The Use of Natural Language Processing and Data Visualization for Ethics Pedagogy and Research

By

Mike Mitri

Professor Emeritus, James Madison University

Owner, Story Analyzer LLC

Abstract

This paper presents a software application called Story Analyzer (SA), and its use for ethical reasoning. SA uses natural language processing (NLP) and data visualization to produce dashboards of textual narratives. These visualizations depict narrative entities like people, groups/organizations, places, times, and ideas conveyed in text documents, and assist users to understand complex textual content. The paper shows how SA is used to facilitate ethical reasoning pedagogy as well as research involving ethical analysis.

Introduction: AI's promise (and threat) for Ethical Reasoning

Much has been said about the ethical dangers presented by artificial intelligence. AI has arguably reached the stage of passing the Turing test (Turing, 1950), which means an AI can convince you that it is human. In other words, it can fool you. Educators are rightly concerned about the implications of this advance in technology, including the ease of faking an essay by using ChatGPT. Other big ethically fraught societal issues include AI's effect on freedom, privacy, employment, distributive justice, political propaganda, war, etc. Underlying it all is the fact that AI is a human construction, and is thereby beset from human biases and stereotypes, which affect the predictions and recommendations made by algorithms that rely on machine learning.

Despite all its dangers, AI also brings exciting possibilities. Although passing the Turing test means that computers can fool people, it also means that computers can communicate with people in an engaging manner that they can relate to. Generative AI (like what's done by ChatGPT and Bing and Bard) allows a computer to talk like a human to a human.

But, can AI convey wisdom, or at least "common sense" (Levesque, 2018)? The internet is full of narratives and textual claims. Some of these are false and harmful, but others contain truths that can help us make better decisions. This is a goal of many universities as they try to foster the ethical reasoning skills of their students, including the effort at James Madison University (JMU). Separating truth from falsehood is essential for a moral world. Can AI help?

This paper presents a software application called Story Analyzer (SA), and its potential use for ethical analysis in pedagogy and in research. The paper starts with a description of SA, how it works, how it was developed, and what it has been used for.

After this background, the paper discusses SA's application to ethical pedagogy at JMU, followed by applications that can assist with research into ethics and philosophy in general. Hopefully, readers will attain a better understanding of how AI/NLP works and how it can be used to facilitate ethical reasoning in a complex sociotechnical world.

Throughout the paper are links to several SA dashboards, and readers are invited to explore these dashboards.

About Story Analyzer

Story Analyzer is a web application that uses natural language processing (NLP) and data visualization technologies to produce dashboards depicting themes, concepts, events, characters, and contexts of textual information (Mitri, 2022).

Story Analyzer uses the CoreNLP software library, created by Stanford University's NLP Group. CoreNLP provides many NLP services for processing and analyzing text documents, including:

- Breaking the text into individual sentences (sentence splitting)
- Tokenizing a sentence (breaking it into individual words)
- Identifying parts of speech (POS) within a sentence (nouns, verbs, adjectives, adverbs, etc.).
- Named entity recognition (NER) – recognizing names of people, places, organizations, dates/times, etc.
- Lemmas – the root form of each word (token)
- Constituency parsing – constructing taxonomies of noun phrases and verb phrases of a sentence.
- Dependency parsing – constructing the graph of linguistic relationships between pairs of terms in a sentence. Examples include adjectival modifiers, compound nouns, prepositions, and subject/ action/object relationships (De Marneffe and Manning, 2008).
- Co-reference resolution – finding all expressions that refer to the same entity in a text. Useful for matching pronouns to people or finding common themes in the text.
- Temporal tagging – recognizing and normalizing temporal expressions (e.g., "next Wednesday", "the previous summer", "July of next year", "Labor Day of 2023", "yesterday", etc.)

Story Analyzer performs two main tasks: information extraction and interactive visualization.

SA's *information extraction* algorithm makes extensive use of the above CoreNLP services. For narrative texts, involving people and groups performing actions that occur in space and time (e.g., news stories, legal documents), the most important services are POS, NER, dependency parsing, and coreference resolution. Dependency parsing is also critical for establishing the conceptual and linguistic structure of sentences in the text. The following sections will show how Story Analyzer uses CoreNLP services to extract narrative information, which is subsequently displayed in visualizations depicting the "what, where, when, who, and how" of a story.

The *interactive visualization* aspect of SA takes the information extracted from the NLP services and presents interactive dashboards allowing users to visualize the main elements of the narratives. This gives users the power to quickly navigate through dense text and come to a better understanding of what the narrative conveys.

