Skyhawk state-of-the-art detection solution

General steps

First of all, in order to create an effective solution, the problem has to be clearly and thoroughly defined. Several important aspects should be considered and discussed by a team of experts.

1. *Purpose*: the overall goal for creating such a system,
2. *Main threats*: known and possible attacks, types, targets, platforms, sources,
3. *Available resources*: computational power, memory space, human capital,
4. *Constraints*: time, speed, scale,
5. *Data*: types, sources, minimal amount, methods and frequency of its collection,
6. *System functionalities*: what it can and cannot do,
7. *System architecture*: data pipelines (acquisition, storing and streaming), processing pipelines (cleaning, representations and transformations), analysis and detection modules (core models), and visualization interface (metrics and formats),
8. *Technologies* to use for system realization,
9. *Evaluation methodology*: how to measure system performance and usefulness,
10. *Methods of improvement and upscaling*.

Proposed solution

A proposed state-of-the-art detection solution will be a multi-functional system containing several modules performing comprehensive analysis while operating on large continuously updated and enriched collections of multi-modal data.

Architecture

The proposed system will contain the following modules:

1. *Data collection and preprocessing* - a module that regularly crawls targeted social media platforms and web resources to acquire new information to analyse and improve the existing solution;
2. *Global analysis* - a module performing constant analysis of collected social media and web data to detect anomalies and trending content, as well as to narrow down to specific portions of data, categories, and sources for a more detailed analysis;
3. *Content analysis* - a module that categorizes suspicious data and describes its properties and content (type, source, topics, sentiments, main “players” (organizations, people, places), etc.);
4. *Deep fake detection* - a module that performs a binary classification on provided data to indicate the likelihood of it being artificially created;
5. *Misinformation detection* - a module that performs fact-checking to reveal whether a given information was fabricated or distorted in the aim of manipulation, deception, or distraction (intentionally flooding the space with noise to divert attention from authentic information);
6. *Pattern matching* - a module that saves information about previous incidents (source and its specificities) in a database and matches current data against known threats;

7. *Correlation* - a module that discovers and highlights connections between information, sources, people/organizations, and behaviours/intentions, as well as constructs graph-based models and dynamic network diagrams helping understand large and complex misinformation campaigns.
8. *Visualization* - plots the results in a graphical user interface in an intuitive and interactive manner, giving the final user a full and clear understanding of the incident, as well as a composite score defining its severity.

The execution starts from *'Data collection and preprocessing'* and *'Global analysis'* modules that constantly monitor media and web space. When a potential threat is detected, it is more precisely analysed with *'Content analysis'* module. Depending on the results, modules *'Deep fake detection'* and/or *'Misinformation detection'* are applied. Then, the incident is matched against a database of patterns from *'Pattern matching'* module. *'Correlation module'* is now activated to reconstruct a complete picture of the incident. Finally, the user is provided with a full report via graphical interface and prompted to take actions according to proposed suggestions.

All the modules will be executed in a cloud environment to leverage computational speed and fast access to data. The results of their execution will be logged and visualized. At the end, if expert feedback is available, the incident's data is assigned a label, which can be used for further improvements of modules or placing it to the pattern database if positive.

Detection models and networks

The most recent and powerful state-of-the-art models should be used subject to interpretability and respect of imposed constraints. As regards to Natural Language Processing, today's most powerful models for text processing and analysis are based on Transformer architecture. A great example is OpenAI's GPT-3 model that can be trained to perform many different tasks. Another candidate is Google's BERT that has been effectively used for context embedding.

For supervised learning (classification and deep fake detection) transfer learning, in which pre-trained models are used and fine-tuned to a targeted task, would be an effective way of approaching the problem, especially at the beginning when less data is available.

Unsupervised learning using clustering can also be applied for anomaly and outlier detection. Auto-encoder neural networks will be a good choice. At the same time, K-means and hierarchical clustering are still considered to be effective approaches given good features, as for example, word embeddings. Different methods for space transformation/reduction (PCA, SVD) and for topic modelling (LDA, LSI) can also be used for text feature engineering.

Generally speaking, neural network architectures will be a preferable choice because of a possibility of high parallelization. As another advantage, they are also well suited for streaming data and progressive model updates without full retraining.

Data

Data is an important aspect of the solution. Data used for training the system should be as realistic, well annotated and abundant as possible. Preferably, training data should come from the same distribution as production data (source, format, content).

If only a small corpus of data is available, pre-trained models (pre-trained word embeddings as an example) should be used to leverage data from other sources or tasks. Also, transfer learning can be performed using many available corpora with a variety of sizes and content. One of the biggest are Amazon reviews, Google News and Wikipedia Extraction. A corpus with a similar content and style should be chosen.

Cleaning and preprocessing data is also important. Miscellaneous special and non unicode characters should be filtered out. Texts should be normalized in terms of case and tokenized. Tokens might be stemmed or lemmatized.

Appropriate labels (unique or multi-class, full or partial) should be assigned according to a given task. Clustering can be used as an exploratory step to understand the data and pre-assign labels.

Evaluation, explanation, production, scaling

Trained models have to be thoroughly tested before being shipped to production. Besides computing standard metrics as accuracy, precision, recall and ROC curves/AUC, a battery of other tests should be performed. The models should be tested on extreme cases and in different staged scenarios. Their robustness and behaviour in those scenarios have to be documented. This is particularly important, as most deep learning models are considered to be black boxes and hardly interpretable.

Before shipping to production, a full documentation has to be provided for all modules and functionalities of the system, preferably with toy examples and usage tutorials.

A subset of core functionalities has to be implemented first. The scaling will be done by adding auxiliary functionalities, detection/analysis tasks and new sources and streams of data.

Improvements

The system will be constantly improved in terms of performance. Expert feedback will be used as one of the methods of improvement. Constant flow of new correctly labeled data will also be important for adjusting and enriching existing models.

Conclusion

Creating an effective state-of-the-art solution requires a concrete and comprehensive definition of the problem, team effort of experts from different fields, and a data-driven multi-modal and multi-step approach.