

Claim Generation for Fact Verification Models

Information Retrieval

Antwerp, Belgium 15th December 2023

Konstantina Ellina
Nada Chehbouni
Pablo de Vicente

Introduction	2
Zero-shot Fact Verification	3
FEVER Dataset	3
Overview	3
Creation	4
Evaluation	4
Claim Generation Method	5
Named Entity Recognition (NER)	6
Question Generation	6
Claim Generation	7
Supported claims	8
Refuted claims	8
NEI claims	8
Example for the whole procedure	9
Evaluation	10
Results	10
Limitations	11
Complementary work	11
Transforming Question Answering Datasets Into Natural Language Inference Datasets	11
Team Papelo: Transformer Networks at FEVER	12
Better system for FEVER challenge	13
Entailment Module	13
Improvements Over Baseline	13
Evidence Retrieval Strategy	14
Topics Covered	15
Bibliography	16

Introduction

Fact verification models are widely utilized across various domains, including journalism and scientific documentation. With the vast volume of information available today, combating misinformation is a significant challenge. Fact Verification (FV) algorithms serve as a tool to manage this information overload, helping to discern truth from falsehood by checking statements.

The challenge with Fact Verification (FV) models is the significant amount of data required for training, which must be specifically annotated for each field. This process demands substantial resources in terms of both finances and human labor. However, the reality is that obtaining extensive training data for every new domain necessitating fact verification is impractical. Generating this data often involves soliciting humans to formulate claims and seek supporting or opposing evidence—a task that proves exceedingly expensive.

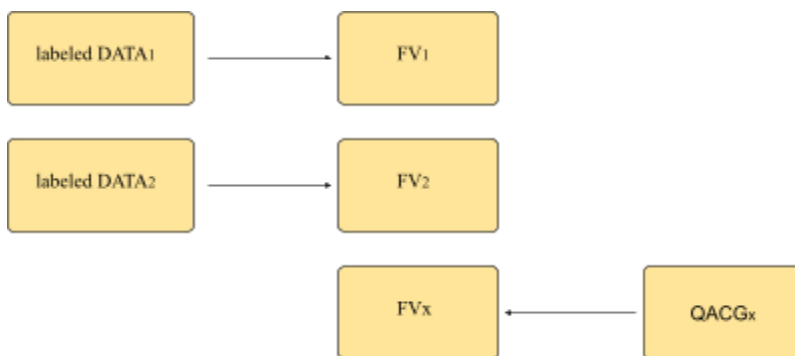


Figure 1: Manually labeling data is an expensive task needed to train specific-domain models. To solve this we use automatically generated claims

In this paper we present a newly proposed method for generating evidence pairs in the form of <Evidence,Claim> pairs that can be used to train a wide variety of models. While the process is intricate, we will introduce relevant scientific research and explain it while at the same time trying to avoid going into too much detail.

This project is divided into two main sections, the first one, **Zero shot Fact Verification** is our main work and the one that we will spend more time on. The second one describes **complementary work** and models (used in ZSFV) in detail, and it is there to add context to the models used.

Zero-shot Fact Verification

The paper addresses this problem by exploring the possibility of automatically generating large-scale (evidence, claim) pairs to train the fact verification model. It proposes a simple yet general framework Question Answering for Claim Generation (QACG) to generate three types of claims from any given evidence: 1) claims that are supported by the evidence, 2) claims that are refuted by the evidence, and 3) claims that the evidence does Not have Enough Information (NEI) to verify. We'll explore the specifics of how this framework achieves said goals in the following sections.

The concept of "zero-shot" learning refers to the ability of a model to perform tasks without the need for any human-annotated training examples. Instead, the model learns solely from the data it generates. While pre-training ("few-shot") can improve performance, the core objective of this project lies in achieving adaptability without relying on additional training data.

This paper will predominantly focus on three main points

1. **Dataset selection:** Introduction to FEVER dataset
2. **Claim Generation**
3. **QA2D Model:** Transforming Question Answering Datasets Into Natural Language Inference Datasets

FEVER Dataset

While not the primary focus of this presentation, we find that understanding the construction and characteristics of the FEVER dataset is essential for understanding its importance and why it sees so much use in applications involving natural language processing (NLP).