These features are described and demonstrated in the following sections, culminating in the section of SA's application to ethical reasoning. We start with early work in SA, then describe scaling up to large text corpuses, then discuss SA's application to ethical reasoning.

iStory Analyzer Task	Software Used	Features and Services used by SA
Information Extraction	Stanford CoreNLP	Sentence splitting, tokenizing, POS, NER, lemmas, dependency parsing, coreference resolution, temporal parsing
Interactive Visualization	Data-driven documents (d3), Google Maps, and Google Visualizations	Libraries of visualizations (charts, graphs, calendars, timelines, treemaps, sunbursts, force graphs, word-cloud, maps, trees). User interactivity. Drill down and aggregation. Navigation and indexing.

Table 1: Software libraries used by Story Analyzer for the main SA tasks

Early Story Analyzer dashboards

Serious work on SA began after the 2016 presidential election, and early work concentrated on events in the Trump presidency, including official government documents such as the Mueller report, Horowitz report, and House impeachment report. These early dashboards can be seen at https://storyanalyzer.org/.

The documents depicted in these dashboards, as well as news articles and historical documents that are also available at the site, are largely *narrative* in form (hence the name "story analyzer"). Key elements of a story involve the actors (people or groups) and their interactions. These interactions occur in various contexts, including space, time, and other situations.

Figure 1 shows a dashboard of the House Impeachment report (section 1 executive summary) from 2019. The visualizations shown are the following:

- Two word-clouds, one containing the groups involved in the story and the other containing the people. Sizes of names depict their prominence in the story.
- A map showing where actions take place.
- A calendar showing when actions take place.
- An "interactions chord" diagram showing the actors and their interactions. Currently the focus is on interactions between Trump and Zelensky.

- The text shows the current focus, i.e., the sentences of interactions between Trump and Zelensky.

Story Analyzer is highly interactive. Users can focus on a place, time, person, or group to see relevant text from the story.

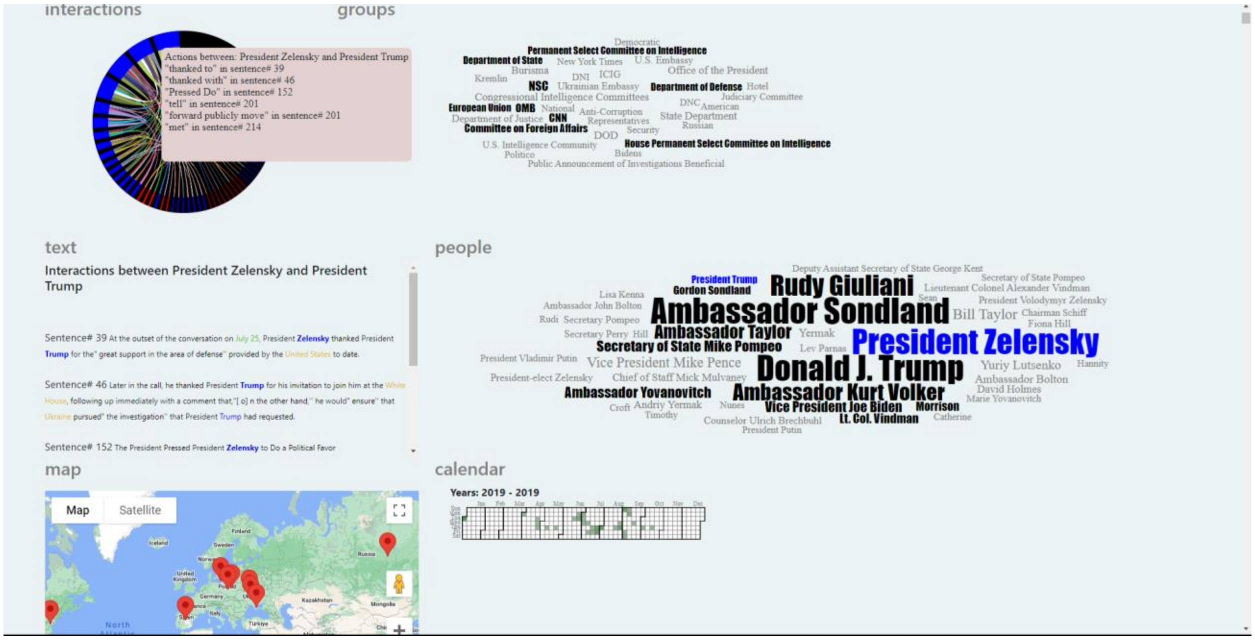

Figure 1: Story Analyzer dashboard depicting the Executive Summary of the House Impeachment report of 2019 (Section I)

A student project using Story Analyzer for analyzing ethical issues of AI and social media

SA was applied to JMU's Ethical Reasoning efforts in a JMU course called Business Intelligence, which is part of the Computer Information Systems major of the College of Business. This course delves into advanced database practices and includes use of AI. Although technical in nature, the course also considers governance and social impact factors of powerful information technologies.

