Fake News Detection in Twitter

Pradnesh Kalkar
Dept. of Computer Science
IIT Guwahati
p.pradnesh@iitg.ac.in
190101103

Saket Kumar Singh
Dept. of Computer Science
IIT Guwahati
saketkumar@iitg.ac.in
190101081

Varenyam Bakshi
Dept. of Computer Science
IIT Guwahati
vbakshi@iitg.ac.in
190101098

***Abstract*—Information extraction is a well-studied topic, and the outputs of such systems enable different natural language technologies such as issue answering and text summarization. Twitter, for example, receives a massive amount of information every second. They are modern modes of communication based on a concept in which the audience scrutinises false claims and applauds true facts. However, because information sources may contain mistakes, it is necessary to confirm the information's veracity. Previous research has shown that bogus news spreads more quicker than actual news on Twitter. As a result, an efficient and accurate claim verification system is critical. To create awareness about this, FEVER workshop and contest is organized where teams develop systems to predict the authenticity of human-generated textual claims against evidence set which is primarily retrieved from Wikipedia articles. In the pipeline proposed by various participating teams, retrieving relevant documents is an important step where a lot of improvement is required. In this paper we would propose a pipeline for twitter fake news detection. We also provide a survey of the document retrieval techniques used in top submissions of the FEVER shared task.**

***Index Terms*—Natural language, textual claims, Twitter**

I. Introduction

The extraction of information is a well-studied subject, and the outputs of such systems allow numerous natural language technologies such as question answering and text summarization. However, because information sources might contain inaccuracies, there is an extra necessity to validate the accuracy of the information. To that end, we organised the inaugural Fact Extraction and VERification (FEVER) shared task to generate interest in and raise awareness of the job of automatic information verification - a research subject that is orthogonal to information extraction. Participants were tasked with developing algorithms that estimate the authenticity of human generated textual assertions against textual evidence gathered from Wikipedia. The systems participating in the FEVER shared task were required to label claims with the correct class and also return the sentence(s) forming the necessary evidence for the assigned label. Most of the submissions in FEVER task divided the whole process of claim verification into three stage pipeline:

1) Document Retrieval
2) Sentence extraction
3) Textual Entailment

The relevant Wikipedia articles for a particular claim are retrieved and sorted in the Document retrieval stage. Given the recovered documents, not all sentences are appropriate to be evaluated in the evidence set during the sentence extraction step. As a result, key sentences are selected and ranked from the retrieved articles. The evidence set is made up of these sorted sentences. In the last stage, the extracted sentences (evidence set) are run through a model with the original claim, and the claim is categorised into one of the following classes: Supported, rejected, and insufficient information.

In this paper however, we tackle the issue of tweets verification. Twitter is plagued with continuous infiltration of fake tweets. Spreading of misinformation results in catastrophic events which can lead to socio-political landslides. Therefore, we propose a pipeline for verifying the tweets by using comments and retrieved hashtags and mentions. In the next section we present a survey of document retrieval techniques employed by the teams over the years in FEVER shared task.

II. Survey of Document Retrieval Techniques in FEVER task

In FEVER task 2018 and 2021 various teams participated to solve the problem of claim verification. They came up with innovative ways of retrieving relevant documents. We studied various methods and gained insights spread across 12 research papers of FEVER shared task (2018), FEVEROUS shared task (2021) and Open domain QA (2021). Here we present those insights:

A. Combining Fact Extraction and Verification with Neural Semantic Matching Networks (2018) - 1st place

They used Keyword Matching to get a set of documents (8 pages per claim). Add all the "non-disambiguative" documents to the final list. Then they randomly selected atmost 5 disambiguative (eg. Title = Savages (band)) documents. A disambiguative title is the one with parentheses, eg. Title = Savages (band). A document is represented as a concatenation of title and first sentence. Then NSMN model is used to calculate <m+, m− > vector for each selected "disambiguative" document.

$$\langle m^+, m^- \rangle = \mathbf{NSMN}(c_i, [t_j, s_j^0]),$$

$$p(x = 1 \mid c_i, j) = \frac{e^{m^+}}{e^{m^+} + e^{m^-}},$$

After that they filter out the documents for which $p(x = 1|c, j)$

¡ threshold. Then sort the remaining documents by m+ values and add top k (k ¡= 5) to the final list of retrieved documents. NSMN model comprises of 4 layers: encoding layer, alignment layer, matching layer, output layer. Its primary use is for ranking the documents according to semantic relatedness with claim $(m + value)$.