Overview

FEVER, an acronym for Fact Extraction and VERification, stands out as a comprehensive dataset designed specifically for claim verification. It comprises a vast collection of sentences, each accompanied by a set of documents that can be used to determine whether the sentence is supported, refuted, or inconclusive. This unique format makes FEVER a valuable tool for evaluating the performance of NLP models in assessing the veracity of claims. The dataset was constructed in two stages:

1. Claim Generation

For this specific task, a Wikipedia dump (as of June 2017) was processed and used to sample sentences from the introductory sections of ~50,000 pages. Annotators were then given these sentences and asked to generate a set of claims containing a single piece of information related to the original topic.

2. Claim Labeling

Following claim generation, annotators categorized the claims into three classes: SUPPORTED, REFUTED, or NOTENOUGHINFO. For the first two classes, annotators were required to provide evidence supporting or refuting the claim. In cases where information from Wikipedia was insufficient to validate or invalidate the claim, the label NOTENOUGHINFO was applied.

Creation

While it is true that we have previously discussed the concepts behind FEVER, we also wanted to investigate **how** a dataset is actually constructed. Thus, we present the methodology used in order to generate both stages mentioned before:

1. **Document Retrieval:** The first step is to obtain the necessary wikipedia pages. Using cosine similarity between binned unigram and bigram Term Frequency (also known as TF-IDF), it is possible to return the k most relevant documents for a query.
2. **Sentence Selection:** After obtaining the documents, the next step was choosing the sample sentences that would be given to annotators later on. In order to achieve this task, the sentences were ranked by TF-IDF similarity to the query (a sample claim), afterwards, the highest ranked ones were chosen.
3. **Not Enough Info class:** Training instances were simulated for the NOTENOUGHINFO class through two methods. The first one sampled a sentence from the nearest page (NEARESTP) to the claim as evidence using the document retrieval component. The second one sampled a sentence at random from Wikipedia (RANDOMMS).

Evaluation

As with any serious work, tests were conducted for each of the previous steps. We will detail the three main tests that were taken out

Classification Task

The task is a 3-way classification task which evaluates and predicts whether a claim is SUPPORTED, REFUTED, or NOTENOUGHINFO. Evaluation is based on accuracy, where correct evidence must be provided at a sentence-level to justify the classification for the first two classes (SUPPORTED and REFUTED).

Accuracy Metric

Accuracy is the primary metric used for evaluating the classification task. The random baseline is mentioned to be approximately 33%, considering the balanced class distributions and ignoring the requirement for evidence for SUPPORTED and REFUTED.

Evidence Retrieval Evaluation

The correctness of the evidence retrieved is evaluated using the F1-score of all predicted sentences compared to the human-annotated sentences for claims requiring evidence. For claims requiring multihop inference involving sentences from more than one document, all sentences must be selected for the evidence to be marked as correct. This is reported as the proportion of fully supported claims. The system is penalized for selecting information that annotators did not choose, emphasizing precision. The goal is to avoid false positives in evidence selection.

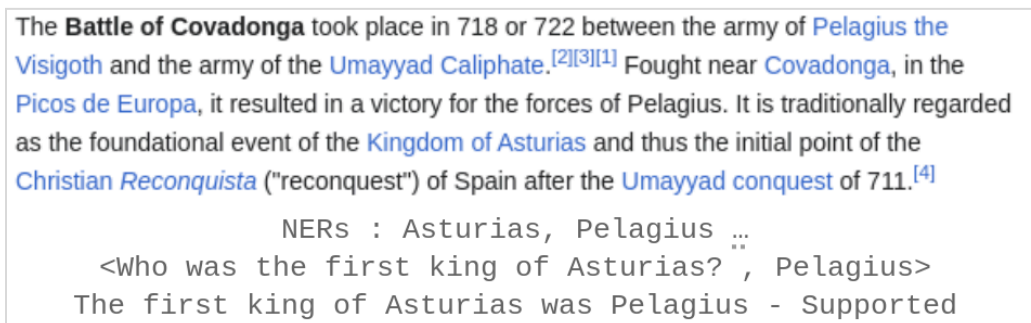
Claim Generation Method

Now that we already have context on what the FEVER dataset contains, it is time to talk about the next step of the process. This is the NER extraction, the Question Generation and finally the Claim Generation, which is our main goal.