The first use of SA for ethical reasoning took place in spring of 2020 during the Covid pandemic, when the course was held online. Students were asked to read an article regarding the Facebook and Cambridge Analytica scandal which occurred during the presidential election of 2016. The article, consisting of 9000 words, was captured in a SA dashboard that students could use to assist with their reading task. Then, the students were asked to analyze the story using JMU's eight key questions and post this analysis to an online discussion forum on JMU's Canvas learning management platform. Following this original post, students read and commented on each other's posts. This discussion was also captured by SA, resulting in a dashboard of the students' discussion posts. More recently, in spring semester 2023, students used the SA application to generate their own dashboards of stories they chose, which helped with beta-testing the software. They again participated in an online discussion, this one speaking more generally about ethical concerns of AI in the light of new advances including Chat GPT.

The use of SA described here demonstrates two potentials of AI in pedagogy. First, AI can be used to help students gain a deeper understanding of complex narratives than reading alone can do. Second, AI can be used to analyze the students' own ethical analyses of these narratives.

Story Analyzer for large text corpuses and social science research

Earlier work with SA was limited to relatively small text sources (not exceeding 10000 words). More recent research focuses on applying SA to larger text corpuses. This is possible when using multiple powerful processors in a cloud environment like Amazon's Web Services (AWS).

When the House January 6 commission released their report in early 2023, the full document consisting of 841 pages and over 360,000 words was submitted to SA for processing, resulting in this dashboard.

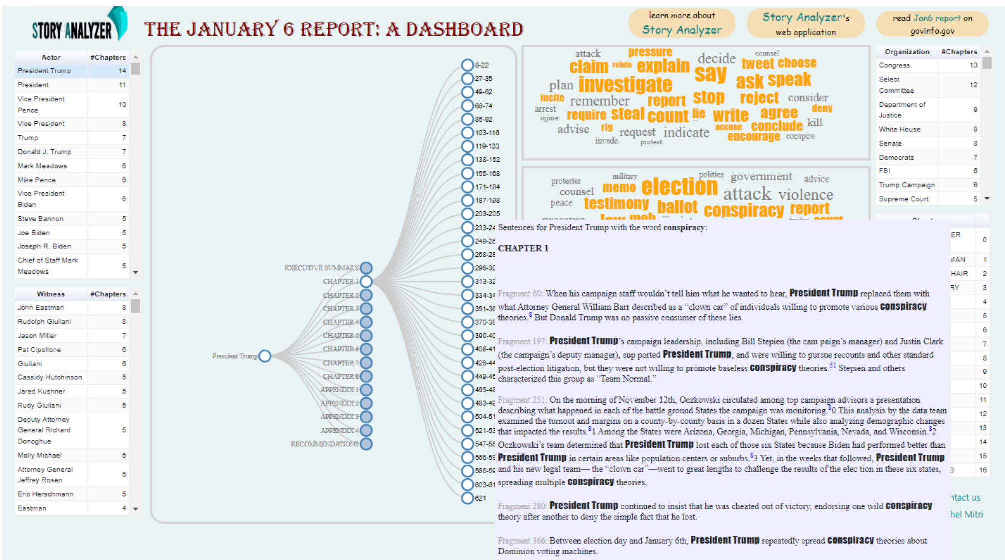

Figure 2: Story Analyzer dashboard of the January 6 Report

Another dashboard shows results of a 140-page memo from January 6 staffers who researched the actions (and inactions) of social media platforms in the weeks leading to the January 6 insurrection.

