NLP Project

Apoorva Kumar - 16CS10006
Gaurav Jha - 16CS10020
Harshad Gavali - 16CS10021

Table of content

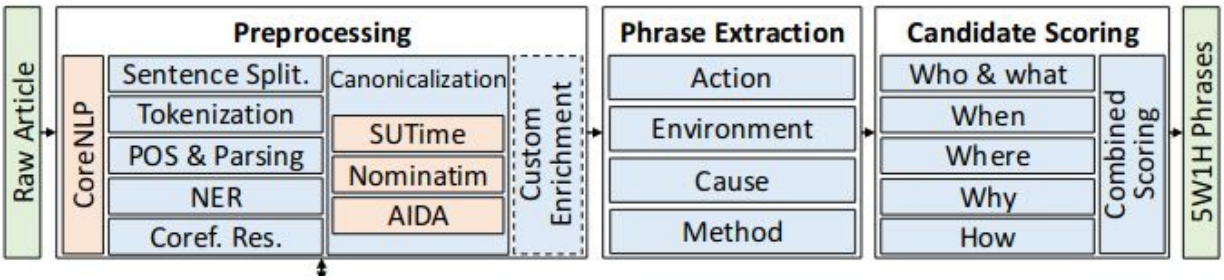
1. Problem Statement

The main objective of this task is to retrieve the arguments of an event (or events) from a news report in an Indian language. Event extraction and retrieval of its corresponding arguments is a much researched topic that has attracted quite some attention in the research diaspora.

2. Introduction

The extraction of a news article's main arguments is an automated analysis task at the core of a range of use cases, including news aggregation, clustering of articles reporting on the same event. Events and arguments are closely related; arguments are often actors or participants in events and events without arguments are uncommon. The interpretation of events and entities is highly contextually dependent. Events are things that happen or occur; they involve entities (people, objects, etc.) who perform or are affected by the events and spatio-temporal as-pects of the world. Understanding events and their descriptions in text is necessary for any generally-applicable machine reading systems. It is also essen-tial in facilitating practical applications such as news summarization, information retrieval, and knowl-edge base construction.

3. Overview of the System



Taliban attacks German consulate in northern Afghan city of Mazar-i-Sharif with truck bomb

The death toll from a powerful Taliban truck bombing at the German consulate in Afghanistan's Mazar-i-Sharif city rose to at least six Friday, with more than 100 others wounded in a major militant assault.

The Taliban said the bombing late Thursday, which tore a massive crater in the road and overturned cars, was a "revenge attack" for US air strikes this month in the volatile province of Kunduz that left 32 civilians dead. [...] The suicide attacker rammed his explosives-laden car into the wall [...].

Figure 1: News article [1] consisting of title (bold), lead paragraph (italic), and first of remaining paragraphs. Highlighted phrases represent the 5W1H event properties (who did what, when, where, why, and how).

The system extracts 5W1H phrases that describe the most defining characteristics of a news event, i.e., who did what, when, where, why, and how. It uses coreference resolution, question-specific semantic distance measures, combined scoring of candidates, and extracts phrases for the 'how' question. The values of the parameters introduced in this section result from a semi-automated search for the optimal configuration of using an annotated learning dataset including a manual, qualitative revision.

3.1 Preprocessing

Preprocessing of News Articles accepts as input the full text of a news article, including headline, lead paragraph, and body text. The user can specify these three components as one

or separately. Optionally, the article's publishing date can be provided, which helps parse relative dates, such as "yesterday at 1 pm".

During preprocessing, we use Stanford CoreNLP for sentence splitting, tokenization, lemmatization, POS-tagging, full parsing, NER and pronominal and nominal coreference resolution. Since our main goal is high 5W1H extraction accuracy (rather than fast execution speed), we use the best-performing model for each of the CoreNLP annotators, i.e., the 'neural' model if available. We use the default settings for English in all libraries.

After the initial preprocessing, we bring all NEs in the text into their canonical form. Canonical information is the preferred output, since it is the most concise form. Because model uses the canonical information to extract and score 'when' and 'where' candidates, we implement the canonicalization task during preprocessing.

We parse dates written in natural language into canonical dates using SUTime. Subsequently, SUTime converts each temporal phrase into a standardized TIMEX3 instance. TIMEX3 defines various types, also including repetitive periods. Since events according to our definition occur at a single point in time, we only retrieve datetimes indicating an exact time, e.g., "yesterday at 6pm".

Geocoding is the process of parsing places and addresses written in natural language into canonical geocodes, i.e., one or more coordinates referring to a point or area on earth. We look for tokens classified as NEs of the type location. Lastly, we geocode the merged phrases with Nominatim , which uses free data from OpenStreetMap.

