Examples

Whole bunch of things if you Google for it; not a lot of info in most cases as mostly companies/really simple:
https://www.mediaobservatory.com/ (Focused on event selection, has two papers!)
https://towardsdatascience.com/the-isolated-den-of-fox-news-f31126e605cd (based on shared citations - not really what we want)
https://www.adfontesmedia.com/ (Human judgement I think?)
https://onesub.io/ (Actually based on Bath, personalisation, no info on how it works)

There are also fact checkers, which is a related but different thing. Note that bias can be defined in different ways: political of course, but also truth vs not, opinion vs facts etc.

Steps

1. Web scraping - Direct website scraping or links from social media API. Have a dataset if you're more interested in other steps.
2. Extract title/date/author/text - template/heuristics. Keep simple.
3. Matching news articles about the same event - basic ML, simple features.
4. **Distance metric**
5. Visualisation - ideally a web interface. Graph layout algorithms, e.g. spring-mass systems / multidimensional scaling / trilateration. Showing how they evolve over time would be awesome, at both article and news entity levels.

Alternate approach: Scrape web/twitter etc. and look for link pairs - same person linking to two different new sources implies they are close (person reads both). Crude however, so not really the objective. Per-article is better!

Flaw with above: It looks at what is said, not at what is not said. Bias can also appear in article selection (the first link above is all about this, and uses recommender systems to model it).

Ideal would be to build a live website, but that would be very challenging.

Distance metrics

Many possible, here are some suggestions:

- Baselines: Overlap (intersection / union) of bag of words. Cosine distance between word count vectors. BLEU. These could all be weighted by tf-idf.
- Sentiment analysis. Every word in the article or selected words, e.g. part-of-speech to get adjectives. Thesaurus comparative - match words between articles that have the same meaning and do sentiment analysis on just those words to calculate a delta.

- Build per-publication language models (whole corpus) then rate each article by how similar it is to each publication (log probability of being generated by model) - this is then a coordinate system. Language models:
    - N-gram (letter or word)
    - A Hierarchical Bayesian Language Model based on Pitman-Yor Processes
    - TextRank
- Use low dimensional word vectors (dimensions potentially chosen via sentiment analysis) to generate a cloud in nD space. Do density estimate and then measure the distance between articles using earth mover distance. Can calculate using the sinkhorn distance technique.
- Topic models (e.g. LDA, LSA) can provide a distance in terms of their topic distribution vector. Doesn't have to be entire article:
    - Applied to emotive words only
    - Applied to adjective-noun pairs
    - Negative topic membership: Classify document, words that are worst described by the topic may be interesting from a bias point of view.
- Use open information extraction ("Identifying Relations for Open Information Extraction" is surprisingly easy to code) to obtain relation tuples. The words around them can then be of interest, and run through any of the above. Plus shared facts is a good indicator of news articles that are about the same event!

Related techniques

Part of speech and named entity recognition are necessary for many of the above. Can train model using Groningen Meaning Bank dataset, https://www.kaggle.com/abhinavwalia95/entity-annotated-corpus (version 4, but download without stupid features) Works best having applied word vectors (I like Glove, but many options) first and with Markov chain to clean up. Or just use spacy/nltk, which both have this built in!