SA's recent application to research in political science concerns analysis of European Commission meetings going back to 1999 (Scherpereel et al, 2023). This involved minutes from the Prodi, Barroso I, Barroso II, Juncker, and von der Leyen (December 2019-June 2022) commissions, analyzing both structural features of college meetings and speech acts that occurred within those 978 meetings. SA extracted information about the meetings, and the actions/mentions of all EC members during the meetings. Actions by the commission as a whole, the commission's president, and the other members were captured. Significant actions were grouped along nine action categories: decide (14 keywords), deliberate (16), emphasize (21), express hope or fear (5), refer-back-to (19), request (5), speak (64), take-note-of (1), and validate (16). SA gathered this data and aggregated it across two dimensions: session (meeting) and actor (member). In addition to producing data for statistical analysis, the data visualization features of SA results in this dashboard. For each action category, users can drill down to see specific sentences in meeting minutes indicating actions in the category by meeting or actor. The minutes for these meetings total almost seven million words.

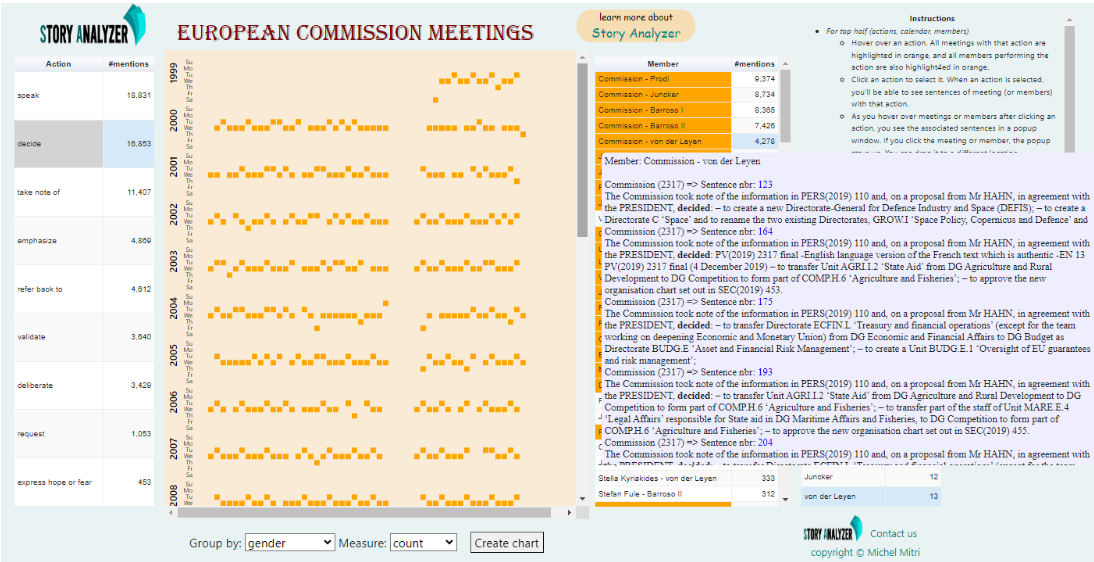

Figure 3: Story Analyzer dashboard for European Commission meeting minutes

Story Analyzer and the Stanford Encyclopedia of Philosophy

Perhaps more directly relevant for a conference on ethical reasoning is SA's recent application to the full corpus of Stanford Encyclopedia of Philosophy (SEP). After web scraping almost 1800 articles, totaling about 24 million words, SA is applied to various themes. Current dashboards include topics like truth, justice, political theory, and most recently ethics and morality. This section focuses on the ethics dashboard, which analyzes 94 SEP articles totaling about 1.5 million words.

The ethics dashboard focuses on 130 keywords and phrases. These were selected based on common terms in ethical/moral philosophy, and subsequently refined by discovering the frequencies of terms in the SEP corpus itself. The dashboard shows two word-clouds, one with the keywords and the other with the people (philosophers and authors) who are referenced in the article. A user can click a word in the keywords cloud which causes a "sunburst visualization" to display. This shows the linguistic dependencies between the selected keyword and other words in sentences of the corpus. Also, when a keyword is selected, all people associated with the keyword are highlighted.

