Agency plus automation: Designing artificial intelligence into interactive systems

Jeffrey Heer[a,1]

[a]Paul G. Allen School of Computer Science & Engineering, University of Washington, Seattle, WA 98195-2350

Edited by Ben Shneiderman, University of Maryland, College Park, MD, and accepted by Editorial Board Member Eva Tardos October 15, 2018 (received for review May 22, 2018)

Much contemporary rhetoric regards the prospects and pitfalls of using artificial intelligence techniques to automate an increasing range of tasks, especially those once considered the purview of people alone. These accounts are often wildly optimistic, understating outstanding challenges while turning a blind eye to the human labor that undergirds and sustains ostensibly "automated" services. This long-standing focus on purely automated methods unnecessarily cedes a promising design space: one in which computational assistance augments and enriches, rather than replaces, people's intellectual work. This tension between human agency and machine automation poses vital challenges for design and engineering. In this work, we consider the design of systems that enable rich, adaptive interaction between people and algorithms. We seek to balance the often-complementary strengths and weaknesses of each, while promoting human control and skillful action. We share case studies of interactive systems we have developed in three arenas—data wrangling, exploratory analysis, and natural language translation—that integrate proactive computational support into interactive systems. To improve outcomes and support learning by both people and machines, we describe the use of shared representations of tasks augmented with predictive models of human capabilities and actions. We conclude with a discussion of future prospects and scientific frontiers for intelligence augmentation research.

visualization | data science | artificial intelligence | human–computer interaction | automation

Although sharing overlapping origins in midcentury computer science, research programs in intelligence augmentation (IA; using computers to extend people's ability to process information and reason about complex problems) and artificial intelligence (AI; developing computational methods for perception, reasoning, and action) have to date charted largely separate trajectories. The former, beginning with the work of early luminaries such as Douglas Engelbart and Alan Kay—and continuing in the modern field of human–computer interaction (HCI)—led to now-ubiquitous technologies such as the personal computer, graphical user interfaces, the web browser, and touch-driven mobile devices. Meanwhile, AI has similarly contributed to numerous innovations, from spam filters, fraud detection, and medical diagnostics to recent developments in autonomous vehicles. Advances in the AI subfield of machine learning (ML), including the resurgence of neural network approaches trained on increasingly large datasets, has fed a frenzy of both research and industrial activity.

However, current rhetoric around AI often exhibits a myopic focus on fully-automated methods, for example, as a replacement for human labor. Certainly, people have their lapses. We do not "scale" to large repetitive tasks, we may commit errors in judgment due to insufficient information or cognitive biases, and we may have ambiguous intents that AI systems might help refine. Nevertheless, a sole focus on automation is concerning for multiple reasons. Automated approaches may mislead due to inappropriate assumptions or biased training data and may optimize fixed objectives that do not adapt in the face of changing circumstances or adversarial manipulation. In addition, users of such systems may come to overly rely on computational suggestions, leading to a potential loss of critical engagement and domain expertise. Real-world examples of such pitfalls range from flawed disease predictions (1) to preventable airplane crashes resulting from failed coordination among automated systems and the flight crew (2, 3). As these examples illustrate, safeguarding human agency is not only a value to uphold, but in many cases a necessity for effective and appropriate use.

This state of affairs has prompted prominent ML researchers to call for more "human-centered" approaches to AI. For example, Li (4) writes that a major goal of AI should be "enhancing us, not replacing us," while Jordan (5) notes that we "need well-thought-out interactions of humans and computers to solve our most pressing problems." AI experts are thus "rediscovering" HCI and recognizing the need to integrate AI and IA perspectives.

Although both exciting and overdue, we should note that the goal of integrating agency and automation is hardly new. In the early 1960s, Bar-Hillel (6) wrote (in the context of natural language translation): "The decisive problem becomes to determine the region of optimality in the continuum of possible divisions of labor" between people and computers. In the 1990s, HCI and AI researchers engaged in debate over direct manipulation vs. interface agents (7), arriving at a consensus in support of "increased automation that amplifies the productivity of users and gives them increased capabilities in carrying out their tasks, while preserving their sense of control and their responsibility." Meanwhile, research on creativity support tools (8) has explored nonanthropomorphic modalities for interacting with automated methods, including simultaneous interaction with multiple design alternatives (9) and integration of manual input and procedural programming (10).

Still, the need to productively couple agency and automation poses vital challenges for design and engineering. How might we effectively integrate automated reasoning into user-centered interactive systems? We examine one promising strategy: the design of shared representations of possible actions, enabling computational reasoning about people's tasks alongside interfaces with which people can review, select, revise, or dismiss algorithmic suggestions.