B. FaBULOUS: Fact-checking Based on Understanding of Language OverUnstructured and Structured information (2021)

They indexed the documents using title concatenated with all the passages of the article. Ranking of documents was done based on query terms appearing in the document using BM25, which is a bag-of-words technique.

$$\text{score}(D,Q) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f(q_i,D)\cdot(k_1+1)}{f(q_i,D)+k_1\cdot\left(1-b+b\cdot\frac{|D|}{\text{avgdl}}\right)}$$

However BM25 neither takes into account the proximity of terms in the document nor the structure of the sentences. Therefore considerable semantic information is lost in this technique. Mean Average Precision (MAP) evaluation metric was used. They stopped at retrieving 3 documents because increasing the number of documents was counterproductive for passage selection.

C. Team Papelo: Transformer Networks at FEVER (2018)

The team proposed three modifications to the baseline model in order to upgrade the performance.
Modification1: They concatenated every article sentence (premise) with its title to calculate TFIDF values. This increased evidence retrieval accuracy from 66.1 to 68.3 percent.
Modification2: Took help of NER based model in Spacy to find named entities. They also included capitalized phrases in named entities. This increased the accuracy to over 80 percent at the cost of computation time.
Modification 3: As the document consists of many movie related documents, at the time of retrieval they gave preference to the documents with film names in brackets. This is just a dataset related hack and not generalizable.

D. Dense Hierarchical Retrieval for Open-Domain Question Answering - Liu et. al. (2021)

E. Verdict Inference with Claim and Retrieved Elements Using RoBERTa (2021)

They utilized an information retrieval toolkit called *Anserini* built over *lucene*. Lucene is a Java library providing powerful indexing and search features. Anserini is efficient in indexing large document collections and providing state of the art ranking methods. So all the retrieved documents were indexed using this toolkit.
The title and the first 10 elements of each Wikipedia document were first normalized by removing the links. Then Anserini was used to query each claim with the indices built and retrieve k Wikipedia documents most related to the claim as well as their relatedness scores. All of this was done utilizing the indexing and ranking power of Anserini.

F. UCL Machine Reading Group:Four Factor Framework For Fact Finding (HexaF) (2018)

They followed a simple two stage pipeline for document retrieval. First, construct a dictionary of article titles. This is based on observation that majority of the claims include the title. Next Logistic regression is used to calculate the probability of containing the gold evidence using the following features of claim: Position, Capitalization, Presence of stop words. Token match counts between 1st sentence of article and claim. It is different from traditional entity linkage approaches as it considers a wider range of lexical items including the adjectives and verbs.

G. A Fact Checking and Verification System for FEVEROUS Using a Zero-Shot Learning Approach - Temiz et. al. (2021)

It follows a two step process. First is Keyword extraction and preprocessing. First they extract entities from the claim using spaCy. Uppercase words are also considered for entities. They employed a contingency parser for noun chunks. Entities having date elements were ignored. Entities are concatenated twice to the claim to give more weight to entities, since they were using OKAPI BM25. Next step is the actual document retrieval. They used Anserini indexing (which uses OKAPI BM25) for indexing wikipedia pages. Anserini, as mentioned earlier, is a powerful toolkit used for indexing and ranking pages. 10 documents were retrieved for every query (claim + appended entities).

H. Neural Re-rankers for Evidence Retrieval in the FEVEROUS Task (2021)

Baseline model provided by FEVER organizers uses a combination of TF-IDF with entity matching approach for document retrieval. The team tried to enhance baseline model by focusing on the retrieval component through a re-ranking process of pages, resulting in a more precise model. They made two modifications in the model. In the first stage they retrieve as many documents using standard TF-IDF or BM25 approaches with entity linking. Next, they scores and re-ranked the articles using re-ranker model (based on pre-trained BERT model that is fine-tuned on passage re-ranking of MS MACRO dataset to minimize cross-entropy loss). Eventually the best m pages based on the score were returned.

I. Combining sentence and table evidence to predict veracity of factualclaims using TaPaS and RoBERTa (2021)

The technique of matching of TFIDF vectors was employed to efficiently compare a claim with all documents in the dataset. TFIDF matrix size was reduced by

1) stemming the words
2) removing top 10 percent most frequent words
3) removing words with count less than 2

A separate TFIDF matrix for titles was also created to improve the retrieval accuracy. Moreover, bigrams were used to improve accuracy.