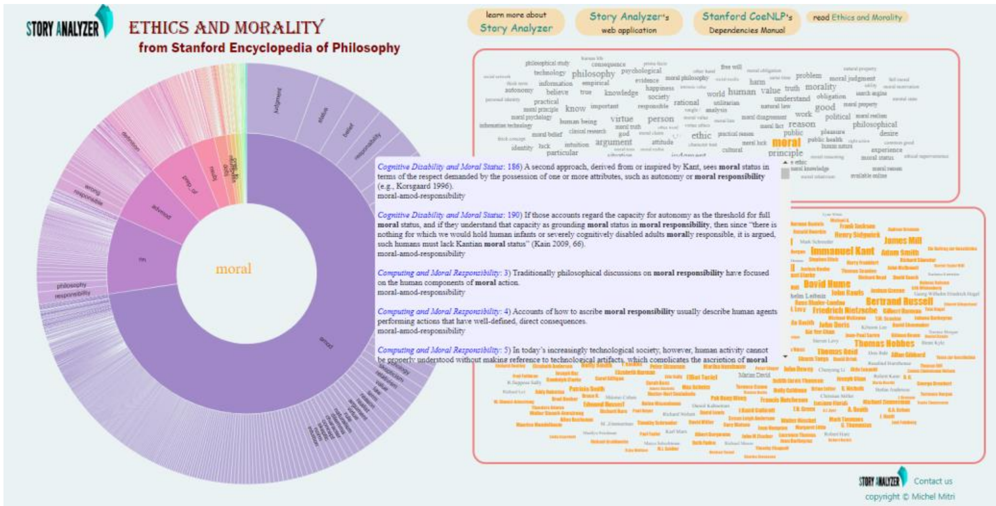

Figure 4: Dashboard showing ethics/morality articles from Stanford Encyclopedia of Philosophy

The circular sunburst visualization depicts the frequent linguistic relationships between a keyword and other words in sentences of the articles. Figure 4 shows some sentences from articles where the word "moral" is an adjectival modifier of the word "responsibility". Figure 5 shows some preposition relationships with the keyword "social media".

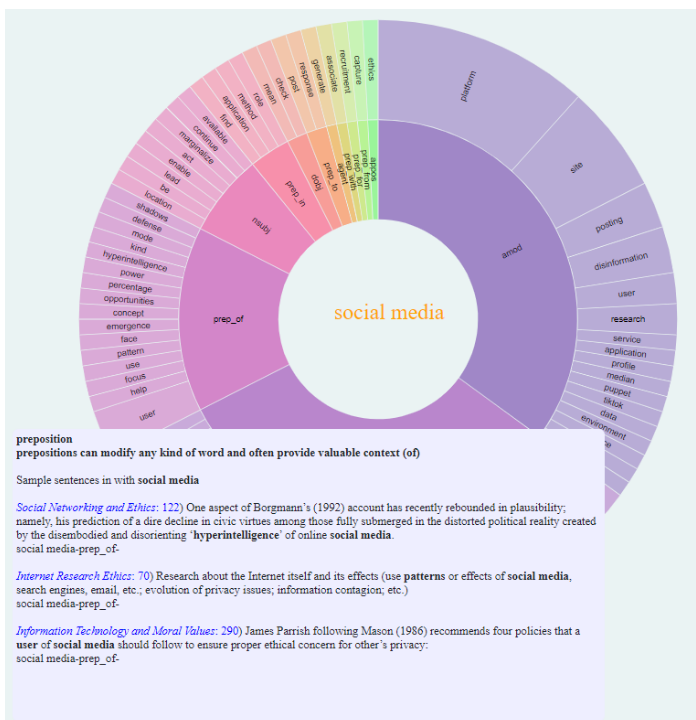

Figure 5: Some preposition relationships with "social media"