We canonicalize NEs of the remaining types, e.g., persons and organizations, by linking NEs to concepts in the YAGO graph using AIDA. The YAGO graph is a state-of-the-art knowledge base, where nodes in the graph represent semantic concepts that are connected to other nodes through attributes and relations.

3.2 Phrase Extraction

Extraction Phase performs four independent extraction chains to retrieve the article's main event: the action chain extracts phrases for the 'who' and 'what' questions, environment for 'when' and 'where', cause for 'why', and method for 'how'.

The action extractor identifies who did what in the article's main event. The main idea for retrieving 'who' candidates is to collect the subject of each sentence in the news article. For each 'who' candidate, we take the VP that is the next right sibling as the corresponding 'what' candidate . To avoid long 'what' phrases, we cut VPs after their first child NP, which long VPs usually contain.

The environment extractor retrieves phrases describing the temporal and locality context of the event. To determine 'when' candidates, we take TIMEX3 instances from preprocessing. Similarly, we take the geocodes as 'where' candidates.

The cause extractor looks for linguistic features indicating a causal relation within a sentence's constituents. We look for three types of cause-effect indicators: causal conjunctions, causative adverbs, and causative verbs. Causal conjunctions, e.g. "due to", "result of", and "effect of", connect two clauses, whereas the second clause yields the 'why' candidate. For

causative adverbs, e.g., “therefore”, “hence”, and “thus”, the first clause yields the ‘why’ candidate.

Causative verbs, e.g. “activate” and “implicate”, are contained in the middle VP of the causative NP-VP-NP pattern, whereas the last NP yields the ‘why’ candidate. For each NP-VP-NP pattern we find in the parse-tree, we determine whether the VP is causative. To do this, we extract the VP’s verb, retrieve the verb’s synonyms from WordNet and compare the verb and its synonyms with the list of causative verbs, which we also extended by their synonyms. If there is at least one match, we take the last NP of the causative pattern as the ‘why’ candidate.

The method extractor retrieves ‘how’ phrases, i.e., the method by which an action was performed. The combined method consists of two subtasks, one analyzing copulative conjunctions, the other looking for adjectives and adverbs. Often, sentences with a copulative conjunction contain a method phrase in the clause that follows the copulative conjunction, e.g., “after [the train came off the tracks]”. If a token matches, we take the right clause as the ‘how’ candidate. The second subtask extracts phrases that consist purely of adjectives or adverbs, since these often represent how an action was performed. We use this extraction method as a fallback, since we found the copulative conjunction-based extraction too restrictive in many cases.

3.3 Candidate Scoring

The last task is to determine the best candidate of each 5W1H. The scoring consists of two sub-tasks. First, we score candidates independently for each of the 5W1H questions. Second, we perform a combined scoring where we adjust scores of candidates of one question dependent on properties, e.g., position, of candidates of other questions.

To score ‘who’ candidates, we define three scoring factors: the candidate shall occur in the article early and often, and contain a named entity. The first scoring factor targets the concept of the inverse pyramid: “news mention the most important information, i.e., the main event, early in the article, while later paragraphs contain details”. However, journalists often use so called hooks to get the reader’s attention without revealing all content of the article . Hence, for each candidate, we also consider the frequency of similar phrases in the article, since the primary actor involved in the main event is likely to be mentioned frequently in the article. We also weight candidates which contains Named Entity more since actors are often named Named Entities.

Due to the strong relation between agent and action, we rank VPs according to their NPs’ scores. Hence, the most likely VP is the sibling in the parse tree of the most likely NP.

We score temporal candidates according to four scoring factors: the candidate shall occur in the article early and often. It should also be close to the publishing date of the article, and of a relatively short duration. Reasons for the first two are the same as given for ‘who’ candidates. Events reported on by news articles often occurred on the same day or on the day before the article was published, hence the third criterion. Since according to our definition event happens in exact time, fourth candidates prefer those events.

The scoring of ‘location’ candidates follows four scoring factors: the candidate shall occur early and often in the article. It should also be often geographically contained in other location candidates and be specific. Third factor is another measure of ofteness. According to our definition events also happens on precise location, we prefer candidates which have small area(i.e are specific).

