ner |
Author | Father | Amateur Programmer
Jan 11 · 7 min read

# I trained fake news detection AI with >95% accuracy, and almost went crazy

Fake news is still a real problem

*tl;dr—We made a fake news detector with above a 95% accuracy (on a validation set) that uses machine learning and Natural Language Processing that you can download here. In the real world, the accuracy might be lower, especially as time goes on and the way articles are written changes.*

With so many advances in Natural Language Processing and machine learning, I thought maybe, just maybe, I could make a model that could flag news content as fake, and perhaps take a bite out of the devastating consequences of the proliferation of fake news.

Arguably the hardest part of making your own machine learning model is gathering the training data. It took me days and days to gather pictures of every NBA player in the 2017/2018 season to train a facial recognition model. Little did I know that I'd be diving into a painful, months-long process that exposed some truly dark and disturbing things still being propagated as news and real information.

## Defining fake news

My first obstacle was unexpected. After doing some research into fake news, I very quickly discovered that there are many different categories misinformation can fall into. There are articles that are blatantly false, articles that provide a truthful event but then make some false interpretations, articles that are pseudoscientific, articles that are really just opinion pieces disguised as news, articles that are satirical, and articles that are comprised of mostly tweets and quotes from other people. I Googled around and found some people trying to categorize websites into groups like 'satire', 'fake', 'misleading', etc.

I thought that was as good as place to start as any, so I went ahead and began visiting these domains to try and hunt for some examples. Almost immediately I found a problem. Some websites that were marked as 'fake' or 'misleading' sometimes had truthful articles. So I knew that there would be no way to scrape them without doing a *sanity* check.

Then I started asking myself if my model should take satire and opinion pieces into account, and if so, should they be considered fake, real, or put into their own category?

## Sentiment Analysis

After about a week of staring at fake news sites, I started to wonder if I was already over-complicating the problem. Maybe I just needed to use some existing machine learning models on sentiment analysis, and see if there was a pattern? I decided to build a quick little tool that used a web scraper to scrape article titles, descriptions, authors, and content and post the results to a sentiment analysis model. I used Textbox, which was convenient because it ran locally to my machine and returned results quickly.

Textbox returns a sentiment score which you can interpret as either *positive* or *negative*. I then built a crappy little algorithm to add weights to the sentiments of the different types of text I was extracting (title, content, author etc.) and added it all together to see if I could come up with a global score.

It worked pretty well at first, but after about the 7th or 8th article I tried, it started to fall down. To make a long story short, it was nowhere close to the fake news detecting system I wanted to build.

Fail.

## Natural Language Processing

This is where my friend David Hernandez recommended actually training a model on the text itself. In order to do so, we'd need lots and lots of examples in the different categories we wanted the model to be able to predict.

Since I was pretty worn out from trying to understand patterns in fake news, we decided to just try and scrape domains that were known fake, real, satire, etc. and see if we could build a data set quickly.

After running the crude scraper for a few days, we had a data set we thought was large enough to train a model.

The results were crap. Digging into the training data we realized that the domains never fell into neat little categories like we wanted them to. Some of them had fake news mixed with real news, others were just blog posts from other sites, and some were just articles where 90% of the text were Trump tweets. So we realized we'd have to start over with the training data.

This is when things got bad.

It was a Saturday when I started the long process of manually reading every single article before deciding what category it fell into and then awkwardly copying and pasting text into an increasingly unwieldy spreadsheet. There were some dark, disgusting, racist, and truly depraved things that I read that at first I tried to ignore. But after going through hundreds of these articles, they started to get to me. As my vision blurred and my interpretation of colors got all messed up, I began to get really depressed. How has civilization fallen to such a low level? Why aren't people able to think critically? Is there really any hope for us? This went on for a few days as I struggled to get enough examples for the model to be significant.

I found myself drifting in my own interpretation of fake news, getting angry as I came across articles that I didn't agree with, fighting hard against the urge to only pick ones I thought were right. What was right or wrong anyway?