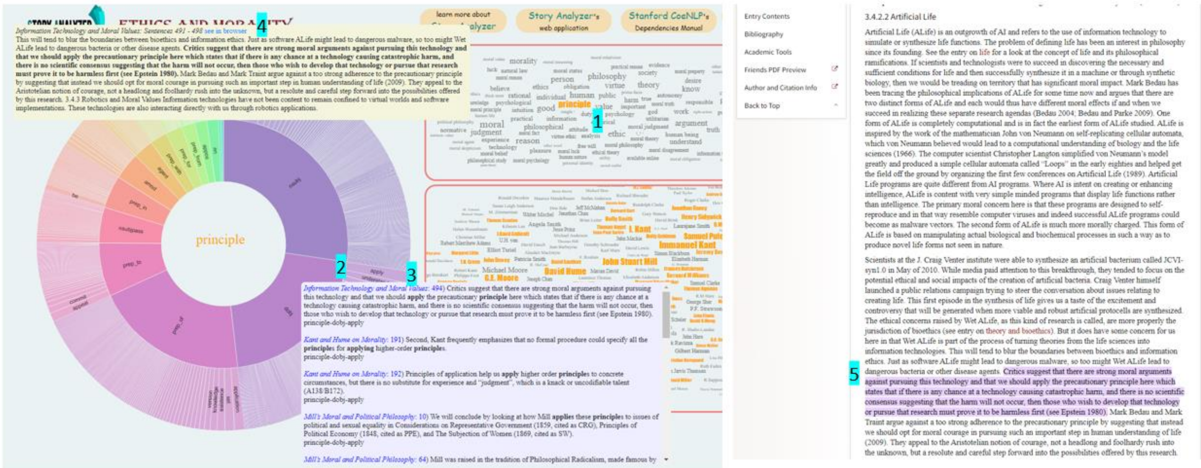

Figure 6: Interaction with the dashboard, navigating to the article's sentence in SEP

The steps numbered in Figure 6 show consecutive clicks. First, the user clicks the word "principle" in the keywords cloud, which produces the sunburst visualization centered on this word. Second, the user chooses the *dobj* (direct object) relationship to the word "apply" by clicking this word in the sunburst, which causes sentences from the corpus containing dobj relationships between "principle" and "apply" to display. Note that "apply" is a verb and "principle" is a noun. Here we see an example of a linguistic dependency showing a verb acting upon a noun. The sentences in the purple box are those retrieved from the corpus involving "apply" (and variants) acting on "principle" (and variants).

The blue italicized text of each sentence indicates the article and sentence number where the sentence can be found. When the mouse hovers over blue text, a yellow box appears expanding the text to show surrounding sentences. If the user clicks the blue italics text (step 3), the yellow box freezes in place. Note the link to the browser in the yellow box. The fourth click takes the user directly to the SEP page and more specifically the exact sentence of the article (step 5). In this case, the article is Information Technology and Moral Values (Sullins, 2018).

To unfreeze, click the yellow text box (if it is frozen), then the purple box.

Sentences can also be shown for highlighted names in the philosopher word-cloud regarding a selected keyword.

This is typical of the user interaction for all the SEP dashboards.

An explanation of CoreNLP's *dependency parsing (*De Marneffe and Manning, 2008) is in order here. Each sentence involves several binary linguistic dependency relationships between pairs of words. Figure 4 focuses on an adjectival modifier (*amod*) between the adjective "moral" and the noun "responsibility". Figure 5 illustrates prepositional dependencies, specifically "of" relationships (*prep-of*) between "social media" and the words "hyperintelligence", "pattern", and "user" in sentences from three different articles. And Figure 6 shows a direct-object (*dobj*) dependency where the verb "apply" operates on the noun (object) "principle". These are just a few of the linguistic dependencies whose identification can assist users to gain deeper understanding of complex text and enable refined navigation of large text corpuses like SEP. Hopefully this shows how useful the linguistic capabilities of NLP can be for researchers studying the analytical discourse in philosophy, including ethical and moral theory.

Discussion: limitations, differences, possibilities, and approaches

Previous sections discussed the services provided by CoreNLP. Sentence splitting, tokenizing, and POS tagging are relatively straightforward tasks and CoreNLP is very accurate with these functions. Named entity recognition and dependency parsing are highly useful but are considerably less accurate in NLP's current state of the art, with accuracy rates in the 80%+ range. Coreference resolution, which is vital for connecting related themes in a story, is far less accurate. 65% accuracy is considered a good score in the current NLP climate. These limitations impact the accuracy of Story Analyzer's results. Machine learning and data mining of natural language (unstructured data) is still considerably less accurate than that of categorical and quantitative (structured data). This caveat applies to AI in general, including the promising technologies of products like GPT, Bing, and Bard.

From the standpoint of functionality and purpose, there is some overlap, but also significant differences between Story Analyzer, CoreNLP, and ChatGPT. ChatGPT is an example of "generative AI". Generative AI is designed to produce content, it "generates". GPT (generative pretrained transformer) technology uses a complex neural network architecture operating on huge amounts of annotated text data to *create* something, be it an essay, a computer program, or an image. This is a classic example of *machine learning* (also called data mining or pattern recognition).

Machine learning (including use of neural networks) is also a key technique deployed in CoreNLP, but its services also include computational linguistics (rule-based) that applies theoretical knowledge of language structure. The output of CoreNLP is considerably different from ChatGPT, in that it produces structured data from the services (lists of tokens associated with POS/NER/lemma, lists of coreferences, etc.).