We have provided a short glossary, as the terminology used can be a bit overwhelming. Any further terms will be described as they come up

- P : Original passage from which to extract information
- (Q,A) : Questions and Answers generated from the passage
- C : Claims are produced for each question according to the following rules:
 - If P answers Q and A is correct then C = supported
 - If P answers Q and A is incorrect then C = refuted
 - If P can not answer Q then C = NEI (not enough information)
- G : pretrained question generation model
- M : pretrained QA to claim model
- E : All entities contained in a passage

There are three major steps in the process (as illustrated in Figure 2), the first of which is the Named Entity Recognition (**NER**), so we will focus on said one first.



The **Battle of Covadonga** took place in 718 or 722 between the army of **Pelagius the Visigoth** and the army of the **Umayyad Caliphate**.^{[2][3][1]} Fought near **Covadonga**, in the **Picos de Europa**, it resulted in a victory for the forces of Pelagius. It is traditionally regarded as the foundational event of the **Kingdom of Asturias** and thus the initial point of the **Christian Reconquista** ("reconquest") of Spain after the **Umayyad conquest** of 711.^[4]

NERs : Asturias, Pelagius ...

<Who was the first king of Asturias? , Pelagius>

The first king of Asturias was Pelagius - Supported

Figure 2: Example of passage, entity extraction, <question, answer> pair and supported claim

Named Entity Recognition (NER)

The goal in this step is to extract Named Entities from the evidence of each sample. Named entities are essential for generating meaningful claims, they are often central to the meaning of a text, their extraction helps in understanding the key entities involved in a claim or statement. Named entity recognition involves identifying and classifying entities, such as persons, organizations, locations, and more, within a given text. We have presented the reader with an example.

Context

"Over 120 colleges and universities are located in New York City , including Columbia University , New York University , and Rockefeller University , which have been ranked among the top 35 in the world."

Entity	Type
120	CARDINAL
NEW YORK CITY	GPE
COLUMBIA UNIVERSITY	ORG
NEW YORK UNIVERSITY	ORG
ROCKEFELLER UNIVERSITY	ORG
35	CARDINAL

When generating claims or statements from a given context, the entity types can be used to ensure that the generated claims are coherent and semantically meaningful. The entity types provide semantic labels of the extracted entities. In this example, "Cardinal" has the same meaning as in mathematics , "GPE" stands for Geopolitical Entities (Countries, cities, states...) and "ORG" is short for Organizations.

The extraction is executed with **Stanza**. Stanza is a Python natural language processing library that includes a Named Entity Recognizer (NER) among its tools. Stanza's NER component utilizes deep learning techniques to accurately identify and categorize named entities in multiple languages. In *Extract_NERs.py*, the processing of the text and the identification of the named entities are completed by the Stanza NLP library with the use of a Pipeline. The script iterates through the samples in the FEVER dataset, processes the context of each sample using Stanza, and extracts the identified named entities.

Question Generation

Afterwards comes the **Question Generation** step, where, using a pre-trained Question Generation (QG) model, we generate (question, answer) pairs from the evidence.

To accomplish this task, the BART model was enhanced through fine-tuning on the SQuAD (Stanford Question Answering Dataset) dataset. This model is designed to generate questions given input text and answer entities, making it suitable for converting evidence into interrogative queries.



Figure 3.1 : Sequence overview of SQuAD query generation

So, taking the results of the NER extraction in the previous step, we convert these named entities to questions. Given an input text **D** (e.g. Asturias) and an answer **A** (e.g. Pelagius), the question generator outputs a question **Q** (e.g. Who was the first king of Asturias? | Pelagius).

Claim Generation

Lastly, claims are then generated based on pairs of **Question** and **Answers (QA-to-Claim)**. The QA-to-Claim Model takes the generated question **Q** (e.g. Who was the first king of Asturias?) and the corresponding answer **A** (e.g. Pelagius) as inputs and produces a declarative sentence **C** (e.g. The first king of Asturias was Pelagius) for the **(Q, A)** pair.

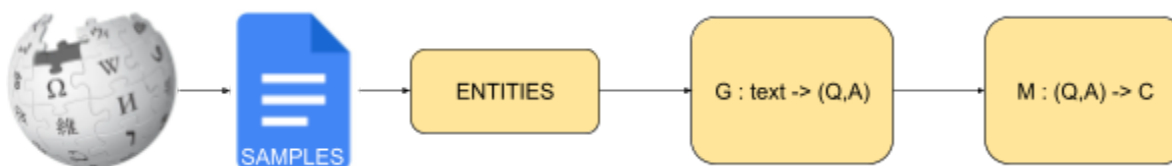


Figure 3: Sequence overview of the hole process

This model uses the BART paradigm and the **SQuAD** dataset to approach the problem as a sequence-to-sequence (**seq2seq**) task. The model may produce logical and contextually appropriate declarative sentences by taking into account the contextual relationships between the inquiry and answer, thanks to the **seq2seq** technique.

