State of the art + Risk Assessment of Hyperpartisan News Detection

Others' achievements, techniques and methodologies used, evaluation methods, results

News are our portal to the events happening in the world. As consumers of news, we have to distinguish between facts and fabrications, between objective truths and opinionated stances, between indisputable reality and unreliable falsehood. Everyday, we are prone to being influenced or downright manipulated by the more opinionated news or by those that align with our own thoughts and views of the world, confirming our biases and strengthening our prejudices. Thus, we need reliable means of telling whether an article is merely a fact-stating objective piece of news or if it has someone else's interest on its mind.

Our specific subject, "Hyperpartisan News Detection", is, as of right now, mostly unexplored territory. Even though the amount of Fake News detection algorithms boomed in the last couple of years, programmers and linguists alike are still dipping their toes in the more subtle domain of written opinion alignment. The progress made in other areas of Natural Language Processing, such as Sentiment Analysis or Stance Detection, will aid us greatly with our future task.

Sentiment Analysis refers to the use of natural language processing, text analysis and computational linguistics to systematically identify, extract, quantify, and study affective states and subjective information from a given text. In other words, sentiment analysis aims to determine the attitude of a speaker, writer, or other subject with respect to some topic or the overall contextual polarity or emotional reaction to a document, interaction, or event. The attitude may be a judgment or evaluation, affective state, or the intended emotional communication (the emotional effect intended by the author or interlocutor). Sentiment analysis is commonly applied to 'voice of the customer' materials such as reviews and survey responses or online and social media, but can be applied to opinionated or polarizing news articles as well.

A basic task in sentiment analysis is classifying the polarity of a given text at the document, sentence, or feature/aspect level - whether the expressed opinion in a document, a sentence or an entity feature is positive, negative, or neutral.

Advanced, "beyond polarity" sentiment classification looks, for instance, at emotional states such as "angry", "sad", and "happy". Our implementation will support “neutral” articles (from a political standpoint), as well as left and right-leaning ones.

Existing approaches to sentiment analysis can be grouped into three main categories: *knowledge-based techniques*, *statistical methods*, and *hybrid approaches*.

Knowledge-based techniques classify text by affect categories based on the presence of unambiguous affect words such as ‘happy’, ‘sad’, ‘afraid’, and ‘bored’. Some knowledge bases not only list obvious affect words, but also assign arbitrary words a probable "affinity" to particular emotions. Statistical methods leverage on elements from machine learning such as latent semantic analysis, support vector machines, "bag of words" and "Semantic Orientation - Pointwise Mutual Information". More sophisticated methods try to detect the holder of a sentiment (the person who maintains that affective state) and the target (the entity about which the affect is felt). To mine the opinion in context and get the feature about which the speaker has opined, the grammatical relationships of words are used. Grammatical dependency relations are obtained by deep parsing of the text. Hybrid approaches leverage on both machine learning and elements from knowledge representation such as ontologies and semantic networks in order to detect semantics that are expressed in a subtle manner, e.g., through the analysis of concepts that do not explicitly convey relevant information, but which are implicitly linked to other concepts that do so.

Stance detection has been defined as automatically detecting whether the author of a piece of text is in favor of the given target or against it.There are cases in which neither inference is likely.

The significant difference is that in sentiment analysis, systems determine whether a piece of text is positive, negative, or neutral. However, in stance detection, systems are to determine author’s favorability towards a given target and the target even may not be explicitly mentioned in the text. Moreover, the text may express positive opinion about an entity contained in the text, but one can also infer that the author is against the defined target (an entity or a topic). This makes the task more difficult compared to the sentiment analysis, but it can often bring complementary information.

The accuracy of a sentiment analysis or a stance detection system is, in principle, how well it agrees with human judgments. This is usually measured by variant measures based on precision and recall over the two target categories of negative and positive texts.