Rather than jumping right to the goal of imitating "human-level intelligence," we can take inspiration from a more humble

This paper results from the Arthur M. Sackler Colloquium of the National Academy of Sciences, "Creativity and Collaboration: Revisiting Cybernetic Serendipity," held March 13–14, 2018, at the National Academy of Sciences in Washington, DC. The complete program and video recordings of most presentations are available on the NAS website at www.nasonline.org/Cybernetic_Serendipity.

Author contributions: J.H. designed research, performed research, contributed new reagents/analytic tools, analyzed data, and wrote the paper.

Conflict of interest statement: This article includes discussion of two academic research projects that were subsequently commercialized as start-up companies: Trifacta Inc. and Lilt Inc. J.H. has financial interests in both of these companies.

This article is a PNAS Direct Submission. B.S. is a guest editor invited by the Editorial Board.

[1]Email: jheer@uw.edu.

Published online February 4, 2019.

starting point: augmentation within everyday interactions, so tightly integrated into our work that we take it for granted. Consider spelling- and grammar-checking routines included within word processors, which unobtrusively highlight problematic spans of text to help authors discover and correct writing errors (Fig. 1). Or, consider autocompletion of text input, as when issuing queries to an Internet search engine: Automatic, yet easily dismissible, suggestions can accelerate search and refine ambiguous intents (Fig. 2).

While modest in scope, these examples exhibit a number of useful principles for structuring interactions between people and computational assistants. First, they provide significant value, promoting efficiency, correctness, and consideration of alternate possibilities that a user might not have otherwise considered. Second, they augment, but do not replace, user interaction: These aids blend into the interactive experience in a nondisruptive manner and can be directly invoked or dismissed. Third, these augmentations require neither perfect accuracy nor exhaustive modeling of the user's task to be useful (e.g., consider the subtask of "spelling" vs. the more general goal of "writing"). Fourth, through interaction, both people and machines can incrementally learn and adapt (e.g., autocomplete may alert a search-engine user to relevant related topics, while other user queries may train the system for future recommendations). These and other principles for "elegant coupling of automated services with direct manipulation" are articulated in Eric Horvitz's *Principles of Mixed-Initiative User Interfaces* (11)—required reading for anyone interested in the intersection of AI and HCI.

In this work, we review approaches to reconciling agency and automation—and realizing the principles above—through the judicious codesign of user interfaces and enabling algorithms. Our goal is to productively employ AI methods while ensuring that people remain in control: unconstrained in pursuing complex goals and exercising domain expertise. We will consider three case studies, drawn from the areas of data wrangling, data visualization for exploratory analysis, and natural language translation. Across each, we examine the strategy of designing shared representations that augment interactive systems with predictive models of users' capabilities and potential actions, surfaced via interaction mechanisms that enable user review and revision. These models enable automated reasoning about tasks in a human-centered fashion and can adapt over time by observing and learning from user behavior. We conclude with a discussion of future challenges and opportunities.

Data Wrangling

Data analysts spend much of their time (sometimes 80% or more) cleaning and formatting data to make it suitable for analysis (12). As a result, domain experts often expend more effort manipulating data than they do exercising their specialty, while less technical users may be excluded from working with data. This "data-wrangling" work is fundamentally a programming task: Users must determine a sequence of transformations of input data, mapping it into a more usable and actionable form.

In the Data Wrangler project (13, 14), we explored means for users to create data-transformation scripts within a visual, direct manipulation interface augmented by predictive models (Fig. 3). We initially explored interface gestures for expressing transformations, but this led to ambiguity, as the same gesture might be used for different transformations. For example, if you select a span of text, do you intend to extract text, replace text, or split up the text? However, we realized that for a range of simple interactions (such as row, column, and text selection), only a limited number of transformations typically apply. We also observed that while most users had a sense of the transforms they wanted to perform, they may not know the precise names or semantics.

Human beings face ever more complex and urgent problems, and there effectiveness in dealing with these problems is a matter that is critcal to the stability and progress of society. Humans are effective in part because they make use of efficeint tools, methods, and strategies.

Fig. 1. Word processor with spelling and grammar checking annotations: a simple yet ubiquitous example of IA. (Text adapted from Douglas Engelbart's 1961 proposal *Program On Human Effectiveness*; ref. 29.)