Seq2seq models, like the ones used in our QA-to-Claim Model and Question Generator, are efficient at converting questions and other variable-length input sequences into meaningful and coherent output sequences, which makes them ideal for complex language creation tasks in natural language processing.

For each of the claims (Supported, Refuted and NEI) there is a specific approach on how to generate them, we will detail this process in the following sections. We have also provided examples at the end, as not to disrupt the reading flow.

Supported claims

Given an evidence **P** (which, to remind the reader, will be a passage of a text) , we use named entity recognition to identify all entities within **P**, denoted as **E**. For each entity $\mathbf{a} \in \mathbf{E}$, we treat **a** as an answer and generate a question $\mathbf{q} = \mathbf{G}(\mathbf{P}, \mathbf{a})$.

The question-answer pairs **(q, a)** are then sent to the **QA-to-Claim model (QA2D)** which then transforms existing question/answer pairs into claims. We will explain this model later.

Supported claims are the easiest to generate, as they follow the steps detailed in the previous sections.

Refuted claims

In order to generate refuted claims, the idea behind the procedure is to replace the original answer with an alternative one, trying to introduce intentional inaccuracies into the claims.

The method starts with a question-answer pair **(q, a)**. It uses answer replacement to replace the answer **a** with another entity **a_0** of the same type *. The goal is to replace **a** with an incorrect answer to the question **q**. To achieve this, we follow the described steps:

1. **Selecting a Replacing Answer:** The paper uses the pre-trained *Sense2Vec* model to find the top-5 most similar phrases to the answer **a**. Then, it randomly selects one of these phrases as the replacement answer **a_0**.
2. **Generating the Refuted Claim:** The new question-answer pair **(q, a_0)** is fed into a model **QA-to-Claim**. This model uses the information in the pair to generate a new statement, which is the refuted claim.

The method tries to avoid cases where **a_0** is actually a correct answer to the question **q**. It defines rules to ensure that **a_0** has less overlap in words with the original answer **a**. Even with these rules in place, it is not assured that correct replacements will be made, but they are few and far between so they are left as natural noise for the model.

*Entities **a** and **a_0** must be of the same type, as to create coherent sentences.

NEI claims

For **NEI** claims, additional context is extracted from Wikipedia using **Stanza NLP**. The **QA2D model**, which transforms Question Answering Datasets into Natural Language Inference Datasets and will be explained later, is applied to produce claims with the help of the generated questions.

For the **NEI**(Not Enough Info) claims, a question q' which is relevant but cannot be answered by P is generated. To this end, the passage is linked back to its original Wikipedia article W and the evidence is expanded with additional context $P+$.

$P+$ consists of five randomly-retrieved sentences from W that are not present in P (original passage). Both P and $P+$ are then concatenated and considered as the expanded evidence, based on which supported claims are generated given an entity in $P+$ as the answer. This results in a claim relevant to, but unverifiable by, the original evidence P .

Example for the whole procedure

The examples below illustrate the sequence of the models used in the information retrieval pipeline, so we can completely understand the procedure followed in this report.

Named Entity Recognition (NER)

- Original Text: "Albert Einstein, a renowned physicist, was born on March 14, 1879, in Ulm, Germany."
- NER Output: [(Albert Einstein, PERSON), (March 14, DATE), (1879, DATE), (Ulm, GPE), (Germany, GPE)]

Question Generation (QG)

- Input to QG: The output of NER for a specific piece of evidence.
- QG Output: Generated (question, answer) pairs related to the identified entities.
 - "Who is Albert Einstein?" → "Albert Einstein is a PERSON."
 - "When was March 14 born?" → "March 14 is a DATE."

Claim Generation

- Input to Claim Generation: Original evidence (the text) along with the (question, answer) pairs generated by QG.
- Claim Generation Output: Synthetically generated claims categorized as SUPPORTED, REFUTED, or NEI.
 - **SUPPORTED** Claim Example: "Albert Einstein is a renowned physicist."
 - **REFUTED** Claim Example: "Stuttgart is the birthplace of Albert Einstein."
 - **NEI** Claim Example: "In addition to Germany, Albert Einstein had connections to other countries."