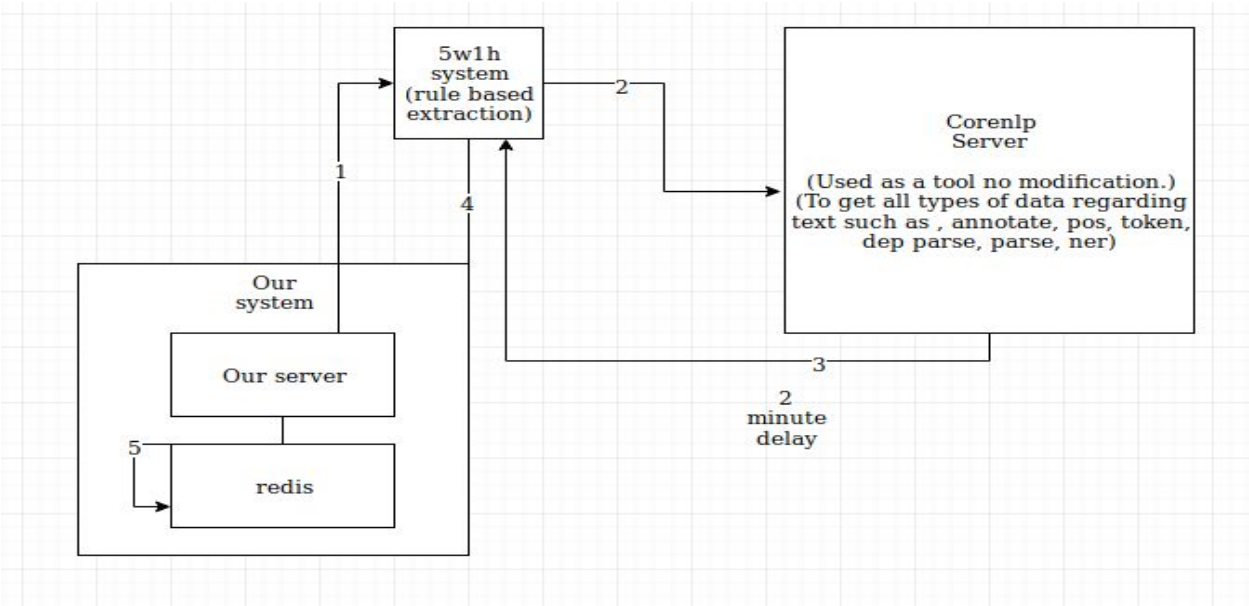
Scoring causal candidates was challenging, since it often requires semantic interpretation of the text and simple heuristics may fail. We define two objectives: candidates shall occur early in the document, and their causal type shall be reliable . The second scoring factor rewards causal types with low ambiguity e.g., “because” has a very high likelihood that the subsequent phrase contains a cause.

The scoring of method candidates uses three simple scoring factors: the candidate shall occur early and often in the news article, and their method type shall be reliable.

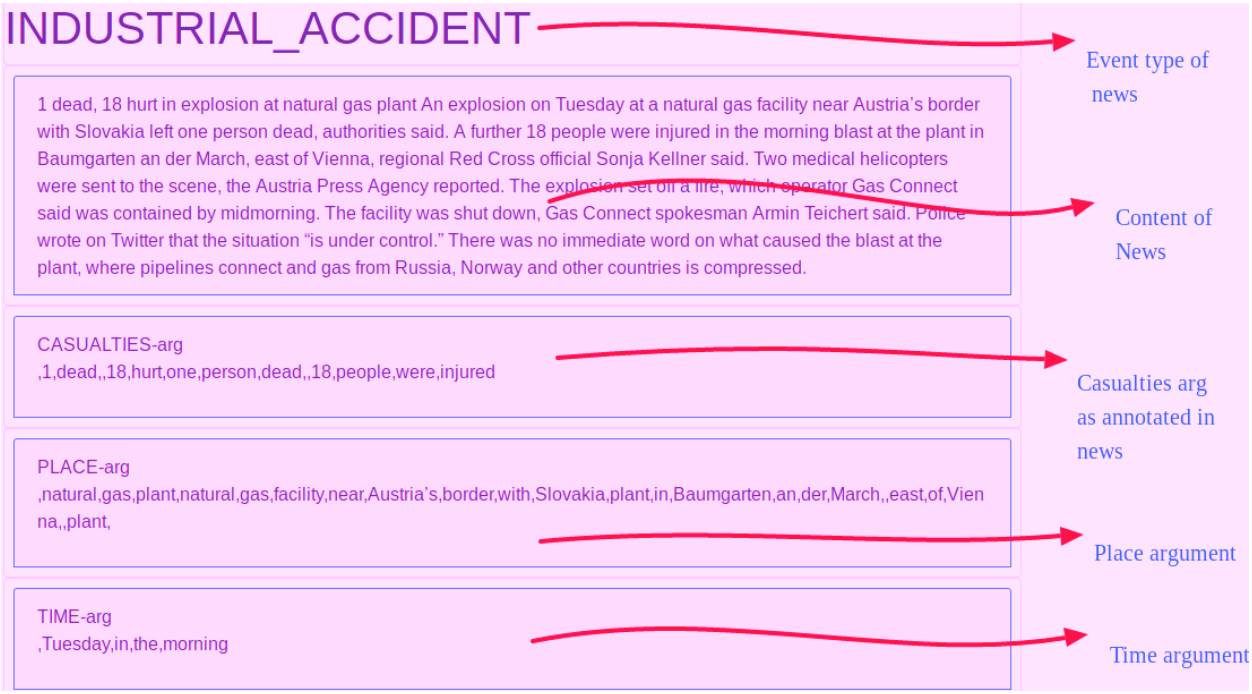
The final sub-task in candidate scoring is combined scoring, which adjusts scores of candidates of a single 5W1H question depending on the candidates of other questions. To improve the scoring of method candidates, we devise a combined sentence distance scorer. The assumption is that the method of performing an action should be close to the mention of the action.