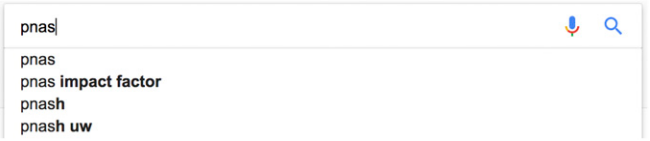

Fig. 2. Internet search query autocompletion. In addition to showing various completions, the recommendations suggest refinements of user intent, such as retrieving the impact factor of *PNAS*.

These insights led us to an approach we call predictive interaction (15), analogous to autocomplete, in which simple selections guide automatic predictions of which transformations to apply. The central idea is to imbue software with domain-specific models of user tasks, which in turn power predictive methods to suggest a variety of possible actions. As users select columns, rows, or cell contents within a table grid, Wrangler reasons about the space of possible actions compatible with the user's selection and provides "autocomplete" suggestions. Suggested actions are visualized to convey their effect on the data, facilitating human assessment of recommended actions. The result is an altered form of interaction: While users can still directly author transformations (e.g., using command menus or writing code), they can also engage in a guide–decide loop, in which initial interactions provide evidence of the user's intent, leading to suggestions that a user can then accept, refine, or dismiss. We bootstrapped our initial recommender system from a set of heuristic rules, then observed user actions over time to collect training data and adaptively improve the suggestions.

To reason about potential actions, we designed Wrangle, a high-level domain-specific language (DSL) for data transformation. Wrangle is a specialized programming language that expresses the necessary operations for cleaning, aggregating, and integrating data. The language also serves as a formal model for reasoning about user actions: We can search over the space of language statements to enumerate and recommend potential transforms. Put another way, the language provides a shared representation through which people and algorithms can work in tandem to achieve a shared objective (here, properly transform data). The use of a DSL also provides an abstraction layer between client-side concerns (reasoning about actions and visualizing results) and server-side concerns (taking the produced transformation script and compiling it to run efficiently across a computing cluster).

In initial user studies (13), we found multiple benefits for the predictive interaction approach, including increased productivity (faster completion times) and users' discovery of useful transforms in response to initially ambiguous intents. We also experimented with "proactive" suggestions (14) that use measures of data table quality to automatically suggest transforms without need of an initiating user interaction. We found that many users expressed a desire to maintain control and roundly ignored suggestions presented before interaction with the table grid. However, users did not express similar concerns when proactive suggestions were inserted among "reactive" suggestions specific to an initiating interaction, even though they were actually unrelated to the user's input. User assessment of suggestions appeared to improve when users saw themselves as the initiators.

We have subsequently commercialized this line of work by founding Trifacta, a provider of interactive data-transformation