However, according to research human raters typically only agree about 80% of the time. Thus, a program which achieves 70% accuracy in classifying sentiment is doing nearly as well as humans, even though such accuracy may not sound impressive. If a program were "right" 100% of the time, humans would still disagree with it about 20% of the time, since they disagree that much about any answer. On the other hand, computer systems will make very different errors than human assessors, and thus the figures are not entirely comparable.

Important names in the field, research teams

- Noam Chomsky ("the father of modern linguistics")
- Jeffrey Elman (introduced the simple recurrent neural network)
- David G. Hays (published the first textbook in computational linguistics)
- Jerry Hobbs (is Fellow of the American Association for Artificial Intelligence)
- Aravind Joshi (co-founder and co-director of the Institute for Research in Cognitive Science)
- Ronald Kaplan (wrote the grammar for the LUNAR system, the first large-scale ATN grammar of English)
- Martin Kay (achievements: development of chart parsing and functional unification grammar and major contributions to the application of finite state automata in computational phonology and morphology)
- Henry Kučera (pioneer in the development of spell checking computer software)
- Sydney Lamb (the father of the relational network theory of language)
- James Pustejovsky (projects: Medstract, TimeML, ISO-Space)
- Luc Steels (pioneer in Artificial Intelligence who has made important contributions to expert systems, behaviour-based robotics, artificial life and evolutionary linguistic)

- Yorick Wilks (designed GATE, an advanced NLP architecture that has been widely distributed)
- Victor Yngve (the author of COMIT, the first string processing language)

Related articles and books

- Speech Acts: An Essay in the Philosophy of Language (speech act theory/ first act)
- From Argument Diagrams to Argumentation Mining in Texts: A Survey (argumentation theory/ second act)
- All Frames Are Not Created Equal: A Typology and Critical Analysis of Framing Effects(framing theory/ third act)
- Crossdomain mining of argumentative text through distant supervision (modeling argumentativeness in debate platforms)
- Deep Learning for Stance Detection in News
- I Couldn't Agree More: The Role of Conversational Structure in Agreement and Disagreement Detection in Online Discussions
- Patterns of Argumentation Strategies across Topics
- Framing: Toward Clarification of a Fractured Paradigm
- Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary
- A frame of mind: Using statistical models for detection of framing and agenda setting campaigns
- Behind the Article: Recognizing Dialog Acts in Wikipedia Talk Pages
- Classifying Frames at the Sentence Level in News Articles
- A News Editorial Corpus for Mining Argumentation Strategies

Relevant links

- http://deeplearning.net/
- https://eu.udacity.com/course/natural-language-processing-nanodegree--nd892
- https://medium.com/@ageitgey/natural-language-processing-is-fun-9a0bff37854e
- https://deeplearning4j.org/
- https://chainer.org/
- https://keras.io/
- https://stanfordnlp.github.io/CoreNLP/
- https://web.stanford.edu/class/cs224n/reports/2754942.pdf
- https://arxiv.org/ftp/arxiv/papers/1701/1701.00504.pdf
- https://www.nltk.org/

Resources and tools available

- TextBlob - a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.
    - TextBlob Documentation – Official documentation and quickstart guide.
    - Natural Language Basics with TextBlob - A short crash course into the basics of Natural Language Processing taught using the TextBlob library. Useful for those who are just starting out in the field of NLP

- spaCy - Marketed as an "industrial-strength" Python NLP library that's geared toward performance, its philosophy is to only present one algorithm (the best one) for each purpose. You don't have to make choices, and you can focus on being productive. It is also advertised as being easy to use with the AI ecosystem present on Python, which should prove useful if we decide to go that route.
    - spaCy Documentation – Official documentation and quickstart guide.
    - Complete Guide to spaCy - A short guide that teaches about the essential features of the spaCy library.

- PyTorch - Python-based scientific computing package targeted at those who are looking for a deep learning research platform that provides maximum flexibility and speed.
    - PyTorch Documentation - Official documentation
    - Deep learning for NLP with PyTorch - a tutorial that walks you through the key ideas of deep learning programming using Pytorch.