decreases until the end of training. Rather than a negative result or an improper training setup, this is indicative of a choice outlined earlier in this paper. For the fake news detection task, the loss function has been weighted to adapt for the class imbalance present in the dataset. Therefore, the model is penalized stronger for misclassifications on fake articles, and as therefore higher performance on these is lead by a decrease in overall accuracy. This is supported by the decreasing loss, and by increasing precision and recall on fake articles over the training process as shown in Tab. 2.

The case is less clear for the Multi-Task learning setup, where the accuracy values do not follow any clear trend. As will be shown in the subsequent subsection, this might have to do with the model not converging over the optimal value.

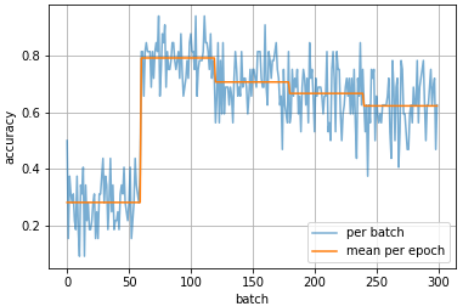

Figure 3: Accuracy on validation set of FNN during the STL training process.

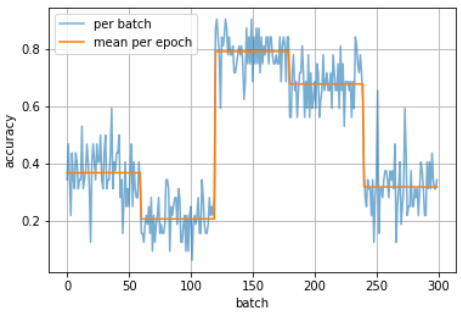

Figure 4: Accuracy on validation set of FNN during the MTL training process.

5.2 Model performance

Using the majority baseline, or simply stating that all articles are non-fake, would lead to a 76% accuracy on the test set of FakeNewsNet. However, this is not favorable, since it is reasonable to assume that correctly labeling fake articles is more important than correctly labeling real articles. Therefore, for the sake of evaluating our models, it is important to look at the precision and recall variables per class.

The main results of the experiment are shown in Tab. 2. As can be seen from the high recall values, the model is indeed focusing more on fake articles. Moreover, the model learns in both the single-task and multi-task learning setups, as can be seen in the increasing precision and recall values over the epochs. Nevertheless, it is not possible to conclude that performance improves in the multi-task learning setup. The trained STL model shows the highest F1-Score values, which is the harmonic mean of the precision and recall, for both real and fake articles. The MTL model, therefore, could not converge, likely due to the presence of a separate language inference task. The exact reasons for this are unclear, as experimenting with the learning rates (the results of which are omitted from this paper) did not lead to convergence either.

6 Discussion

Unfortunately, in our experiment we failed to conclude whether the MTL setup leads to a performance improvement over an STL setup when identifying fake articles. From our experiments we have seen that the model trained solely on FakeNewsNet yields highest performance, but this does not disprove the potential of the joint learning setup. Perhaps, implementing a different weighting scheme between the main task (FND) and the auxiliary task (NLI) as well as considering different parameter sharing schemes might lead to more conclusive results.

One conclusion that can be drawn from this experiment is that text alone may not be enough to reliably detect fake content. Since models as the one proposed in this paper only work with the semantics and the style of articles, the best one can hope for is detecting patterns in the use of language of deceptive content. However, this may be impractical both for applications that would track such articles online and for any automated fact-checking tool. Therefore, it is important that future research investigates the effect of combining language models with information retrieval and propagation network applications such as those described earlier in our research paper.

On a more technical note, it would be beneficial to explore more recent and computationally intensive models, such as the BERT language model or ELMo contextualized word embeddings. In this experiment, we implemented ELMo embeddings but due to their computational complexity could not get presentable results within reasonable timeframes. Perhaps, those results can be finalized and shown in some of our future work.

References

Hunt Allcott and Matthew Gentzkow. 2017. Social media and fake news in the 2016 election. *Journal of economic perspectives*, 31(2):211–36.

Jonathan Baxter. 1997. A bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine learning*, 28(1):7–39.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Rich Caruana. 1993. Multitask learning: A knowledge-based source of inductive bias. In *Proceedings of the Tenth International Conference on Machine Learning*.

Leon Derczynski and Kalina Bontcheva. 2014. Pheme: Veracity in digital social networks. In *UMAP workshops*.

Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. 2016. A joint many-task model: Growing a neural network for multiple nlp tasks. *arXiv preprint arXiv:1611.01587*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hamid Karimi, Proteek Roy, Sari Saba-Sadiya, and Jiliang Tang. 2018. Multi-source multi-class fake news detection. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1546–1557.

Douwe Kiela and Léon Bottou. 2014. Learning image embeddings using convolutional neural networks for improved multi-modal semantics. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 36–45.

David MJ Lazer, Matthew A Baum, Yochai Benkler, Adam J Berinsky, Kelly M Greenhill, Filippo

	STL, E.1: Real/Fake	STL, E.5: Real/Fake	MTL, E.1: Real/Fake	MTL, E.5: Real/Fake
Precision	0.894, 0.248	0.876, **0.369**	0.819, 0.247	**0.904**, 0.262
Recall	0.125, **0.951**	**0.636**, 0.704	0.226 , 0.836	0.203, 0.929
F1-Score	0.219, 0.393	**0.737**, **0.485**	0.355, 0.382	0.332, 0.408

Table 2: Precision, recall and F1-score values for the two models, in the first and the last epochs of the training process. Values are shown per class. E. is a short abbreviation for Epoch. In bold are the highest values.

Menczer, Miriam J Metzger, Brendan Nyhan, Gordon Pennycook, David Rothschild, et al. 2018. The science of fake news. *Science*, 359(6380):1094–1096.

Shikun Liu, Edward Johns, and Andrew J Davison. 2018. End-to-end multi-task learning with attention. *arXiv preprint arXiv:1803.10704*.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*.

Katerina Eva Matsa and Elisa Shearer. 2018. News use across social media platforms 2018.

Yixin Nie, Haonan Chen, and Mohit Bansal. 2018. Combining fact extraction and verification with neural semantic matching networks. *arXiv preprint arXiv:1811.07039*.

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.

Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2931–2937.

Jonathan Rose. 2017. Brexit, trump, and post-truth politics.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan Liu. 2018. Fakenewsnet: A data repository with news content, social context and dynamic information for studying fake news on social media. *arXiv preprint arXiv:1809.01286*.

Kai Shu, Deepak Mahudeswaran, Suhang Wang, and Huan Liu. 2019. Hierarchical propagation networks for fake news detection: Investigation and exploitation. *arXiv preprint arXiv:1903.09196*.

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*.

Udo Undeutsch. 1967. Beurteilung der glaubhaftigkeit von aussagen. *Handbuch der psychologie*, 11:26–181.

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

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.

Xinyi Zhou, Reza Zafarani, Kai Shu, and Huan Liu. 2019. Fake news: Fundamental theories, detection strategies and challenges. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 836–837. ACM.