4. Our contribution

We did rigorous statistical analysis on the current datasets of whose outputs we will be providing in the next section. We also created a tool of great significance where if you add new annotated documents you can view the result of our system immediately. It is fully integrated with the 5W1H system completing the whole pipeline. So the whole pipeline looks like :

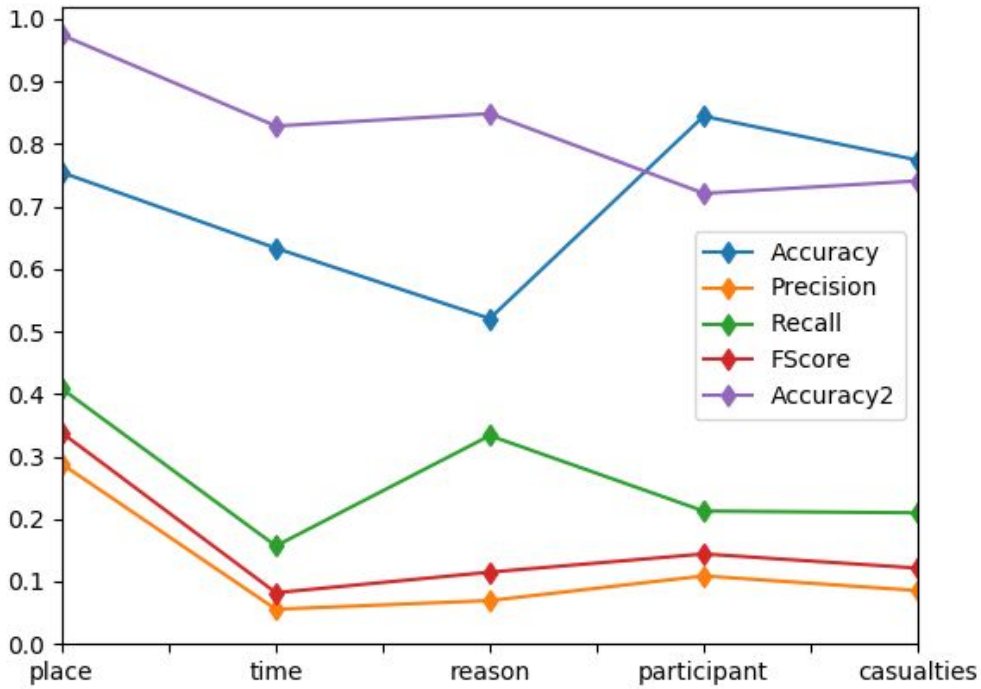


Now the output of the system is provided as shown



5 Results and Discussion

	Accuracy	Precision	Recall	FScore
Place	93.13	25.87	32.26	28.71
Time	97.42	28.77	40.81	33.75
Reason	82.87	5.54	15.68	8.19
Participant	84.84	6.93	33.33	11.48
Casualties	72.09	10.88	21.26	14.39
After-effects	74.08	8.53	20.98	12.13



5.1 Confusion Matrix for Place Argument

Predicted/Actual	1	0
1	2435	6977
0	5112	161686

5.2 Confusion Matrix for Time Argument

Predicted/Actual	1	0
1	1088	2693
0	1578	160504

5.3 Confusion Matrix for Reason argument

Predicted/Actual	1	0
1	620	10563
0	3334	66625

5.4 Confusion Matrix for Participant Argument

Predicted/Actual	1	0
1	282	3783
0	564	24064

5.5 Confusion Matrix for Casualties Argument

Predicted/Actual	1	0
1	3963	32450

0	14676	117813

5.6 Confusion Matrix for After-Effect argument

Predicted/Actual	1	0
1	830	8895
0	3126	33527

Ablation Analysis

2.1 Merits

1. It provides decent accuracy for all the expected arguments.
2. Easy to setup and run.

2.2 Demerits

1. In where it only gives argument which name of a place or state or country. For example, “The fire caught in the building” it won’t give “building” as where argument.
2. As you can see that false positive is high, this is due to the fact that data is not properly annotated and we are mapping one or two arguments from the same arguments provided by our system.

Future Work

We would like to improve our f-score and precision for reason, participant and casualties.