Evaluation

The evaluation of the fact verification depends on three different test sets based on FEVER. The idea is to contrast our generated claims with previously labeled information.

- **FEVER-S/R**: Since only the supported and refuted claims are labeled with gold evidence in FEVER, the claim-evidence pairs of these two classes from the FEVER test set are used for evaluation.
- **FEVER-Symmetric**: This is a carefully designed unbiased test set to detect the robustness of the fact verification model with the supported and refuted claims.
- **FEVER-S/R/N**: The full FEVER test set is used for a three-class verification. It is used to retrieve evidence sentences for NEI claims.

A BERT model and a RoBERTa-large model fine-tuned on the FEVER training set are the supervised models. The models are trained on the generated QACG-Filtered dataset (Question Answering for Claim Generation). For binary classification (FEVER-S/R and FEVER-Symmetric), only the supported and refuted claims are used for training, while for FEVER-S/R/N, the full training set is used.

Claims-evidence couples are difficult to interpret, but the BERT model—which has bidirectional context understanding—and the RoBERTa-large model—which improves and expands on BERT's architecture—are skilled at doing so. They efficiently sort through big databases and extract relevant proof by relying on prior knowledge. We use these models' ability to improve Precision, Recall, and F1 Score by training them on the FEVER training set along with the QACG-Filtered dataset.

Results

The best evaluation results are achieved with the RoBERTa-large model. Across all datasets and among some other zero-shot baselines, this model has the best results in Precision, Recall, and F1 Score. In Figure 4, we can see the baseline models and the supervised models with their evaluation results.

	Model	FEVER -Symmetric	FEVER-S/R	FEVER-S/R/N
		$P / R / F_1$	$P / R / F_1$	$P / R / F_1$
<i>Supervised</i>	S1. BERT-base (Devlin et al., 2019)	81.5 / 81.3 / 81.2	92.8 / 92.6 / 92.6	85.7 / 85.6 / 85.6
	S2. RoBERTa-large (Liu et al., 2019)	85.5 / 85.5 / 85.5	95.2 / 95.1 / 95.1	88.0 / 87.9 / 87.8
<i>Zero-shot</i>	U1. Random Guess	50.0 / 50.0 / 50.0	50.0 / 50.0 / 50.0	33.3 / 33.3 / 33.3
	U2. GPT2 Perplexity	52.7 / 52.7 / 52.7	55.6 / 55.6 / 55.6	35.3 / 35.3 / 35.3
	U3. MNLI-Transfer	62.2 / 55.5 / 58.7	63.6 / 60.5 / 61.8	41.4 / 39.6 / 40.7
	U4. LM as Fact Checker (Lee et al., 2020b)	71.2 / 64.5 / 67.8	77.9 / 65.6 / 70.2	64.3 / 54.6 / 49.8
	U5. QACG (BERT-base)	73.2 / 73.0 / 72.9	74.2 / 74.0 / 74.1	56.5 / 55.7 / 55.9
	U6. QACG (RoBERTa-large)	77.3 / 77.0 / 77.1	78.1 / 78.1 / 78.1	64.6 / 62.0 / 62.6

Figure 4: Fact verification performance for supervised and zero-shot models on the three test sets

Limitations

Naturally, there are still some limitations in the model. We've decided to highlight the three most important ones.

- **Dependency on Pretrained Models:** The system's performance relies on the quality of the pretrained models that are used.
- **Contextual Restrictions:** The scalability of generating NEI claims is limited since it necessitates more context from the FEVER dataset.
- **Named Entity Sensitivity Recognition:** The accuracy of NER extraction affects the accuracy of claim generating.

We should also note that we were not able to run the last part of the fact verification model, as many errors arose (installation wise) that we could not resolve. Instead, we decided to take a deep dive into the different concepts of Natural Language Processing and explain (to the best of our abilities), the complete system overview. We hope that this has provided some insight to the reader and encouraged further exploration of topics that, due to time restrictions, we were not able to address.

Complementary work

In this second section we wanted to mention another big part of this project. While researching we always strived for a good understanding of the topics covered, which often led us deeper and deeper into NLP semantics. We hope that this complementary work is used for expanding the knowledge as well as providing context for the reader.