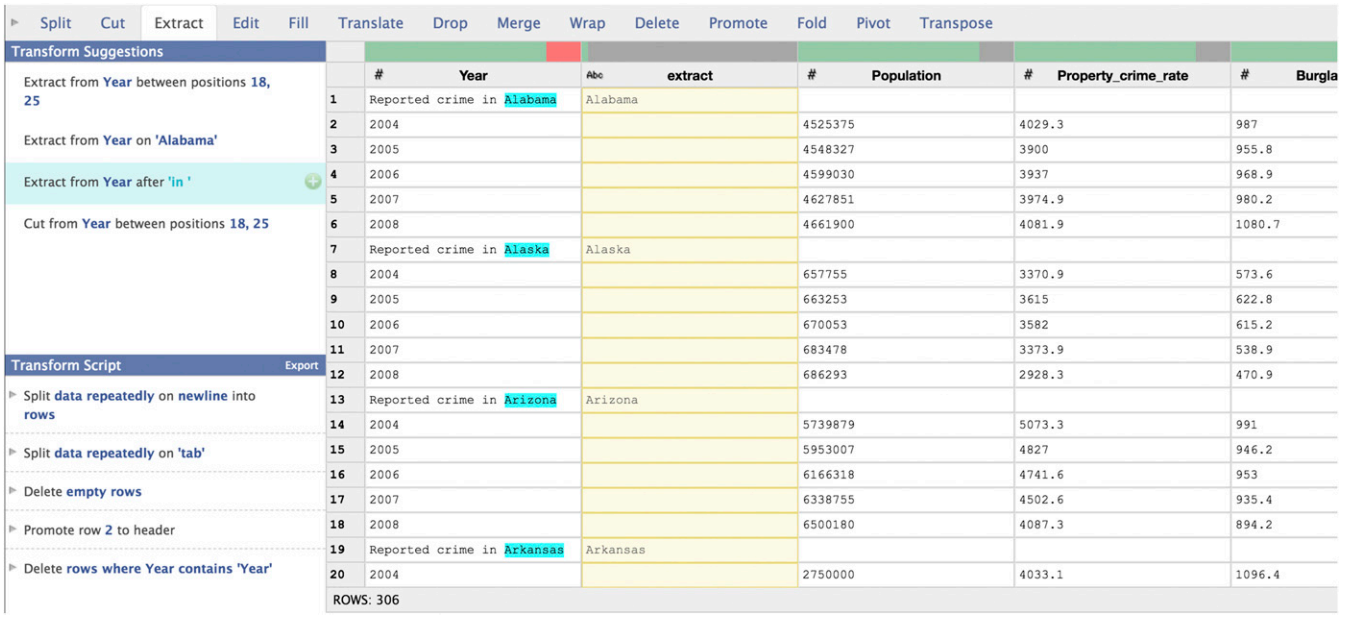

Fig. 3. The Data Wrangler interface for data cleaning and transformation, including a table grid (*Right*) and a transcript of executed transforms (*Lower Left*). Here, a user simply selects the text "Alabama" in the first row. The system responds with automatic transform suggestions (*Left*), such as extracting or cutting text from table cells. The selected transform extracts text after the span "in." Visual previews in the table grid visualize the effect of applying the transform.

tools. Real-world customer experiences have informed additional design iterations. In particular, we have made significant investments not only in improving ML performance, but also in creating structured graphical editors for Wrangle statements. Among other benefits, these editors help users edit "close, but not perfect" suggestions to fit their objectives—providing an improved shared representation for review and revision of automated recommendations.

Data Visualization for Exploratory Analysis

Once a dataset has been suitably cleaned and formatted, it can serve as input to further exploration and modeling. Exploratory data analysis (EDA) involves inspecting one's data—typically by using visualization techniques—to assess data quality and ensure that modeling assumptions are met, or to develop and refine hypotheses that might then be tested in future studies. The process of EDA can be highly iterative, involving both open-ended exploration and targeted question answering.

Making data-transformation and visualization-design decisions while exploring unfamiliar data are nontrivial. Analysts may lack exposure to the shape and structure of their data or begin with vague analysis goals. Decisions of which data to inspect and how to properly visualize it often depend on both domain and visualization design expertise and require a tedious specification process that impedes exploration. Moreover, analysts may exhibit premature fixation on a hypothesis of interest, overlooking data-quality issues or potentially confounding factors.

Voyager (Fig. 4) is an interactive system for exploratory analysis that blends manual and automated chart specification to help analysts engage in both open-ended exploration and targeted question answering (16, 17). Upon initially viewing a dataset, Voyager populates a gallery of univariate summaries for each column in a data table, encouraging more systematic consideration of data fields and potential quality or coverage issues. Users can also manually construct charts using drag-and-drop interactions, mapping selected data fields to visual encoding channels such as x position, y position, color, size, and shape.

As users engage in exploration, Voyager recommends related views based on the current focus chart. The recommended charts can include summary aggregations (for instance-level charts), deaggregrated views (for aggregate plots), alternative visual encodings, and plots incorporating an additional data field. Rather than perform rampant data mining that may unearth any number of (possibly spurious) correlations among data fields, Voyager seeds its search from the current focus view. The recommendations constitute a "search frontier" sensitive to a user's current focus and goals, with automated support parameterized by observations of user behavior.

Akin to Data Wrangler, Voyager uses a formal language—in this case the Vega-Lite visualization grammar (18)—as a shared representation for reasoning about the space of chart designs. User drag-and-drop actions in the interface map directly to Vega-Lite chart descriptions. To generate related view recommendations, Voyager also employs effectiveness rankings distilled from human perceptual studies. Voyager applies logic programming methods to search a space of related charts and "solve" for the best recommendations according to a knowledge base of design guidelines learned from perception experiments (17, 19). This underlying model is adaptive: In response to user actions or new perception experiments, we can retrain our model of preference weights to generate refined chart design recommendations (19).

Both Voyager and Vega-Lite are available as open-source projects and have been adopted within the Jupyter notebook and JupyterLab data science tools. Across multiple user studies, we have found that judicious suggestions (i.e., those that promote both data diversity and perceptually effective charts) can lead to more systematic coverage of a dataset during early-stage exploration. The inclusion of automated support shifted user behavior: Without related views, participants engaged in more targeted "depth-first" search patterns through the space of visualizations, whereas the inclusion of recommendations biased users toward increased consideration of alternative charts and additional data field combinations (17). Recommended related views also accounted for a significant proportion of the charts bookmarked as "interesting" by study participants.