Story Analyzer does not do any machine learning, nor does it even do natural language processing. Instead, it is an application that *uses* the results of CoreNLP's output to extract relevant information and produce graphical visualizations. In this respect, it is an example of what application developers can do with the power that comes from AI products like CoreNLP and GPT. These products include APIs (application programming interfaces) that allow their services to be embedded in anyone's software application, so long as the coder knows how to use them.

SA (and to an extent CoreNLP) seems to have a different goal in mind than generative AIs like chatbots. ChatGPT generates. It acts like an author. Given a prompt, it generates an answer. It is, in this sense, a "writer". The goal behind SA is not to be a "writer", but rather to be a "reader". It attempts to absorb and structure textual narrative rather than creating new narrative. Understanding text is the goal with SA, not necessarily creating it.

This brings to mind the cautionary tale that Levesque (2018) says about the Turing test. Machine learning is very powerful, but it is all about finding patterns in data. It uses *inductive* reasoning, by drawing conclusions (or stereotypes) based on gleaning patterns in vast quantities of data points (observations). It doesn't incorporate *knowledge* or "common sense". The Turing test only tests whether a computer can convincingly imitate a human. This does not guarantee that the computer gets things right. Even ChatGPT makes plenty of recognizable mistakes. Note that the Turing test had several critics from the beginning. Searle (1980) and Dreyfus (1972) both found fault with the idea that computers can be truly intelligent.

Levesque suggests another test that he calls the "Winograd Schema Challenge". This test focuses on identifying what objects a pronoun refers to, which is often problematic for NLP systems (for example, CoreNLP's coreference resolution). Consider this sentence: "The trophy couldn't fit in the briefcase because it was too small." A human, after a little thought, can easily recognize that "it" refers to the briefcase. This is despite the fact that "briefcase" and "trophy" are very rarely used together in a sentence. There is no experience of patterns to rely on. But we all know that briefcases contain other things. And we know that in order to contain an object, a container must be bigger than the object. If the phrase was "too big", then we would know that the pronoun refers to "trophy". That's just common sense. This kind of knowledge in AI systems is what Levesque calls "good old-fashioned AI" (GOFAI).

In earlier periods of AI's history, it was largely a *knowledge representation* domain. The idea was to capture and formalize expert knowledge, represent it in a particular data structure, and apply a particular inferencing algorithm. A very common example is rule-based systems processed via *deductive* logic (Buchanan and Shortliffe, 1984). Note the difference between knowledge-based deduction from the inductive pattern-recognition of machine learning algorithms. The early promise of "expert systems" rested largely on this kind of GOFAI.

But these systems were very brittle and did not scale up well. By the late 1990s the hype of expert systems died down. Meanwhile, machine learning started showing great promise. Neural networks are very good at prediction, much better than knowledge-based systems. Therefore, currently AI is dominated by "pattern recognition", whereas "knowledge representation" tends to play second fiddle.

But nowadays, machine learning may be bumping up against a wall. Levesque argues that for AI efficacy to progress, we need more GOFAI. It will be interesting to see whether and how AI technologies will incorporate GOFAI in the future and how much improvement results.

Conclusion: future goals with Story Analyzer

One of the biggest dangers of AI and social media, especially in a politically divided society, is misinformation. There is such a saturation of narratives in the infosphere, many generated by bots instead of humans, that separating truth from falsehood is both very difficult and very important.

Before judging the truth of a narrative, it is important to recognize and extract the claims being made by the story. Some of this includes simple extraction of the narrative's elements (who, what, where, when, and how). Others are more nuanced. What exactly is the story trying to convince you of? Only after you can reliably understand a story's claim can you evaluate its truth and falsehood. AI research in claim detection is currently in a nascent stage (Caprio and Checco 2015, Habernal and Gurevych, 2016).

The next question is how to evaluate the truth value of a claim. The nature of *truth* is an active area of philosophical study. There are several schools of thought, as shown in Story Analyzer's SEP truth dashboard. The most commonly cited philosophical theories of truth are correspondence theory, coherence theory, and pragmatic theory. For each of these, it is possible to design computational implementations that at least attempt to approach the goals and principles embedded in the theory, thanks to advances in NLP.

Correspondence theory (Moore 1910, Russell 1912) asserts that a statement (or story) is true if it corresponds to facts. In computational terms, a story's claim can be compared against empirical data. If the claim can be gleaned from the text, then this claim can be fact-checked. There are a wealth of web archives containing results of empirical studies. Three examples are Google's Dataset Search, Our World in Data, and data.world. Consider also Google's fact-checker tool, which includes an API (application programming interface) that allows a computer program to interact with it in real time. Future work with SA will enable users to apply claims from stories to these sites/services in order to evaluate their truth in terms of "correspondence with facts".