Transforming Question Answering Datasets Into Natural Language Inference Datasets

The **QA-to-Claim Model** used in the '**Zero-shot Fact Verification by Claim Verification**' is a BART model that is finetuned on the **QA2D** dataset which is the subject of this paper (where QA2D means to combine **Questions** and **Answers** into **Declarative** answer sentences)

The database has been created based on the already existing SQuAD dataset which contains pairs of questions and answers based on Wikipedia articles. For each Q&A pair, a declarative sentence is generated, and based on entailment, an NLI example is then derived from the related Q&A example. Given the text passage **P**:

- If **A** is a correct answer to **Q**, then (P,D) is an entailed **NLI pair**, where P is the text passage and D is the generated sentence.

- If **A** is an incorrect answer or **Q** cannot be answered using the information provided in **P**, then the generated sentence **D** is considered as not being implied by **P**, thus leading to (P,D) being a **negative NLI pair**.

In order to derive declarative sentences from question-answer pairs in this work, three different ways have been explored:

- **A rule-based system:** Generating grammatically correct sentences can still become a challenge when the answers in the Question-Answer pair are named entities (*Where does Sam work?/WHO - Referring to the World Health Organization*). This will lead to sentences being generated without the proper syntax (*Sam works at WHO*). Relying on part-of-speech tagging (POS) and parsing accuracy, its overview can be seen in Figure 5.

Q: Where does Jim go to buy groceries? **A:** Trader Joe's

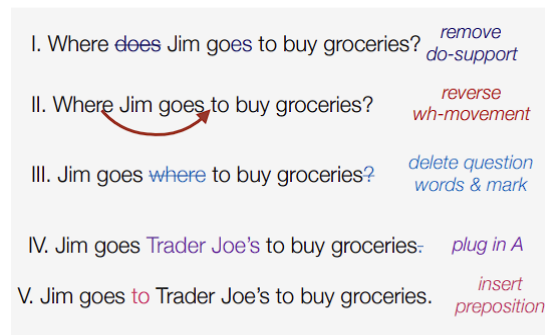


Figure 5: Syntactic transformations needed to perform QA2D using the rule-based system

- **Crowdsourcing:** It would provide a collection of declarative sentences generated by humans. Participants were provided with Question-Answer pairs and asked for two things. First, to generate sentences from scratch, and second, to edit an output generated by the rule-based model. This step was used for comparison purposes to the neural model.
- **Use of neural sequence model:** Because the rule-based system had some weaknesses, the authors aimed to overcome these problems with the use of a neural sequence model.
- with the aim of overcoming the rule-based system weaknesses.

Both the rule-based system and the neural sequence model performed relatively well but the latter had a slightly stronger overall performance.

Team Papelo: Transformer Networks at FEVER

Because of FEVER's baseline low evaluation scores, many researchers tried to find a better solution for it. In our case, we used the system that was introduced by the Papelo team which was used as a retrieval method to collect evidence sentences for the **NEI class**.

Better system for FEVER challenge

The paper addresses the challenges posed by the FEVER challenge by developing a system centered around a transformer network-based entailment classifier. The primary goal is to enhance precision in classifying a wide range of potential evidence, allowing for improved recall in claim verification. The entailment module evaluates evidence statements individually, considering not only articles with the highest TF IDF scores but also additional ones based on named entities and capitalized expressions in the claim. This approach aims to overcome the complexities of FEVER's longer and more abstract sentences, increased prevalence of named entities, and the intricate retrieval process. The system strives to refine baseline approaches to retrieval and entailment by training a sharp entailment classifier. Comparative analysis reveals that **the transformer network, with pre-trained weights, outperforms other models**, especially in handling out-of-vocabulary words.

Entailment Module

The system's core is an entailment module built on a transformer network, a smart tool for understanding sequences. This transformer uses a separator to split the main idea from the argument and has twelve blocks to compare and understand information. It's like a smart reader trained on lots of languages from books. They fine-tuned this smart reader on FEVER examples to help it work better with tricky sentences, which is really good at figuring out complex connections in sentences. The transformer's adaptability in capturing intricate dependencies within sequential data proves crucial in the specific context of entailment classification, particularly when addressing the distinctive challenges presented by the FEVER dataset.

Improvements Over Baseline

There are numerous improvements that the Papelo team accomplishes over the base FEVER dataset. We'll highlight the most impactful causes, but we strongly suggest reading the entire paper.