But finally, I reached the magic number of examples I was looking for, and with great relief, e-mailed them to David.

The next day, he ran the training again as I eagerly awaited the results.

We hit an accuracy of about 70%. At first I thought this was great, but after doing some spot checking with articles in the wild, I realized that this wasn't going to be of any use to anybody.

Fail.

## Fakebox

Back to the drawing board. What was I doing wrong? It was David who suggested that maybe simplifying the problem would be the key to a higher degree of accuracy. So I really thought about what the problem was I was trying to solve. It then hit me; maybe the answer isn't detecting fake news, but detecting real news. Real news is much easier to categorize. Its factual and to the point, and has little to no interpretation. And there were plenty of reputable sources to get it from.

So I went back to the Internet and started to gather training data all over again. I decided to categorize everything into two labels; real and notreal. Notreal would include satire, opinion pieces, fake news, and everything else that wasn't written in a purely factual way that also adhered to the AP standards.

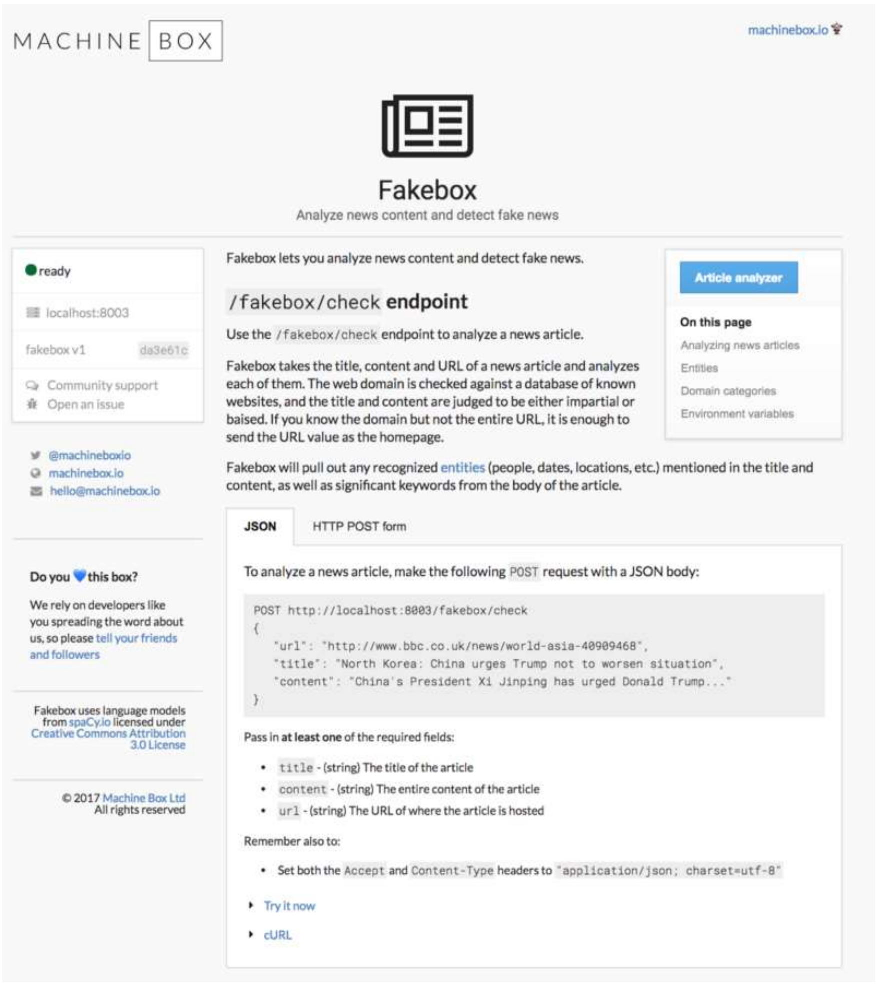
I spent weeks doing this, every day taking a few hours to get the latest content from every kind of website you could imagine from *The Onion* to *Reuters*. I put thousands and thousands of examples of real and notreal content into a giant spreadsheet, that every day I would add hundreds more to. Eventually, I decided I had enough examples to give it another try. So I sent David the spreadsheet and impatiently waited for the results.