J. Papelo at FEVEROUS: Multi-hop Evidence Pursuit

Searching and querying for evidence set using just the claim is a cumbersome task. By just reading the claim one cannot justify the importance of any article. The next hop prediction module attempts to imagine information that is still needed but not retrieved yet. It generates a string consisting of the title of the needed article and the sentence that it wants to retrieve from that article. They supported Multi-hop functionality by predicting whether evidence chains are complete and generating additional search queries based on the preliminary evidence.

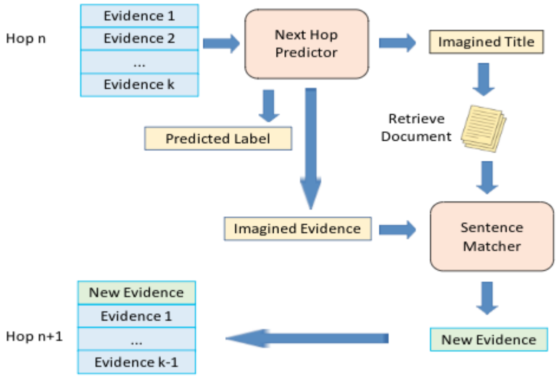

Fig. 1. Multi-hop technique for evidence retreival

This module retrieves documents whose titles match named entities that appear in the claim, plus documents with a high TF-IDF score against the claim, up to five total documents. This is considered the first hop. Next the evidence set and claim are passed through the evidence predictor module where an imaginary If available, the article with that title is retrieved; otherwise, sentences from previously retrieved articles will be searched. Then we choose one sentence with the best word overlap against the imagined evidence. The bottom ranked elements of the evidence set for hop n are pushed out, and these chosen elements are pushed to the top of the evidence set for hop (n+ 1). This way the hopping continues until the model predicts that the evidence is sufficient enough to predict the class of the claim.

III. Fake news and Twitter

People are increasingly seeking and consuming news from social media rather than conventional media as smartphones become more widely available (e.g., newspapers and TV). We may now share various forms of knowledge and debate it with other readers thanks to social media. However, it appears to have become a hotspot of fake news, with possibly harmful societal consequences. The propagation of incorrect information is not primarily the result of bots trained to propagate inaccurate reports. Instead, fake news spreads quicker on Twitter as a result of people retweeting incorrect news pieces. Like the claim verification problem of FEVER shared task, fake news detection on twitter is also of serious concern and an open challenge to all the pioneers of deep learning. Huge amount of effort and resources are being poured into this research. The two problems are similar in the following ways:

1) Both the problems can be solved by retrieving evidence set regarding the hypothesis (claim in case of FEVER shared task and tweet in case of fake news detection in twitter).
2) The size of claim is comparable to a tweet size (which is 280 characters at maximum).

It may seem the two problems are interchangeable. But that is not the case. The two problems have contrasting differences in the following ways:

1) Tweet language is informal and may contain slang. The claim dataset provided on the other hand, is formal and carefully curated by the organizers.
2) Tweets contain additional information in form of mentions, hashtags and comments by other users. This information is absent in the claim but can be very helpful in verifying a tweet.

The task of extracting and ranking the evidence set for each tweet is a time consuming as well as challenging task. In the upcoming section we shall discuss the evidence set extraction pipeline that we propose.

IV. Our Pipeline

Taking inspiration from the survey we conducted on FEVER submissions, and modifying their implementations in order to accommodate the environment of tweets, we propose a pipeline focusing on evidence extraction. Here we take into account the mentions and hashtags occurring in the original tweet. For ranking a given set of documents, we used stsb-mpnet-base-v2 model with pretrained parameters. This model was invoked three times throughout the pipeline. We used politifact dataset to test our model. Next we divide the pipeline into smaller steps and discuss them in detail.

A. Using Hashtags and Mentions

In the politifact dataset used, there were several hashtags and mentions of other users in most of the tweets. This provided an enormous source of potential evidence set. We take this into account by extracting the related tweets of the same day using both hashtags and mentions. Using Tweepy library, we queried the tweets on the basis of hashtags and users profiles mentioned. For one hashtag or mention, 20 tweets were retrieved. They were passed through stsb-mpnet-base-v2 pretrained model to sort them according to relevance with the original tweet. This model was further invoked for ranking documents in later stages of pipeline as well. After ranking, a total of 5 additional tweets were retrieved.

B. Named Entity recognition

Combining the original tweet with the additional tweets, we now have a corpus from which we shall extract named entities. Named entities correspond to people, place and organization. These named entities are required for the next step where they are used as a query to fetch related news and wikipedia articles. For performing named entity recognition we used dslim/bert-base-NER of hugging face library. It has been trained to recognize four types of entities: location (LOC), organizations (ORG), person (PER) and Miscellaneous (MISC).