Coherence theory (Joachim 1906, Bradley 1914) asserts that a statement or story is true if it coheres with other well-accepted beliefs. In computational terms, stories can be compared with other stories to see points of similarity and differences. If SA's information extraction process is applied to many stories on the internet, the structured information of any given narrative can be compared and measured against others. Fine-grained analysis and comparison of linguistic and grammatical data is possible, for example, via CoreNLP's dependency parser, and named entity recognition can compare object identifiers in different narratives to establish degrees of coherence. Future work with SA will include exploration of coherence approaches to truth-finding in textual narratives.

Pragmatism (Peirce 1878, James 1908) states that the truth of a statement hinges on its practical benefit. This is a largely utilitarian account of truth, which lends itself to *prescriptive analytics* (Dash et al, 2019) a decision science approach involving optimization methods like linear programming, decision trees, expert systems, and other computational approaches at evaluating options and "choosing the best solution". Integrating some of these approaches with NLP technologies can help to bring "actionable intelligence" to SA's services.

References

Bradley, F. (1914), Essays on Truth and Reality, Oxford: Clarendon Press.

Buchanan, B. G., and Shortliffe, E. H. (1984). Rule-based expert systems: The MYCIN experiments of the Stanford Heuristic Programming Project. Addison-Wesley.

Cabrio, E., Villata, S., and Checco, A. (2015), "Identifying and Classifying Claims in Online Political Debate". Proceedings of the 15th International Conference on Artificial Intelligence and Law (ICAIL 2015)

Dash, S., Sharda, R., & Chandrashekaran, M. (2019). Prescriptive analytics for management decision-making: A review. Journal of Business Research, 104, 352-365.

De Marneffe M-C, Manning CD. (2008) Stanford typed dependencies manual http://nlp.stanford.edu/software/dependencies_manual.pdf.

Dreyfus, Hubert (1972), What Computers Can't Do, New York: MIT Press, ISBN 978-0-06-090613-9

Habernal, I. and Gurevych, I. (2016), "Deep Argument Mining: Exploiting Multiple Levels of Linguistic Structure", Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)

James, William (1907) [1975], Pragmatism: A New Name for Some Old Ways of Thinking, New York and London: Longmans, Green & Co.; reprinted Cambridge, MA: Harvard University Press, 1975.

Joachim, H. (1906), "The Nature of Truth", Oxford, Clarendon Press.

Levesque, H. J. (1918),” Common Sense, the Turing Test, and the Quest for Real AI”, MIT Press, ISBN 9780262535205

Mitri, M. (2022), “Story Analysis Using Natural Language Processing and Interactive Dashboards”, Journal of Computer Information Systems (62:2), 216-226, DOI: 10.1080/08874417.2020.1774442

Moore, G. E. (1910), “Some Main Problems of Philosophy”, London: George Allen & Unwin, 1953.

Peirce, C.S. (1878) [1986], “How to Make Our Ideas Clear”, Popular Science Monthly, 12(January): 286–302; reprinted in Writings of Charles S. Peirce: A Chronological Edition (Volume 3: 1872–1878), C. Kloesel, M. Fisch, N. et al. (eds.), Bloomington, IN: Indiana University Press, pp. 257–276.

Russell, B. (1912), “Problems of Philosophy”, reprinted at Oxford: Oxford University Press 1971.

Scherpereel, J. A., Mitri, M. and Schreuders, A. (2023). "Von der Leyen's Gender-balanced Team: A New Way of Operating in the College of Commissioners?" Paper presented at workshop on The Governance of Gender Equality Policy in EU Institutions: Assessing Developments during the Von Der Leyen Commission Presidency (2019-2024). University of Tübingen.

Sullins, J. (2018), "Information Technology and Moral Values", The Stanford Encyclopedia of Philosophy (Summer 2023 Edition), Edward N. Zalta & Uri Nodelman (eds.), URL = <https://plato.stanford.edu/archives/sum2023/entries/it-moral-values/>.

Searle, John (1980), "Minds, Brains and Programs", Behavioral and Brain Sciences, 3 (3): 417–457, doi:10.1017/S0140525X00005756

Turing, A. M. (1950) “Computing Machinery and Intelligence”, Mind (49), 433-460