I nearly jumped for joy when I saw the accuracy was above 95%. That means we found a pattern in the way articles are written to detect the difference between real news and stuff that you should take with a grain of salt.

Success (sort of)!

## Stop fake news

The whole point of this exercise was to stop the spread of misinformation, so it gives me great pleasure to share that model with you. We call it Fakebox, and its really easy to use.


MACHINE BOX

machinebox.io

Fakebox

Analyze news content and detect fake news

● ready

localhost:8003

fakebox v1 da3e61c

Community support

Open an issue

@machineboxio

machinebox.io

hello@machinebox.io

Do you 💙 this box?

We rely on developers like you spreading the word about us, so please tell your friends and followers

Fakebox uses language models from spaCy.io licensed under Creative Commons Attribution 3.0 License

© 2017 Machine Box Ltd All rights reserved

Fakebox lets you analyze news content and detect fake news.

**`/fakebox/check` endpoint**

Use the `/fakebox/check` endpoint to analyze a news article.

Fakebox takes the title, content and URL of a news article and analyzes each of them. The web domain is checked against a database of known websites, and the title and content are judged to be either impartial or baised. If you know the domain but not the entire URL, it is enough to send the URL value as the homepage.

Fakebox will pull out any recognized entities (people, dates, locations, etc.) mentioned in the title and content, as well as significant keywords from the body of the article.

Article analyzer

**On this page**

Analyzing news articles

Entities

Domain categories

Environment variables

**JSON** HTTP POST form

To analyze a news article, make the following `POST` request with a JSON body:

```
POST http://localhost:8003/fakebox/check
{
   "url": "http://www.bbc.co.uk/news/world-asia-40909468",
   "title": "North Korea: China urges Trump not to worsen situation",
   "content": "China's President Xi Jinping has urged Donald Trump..."
}
```

Pass in **at least one** of the required fields:

- `title` - (string) The title of the article
- `content` - (string) The entire content of the article
- `url` - (string) The URL of where the article is hosted

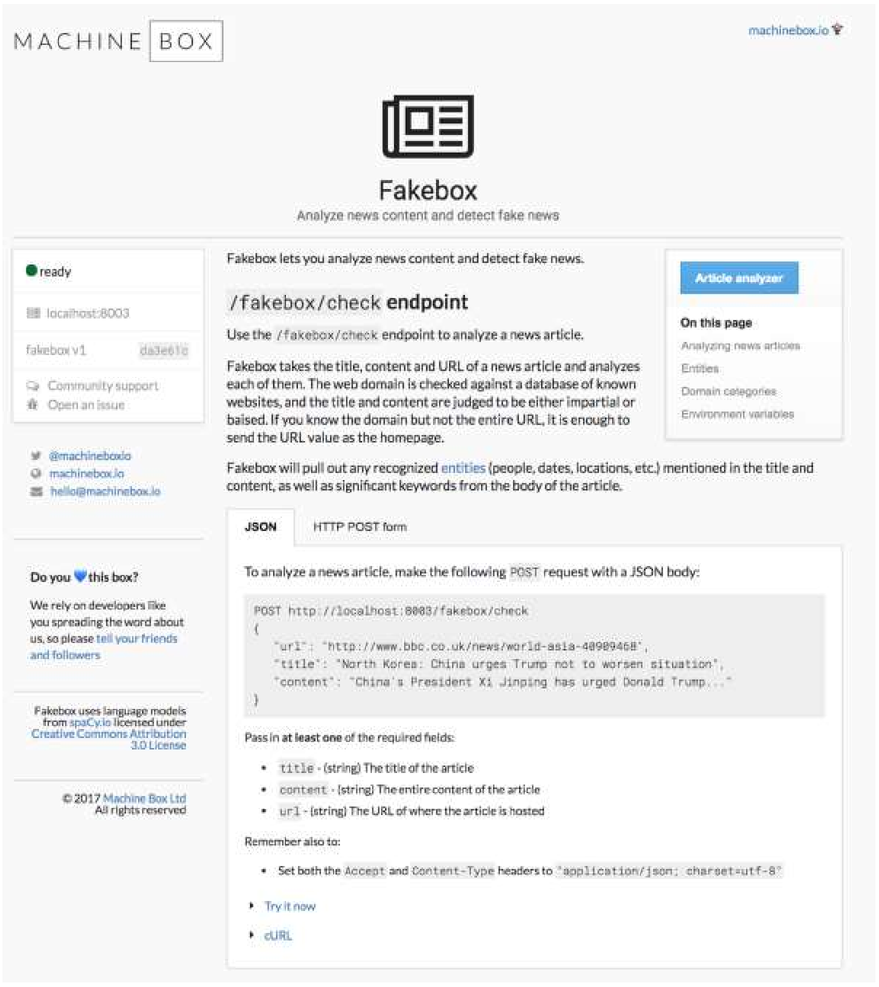
Remember also to:

- Set both the `Accept` and `Content-Type` headers to `"application/json; charset=utf-8"`

▸ Try it now

▸ cURL

Paste the content of an article you're unsure about and click analyze.



MACHINE BOX

machinebox.io

Fakebox

Analyze news content and detect fake news

ready

localhost:8003

fakebox v1 da3e61c

Community support

Open an issue

@machineboxio

machinebox.io

hello@machinebox.io

**Do you 💙 this box?**

We rely on developers like you spreading the word about us, so please tell your friends and followers

Fakebox uses language models from spaCy.io licensed under Creative Commons Attribution 3.0 License

© 2017 Machine Box Ltd
All rights reserved

Fakebox lets you analyze news content and detect fake news.

## `/fakebox/check` endpoint

Use the `/fakebox/check` endpoint to analyze a news article.

Fakebox takes the title, content and URL of a news article and analyzes each of them. The web domain is checked against a database of known websites, and the title and content are judged to be either impartial or baised. If you know the domain but not the entire URL, it is enough to send the URL value as the homepage.

Fakebox will pull out any recognized entities (people, dates, locations, etc.) mentioned in the title and content, as well as significant keywords from the body of the article.

Article analyzer

**On this page**

Analyzing news articles

Entities

Domain categories

Environment variables

**JSON** HTTP POST form

To analyze a news article, make the following `POST` request with a JSON body:

```
POST http://localhost:8003/fakebox/check
{
   "url": "http://www.bbc.co.uk/news/world-asia-40989468",
   "title": "North Korea: China urges Trump not to worsen situation",
   "content": "China's President Xi Jinping has urged Donald Trump..."
}
```

Pass in **at least one** of the required fields:

- `title` - (string) The title of the article
- `content` - (string) The entire content of the article
- `url` - (string) The URL of where the article is hosted

Remember also to:

- Set both the `Accept` and `Content-Type` headers to `"application/json; charset=utf-8"`

▸ Try it now

▸ cURL

Integrate it into any environment with a nice RESTful API. Its a Docker container so you can deploy and scale it anywhere and everywhere you like. Churn through an unlimited amount of content as quickly as you desire, and automatically flag stuff that might need some attention.

**Remember, what its telling you is if an article is written in a similar way to a real news article**, so if the score comes back really low, it might mean the article is fake, an opinion piece, satire, or something other than a straightforward, facts-only news article.

In summary, we trained a machine learning model that analyzes the way an article is written, and tells you if its similar to an article written with little to no biased words, strong adjectives, opinion, or colorful language. It can have a hard time if an article is too short, or if its primarily comprised of other people's quotes (or Tweets). It is not the end all solution to fake news. But hopefully it will help spot articles that need to be taken with a grain of salt.

Please enjoy!