**Topic**

Create a summary of objective claims given a set of news articles in the same subject

**Description**

Much of the information contained in news articles are non-claims that are influenced by the author’s stance on a subject. Our goal is to filter out this subjectivity and present only the objective facts to the user. This way they are not biased to think about a subject in any particular way. Given a set of news articles in a particular subject, our tool will extract the objective claims and generate a human-readable summary.

**Pipeline**

* Step 1: Classify news articles by subject and preprocessing.
  + Stretch goal if we complete the other steps ahead of schedule.
  + For now, we select a few articles in one subject manually.
  + Dataset should include news articles from a variety of sources e.g. conservative vs liberal, reliable vs unreliable etc.
  + These articles are then stored in a \*.csv file (as described below) to be used later
  + Anaphora resolution should happen as a preprocessing step before the text gets dismantled into sentences. Using 3rd party software *enrycher* [2] or *render* [3]
* Step 2: Claim extraction
  + 1st filter (ClaimBuster):
    - make a standard for file format
    - Extract claims on the same news subject from different sources using the ClaimBuster API (website is offline but API is still working)
    - Filtering can be done based on:
      1. Threshold for ClaimBuster score (possibly with an upper limit).
      2. Sorting by CalimBuster score and keeping the top n sentences.
      3. Sorting by ClaimBuster score and keeping the top n%.
  + 2nd filter (TF-IDF summarizer):
    - Build a TF-IDF vector for each sentence. This vector represents how relevant the sentence is to the topic, as well as its information density.
    - We rank the sentences according to the vectors, keeping the top N sentences and discarding the rest. We make the assumption that the highest ranked sentences are the ones that best summarize the article.
    - NLTK’s Reuters corpus used as background corpus for calculating IDF.
  + It is still unclear whether is better to filter sequentially or in parallel. Sorting in “parallel” could be achieved by filtering based on a score derived from ClaimBuster’s check-worthiness factor and the TF-IDF score.
* Step 3: Post-process claims and references
  + Merge similar claims/separate contrasting claims
    - Sentences/claims could be clustered based on SVM as described in [1]
    - Do we want to go further and get into semantics?
      1. Will shallow parsing do? Deep (semantic) parsing is difficult to implement.
      2. One way around parsing may be to separate news sources by stance (e.g. left-wing, centre, right-wing) and create a summary for each stance.
  + Generate human-friendly summary given list of claims extracted
    - Sort claims in a readable order. Order must be logical and sequential, based on sequence of claims in original article.
    - Give score and sources
  + Provide references
    - Specify references for claims, if available
    - Create reference map to illustrate how claims spread?

Step 4: Fake news detection/indicate reliability of claims

* ClaimBuster doesn’t fact check claims. We may want to indicate the reliability of the claims e.g. by assigning weights based on the claim’s origin.

**Data sources**

* BBC News
* The Guardian
* New York Times
* etc.
* It might be interesting to add more controversial news sites e.g. Breitbart, InfoWars, occupyDemocrats etc.
* Chart of bias of news sources: https://www.marketwatch.com/story/how-biased-is-your-news-source-you-probably-wont-agree-with-this-chart-2018-02-28
* List of fake news website: https://en.wikipedia.org/wiki/List\_of\_fake\_news\_websites

**Data collection methods**

Currently data collection is happening manually. If the is enough time it could be interesting to gather data through:

* Scraping
* RSS feed readers
* APIs
* etc.

**Subjects data has been gathered on**

* Euthanasia
* North South Korean Handshake
* Edward Snowden
* Florida midterm vote recount
* Accidental shooting
* Synagogue Shooting

**Example**

|  |  |  |
| --- | --- | --- |
| **Source 1** | **Source 2** | **Objective summary** |
| A mass shooting occurred at a school in X yesterday at 2:00 pm.  10 people were injured.  This is second mass shooting in state Y this year.  More gun control is needed. | 10 people were hurt in a high school shooting yesterday afternoon.  The incident happened at 2:00 pm.  It's the second mass shooting that state Y has experienced in 2018.  According to some eyewitness accounts, the shooter was prevented from causing more damage by an armed security guard.  Gun-rights groups say this is proof that equipping school security with more arms is a solution to the school shooting crisis. | A mass shooting occurred at a school in X yesterday at 2:00 pm.  (score=xxx)[source:1]  10 people were injured. (score=xxx)[source: 1,2]  It's the second mass shooting that state Y has experienced in 2018.  (score=xxx)[source:1,2] |

**Dataset**

Our data is gathered manually from many different sources that could be liberal/conservative, reliable/unreliable. For instance, we select 4 topics e.g. Synagogue Shooting, Edward Snowden, … and for each topic, we collect 9-10 articles. The dataset is in CSV format which includes about 40 samples and 6 variables.