- The system improves upon the baseline FEVER approach, which concatenated five premise statements and assessed them together. Instead, this system evaluates each premise statement individually, allowing for a **more precise understanding**.
- Some FEVER claims need multiple statements as evidence, but our system focuses on claims that can be supported or refuted with a single sentence, **avoiding complexities** in handling multiple evidence statements.
- Decisions about each sentence are aggregated to decide the **overall claim classification**. If any sentence supports the claim, the claim is classified as supported. If any sentence refutes

the claim without support, the claim is classified as refuted. If no evidence supports or refutes, the claim is marked as lacking information.

- To handle ambiguity, the system is made aware of the Wikipedia page title, providing contextual information. **Adding titles to sentences** significantly improves system performance, making it more accurate in assessing the claim.
- The system naturally handles **out-of-vocabulary words**, especially names and terms, without additional modifications, providing an advantage over other models.
- Various strategies, such as incorporating titles, named entities, and film-related terms, progressively improve evidence retrieval accuracy. **Retrieving entire articles**, along with **named entities and film-related terms**, achieves the highest accuracy.
- The FEVER baseline system uses the Enhanced Sequential Inference Model (ESIM), which is a neural network architecture commonly used for natural language inference tasks. To balance these data, class reweighting is applied, ensuring fair evaluation and preventing bias. When the ESIM is replaced by the transformer network in our case, the transformer exhibits superior performance, **eliminating the need for class reweighting** and showcasing its effectiveness in intricate entailment tasks on the FEVER dataset.

Evidence Retrieval Strategy

Commencing with a baseline retrieval system that extracts sentences from the top five TFIDF-ranked articles, the study progressively refines this process. Simple yet effective modifications, such as adding titles to premise statements during TF IDF calculation, contribute to a modest increase in evidence retrieval to 68.3%. However, the adoption of more sophisticated strategies, including named entity recognition (NER) and film-related queries, substantially elevates the evidence retrieval rate to 81.2%. The most transformative adjustment involves eliminating TFIDF sentence ranking, enabling the retrieval of entire articles. This leads to a remarkable **evidence retrieval rate of 90.1%**, showcasing the system's adaptability to handle a diverse pool of 2.6 million statements for classification. These rates can be found in the table below.

System	Retrieval
FEVER Baseline (TFIDF)	66.1%
+ Titles in TFIDF	68.3%
+ Titles + NE	80.8%
+ Titles + NE + Film	81.2%
Entire Articles + NE + Film	90.1%

Table 3: Percentage of evidence retrieved from first half of development set. Single-evidence claims only.

Topics Covered

During the making of the project we came upon a wide range of topics covered in class, some of which we saw worthwhile mentioning.

One of the first contents we saw in our lectures was **Term frequencies** and **TF-IDF cosine similarity**. Although we did not explicitly use this, it was necessary for the creation of the initial FEVER dataset that we took as input.

When extracting entities from sentences (our NER module), we make use of the **Stanza NLP** library. This is also used when generating NEI claims from <q,a> pairs.

We used **Transformers** for the generation of Question-Answer pairs implemented with **BART**, which is a transformer-based model. We also use **BERT** and **RoBERTa-large** for the fact verification problem and the fine-tuning of the FEVER-training set.

As with most training models, when it comes to evaluating results these are measured with **Precision**, important to assess the system's accuracy in identifying true positive claims, **Recall**, that ensures that the system does not miss relevant claims, **F1-score**, a balance between precision and recall, and other metrics related to (not exclusively) information retrieval.

We also came across some definitions that are used in the baseline system of the FEVER dataset. The first one is **multi-layer perceptron (MLP)** and the second one is **decomposable attention (DA)**, both used for textual entailment recognition.

Bibliography

Thorne, J., Vlachos, A., Christodoulopoulos, C., & Mittal, A. (2018). FEVER: a Large-scale Dataset for Fact Extraction and VERification. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*.

Dorottya Demszky, Kelvin Guu, and Percy Liang. 2018. Transforming question answering datasets into natural language inference datasets. arXiv.org.

Pan, L., Chen, W., Xiong, W., Kan, M., & Wang, W. Y. (2021b). Zero-shot Fact Verification by Claim Generation. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*

Christopher Malon. 2018. Team Papelo: Transformer Networks at FEVER. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 109–113, Brussels, Belgium. Association for Computational Linguistics.