C. Fetching Documents

Now we start fetching articles. Using the named entities extracted in the last stage we query to APIs for Wikipedia and New York Times (using pynytimes API). For each entity we are fetching 5 Wikipedia and 2 news articles. These numbers are a subject of experimentation and computation time. For news article we consider only the abstract and leading paragraph. Whereas for the wikipedia article we are fetching the summary of the article (upto three sentences).

Fetching a news article takes a lot more time than fetching a wikipedia article. Once we have retrieved enough documents, we pass them through the stsb-mpnet-base-v2 model to rank them on relevance with the original tweet. Then we select the top 5 articles. Now these comprise our retrieved document set.

D. Extracting Sentences

Now all the sentences of the retrieved documents are compared to the original tweet using the ranking model used earlier. Then the sentences are sorted according to the relevance score and top 5 sentences are taken, rest all are discarded. These 5 sentences are the final evidence set retrieved for the original tweet and will be used for verifying the tweet.

E. Entailment

The final evidence set sentences are concatenated and they form the premise. The original tweet is the hypothesis, which we are testing. The hypothesis and premise are passed through the bert-base-uncased model. This model is primarily aimed at being fine-tuned on tasks that use the whole sentence (potentially masked) to make predictions, such as sequence classification, token classification or Entailment. On passing the premise and hypothesis through the model, a classification is made for the original tweet as either fake or real.

V. Dataset and Training

For this task we used *politifactreal* and *politifactfake* dataset which contains around 10,90,000 (1 million) tweets. It has various fields like timestamp, tweet data, user id and tweet id. For each tweet we used *Tweepy* library to retreive the tweets of hashtags and mentions used in the original tweet. This way we succesfully created a dataset of related tweets which could be later used for evidence extraction and eventual textual entailment.

For training, we fine tuned the pretrained model by adding a feed forward layer at the end of the BERT models used for entailment and ranking documents.

VI. Conclusion

Although we could not test on a considerable number of tweets, on the tweets we could test our pipeline our model had an accuracy of over 63 percent. But the number of tweets which could be tested, due to lack of time, were only a few hundred.

VII. Future Work

We have not taken special measures to take into account the english slangs which are common in many tweets. A transformer based model could be employed in order to convert the slang english words to proper english vocabulary. Number of tweets of each hashtag and mention to consider for validating the tweet can be tuned as hyperparameters instead of just hardcoding it as 20. This number can be experimented upon and with the help of hyperparameter tuning, better results can be acheived.

So far, our focus has been mostly on enhancing the document retrieval part. We used transfer learning for the entailment stage, by fine tuning an already pretrained BERT based model. New model architectures can be explored for the entailment task.

References

[1] UKP-Athene: Multi-Sentence Textual Entailment for Claim Verification Hanselowski AZhang HLi ZSorokin DSchiller BSchulz CGurevych I

[2] Extracting topical information of tweets using hashtags Alp ZOduducu S

[3] Opinion Retrieval in Twitter Argument Mining View project AILaw View project Opinion Retrieval in Twitter Luo ZWang TOsborne M

[4] Improving Twitter Retrieval by Exploiting Structural Information Luo ZOsborne MPetrovi´c SWang T

[5] A Fact Checking and Verification System for FEVEROUS Using a Zero-Shot Learning Approach Temiz OKılıç OKızılda˘ AGba Ta¸skaya TTemizel T

[6] Neural Re-rankers for Evidence Retrieval in the FEVEROUS Task Saeed MAlfarano GNguyen KPham DTroncy RPapotti P

[7] Combining sentence and table evidence to predict veracity of factual claims using TaPaS and RoBERTa Funkquist M

[8] Dense Hierarchical Retrieval for Open-Domain Question Answering Liu YHashimoto KZhou YYavuz SXiong CYu P

[9] Team Papelo at FEVEROUS: Multi-hop Evidence Pursuit Malon C

[10] Verdict Inference with Claim and Retrieved Elements Using RoBERTa Gi IFang TTzong RTsai H

[11] FaBULOUS: Fact-checking Based on Understanding of Language Over Unstructured and Structured information Bouziane MPerrin HSadeq ANguyen TCluzeau AMardas JAi B

[12] Team Papelo: Transformer Networks at FEVER Malon C

[13] Combining Fact Extraction and Verification with Neural Semantic Matching Networks Nie YChen HBansal M

[14] The Fact Extraction and VERification (FEVER) Shared Task Thorne JVlachos ACocarascu OChristodoulopoulos CMittal A