**id -** database id

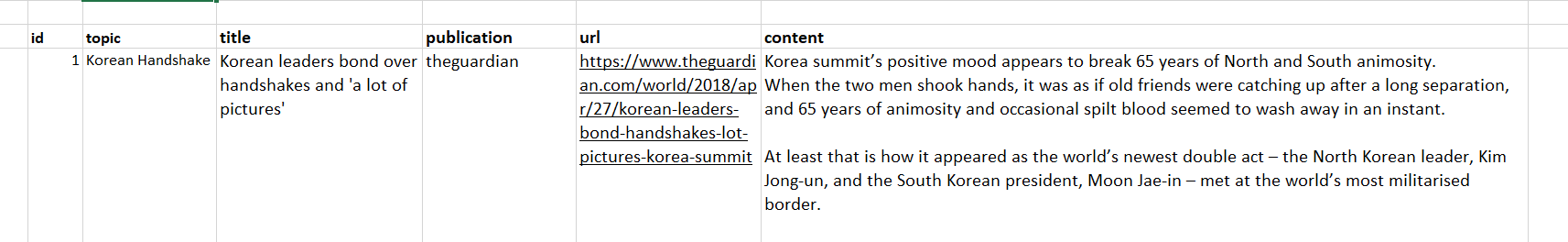
**topic** - topic of article

**title** - Article title

**publication** - publication name

**url -** url for article

**content** – article content

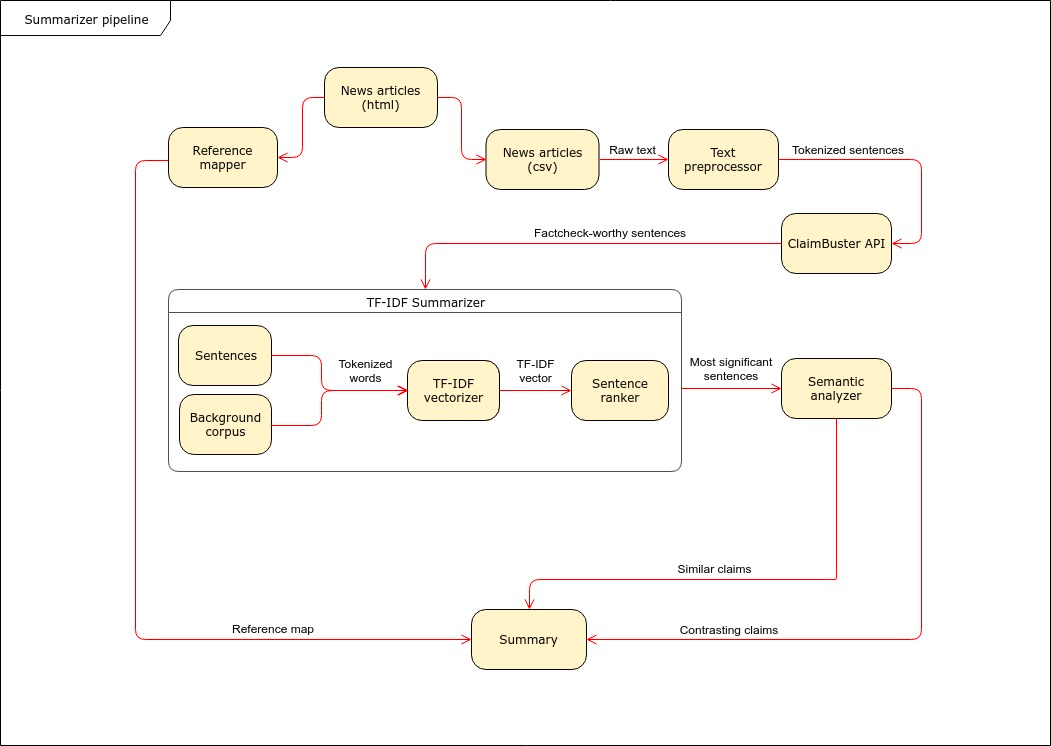


To go further, we might add 2 more columns in the dataset:

**reliability** - score of reliability of articles

**type** - liberal or conservative

**Pipeline sequence diagram**



**Bibliography**

[1] <http://ailab.ijs.si/dunja/SiKDD2008/Papers/Dali_Final.pdf>

[2] <http://ailab.ijs.si/tools/enrycher/>

[3] <http://render-project.eu/wp-content/uploads/2010/05/D2.2.1.pdf>