[15] The Fact Extraction and VERification Over Unstructured and Structured information (FEVEROUS) Shared Task Aly RGuo ZSchlichtkrull MThorne JVlachos AChristodoulopoulos CAlexa ACocarascu OFacebook A

[16] The Fact Extraction and VERification (FEVER) Shared Task Thorne JVlachos ACocarascu OChristodoulopoulos CMittal A

[17] Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks Tai KSocher RManning C

[18] Enhanced LSTM for Natural Language Inference Chen QZhu XLing ZWei SJiang HInkpen D

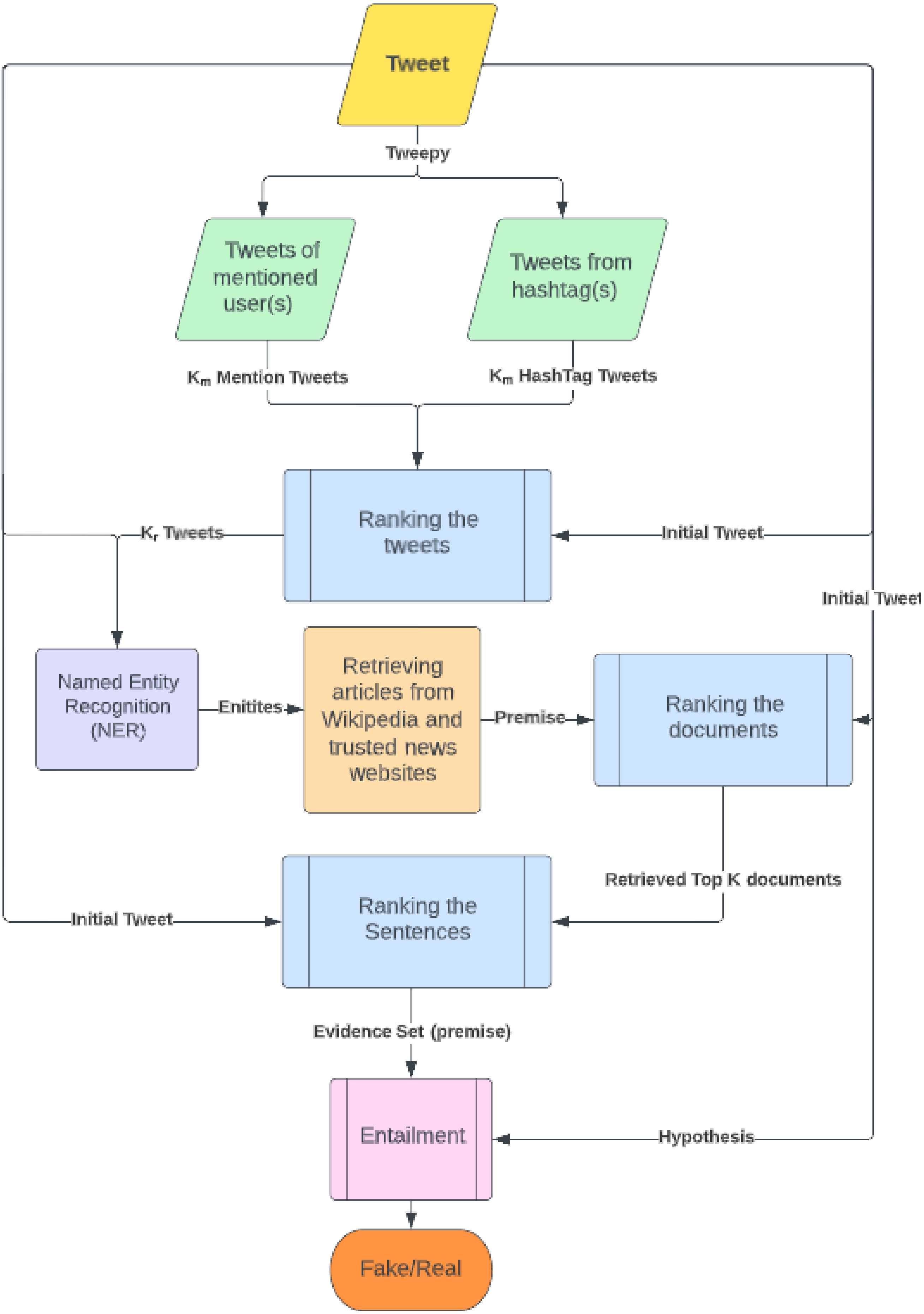

Fig. 2. Our Pipeline for Evidence retrieval and subsequent tweet verification