Participants reported that Voyager helped to promote data-quality assessment and combat confirmation bias due to fixation

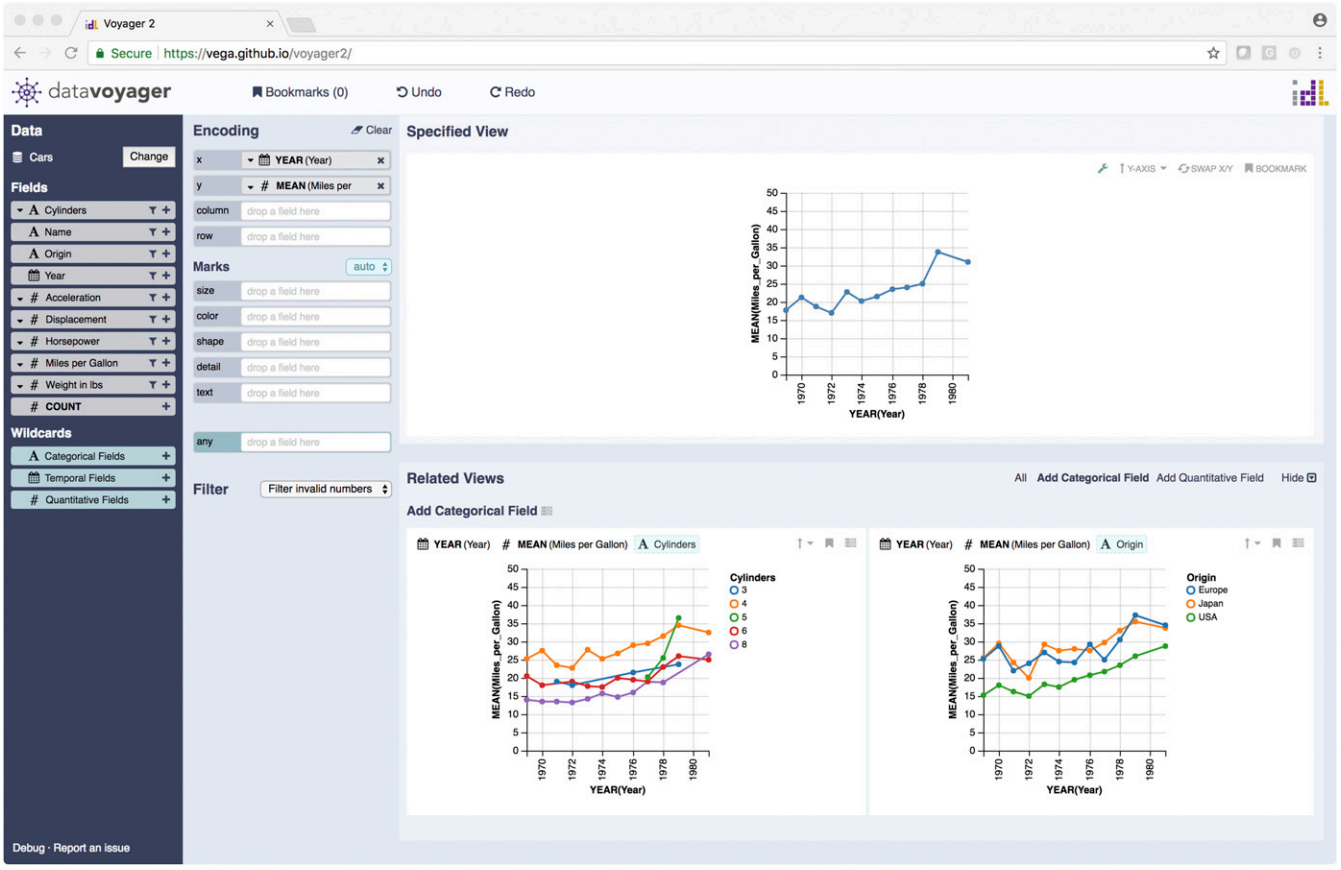

Fig. 4. The Voyager system for exploratory visual analysis. Here, an analyst views a chart showing the average fuel efficiency of cars over multiple years (*Upper*). The system automatically searches over the space of visualizations to suggest related views (*Lower*), such as subdividing the data by a categorical field (cylinder count or region of origin) to provide additional insight.

on specific relationships (e.g., "the related view suggestion function in Voyager accelerates exploration a lot"). Participants also stated that recommended views helped them learn how to use the tool (e.g., "I feel more confident using Voyager. It helped me to learn"). However, some also noted a shift in their behavior as a result of the recommendations, with one user reporting: "These related views are so good but it's also spoiling that I start thinking less. I'm not sure if that's really a good thing." As we will discuss, these reflections point to potential design trade-offs and suggest a need for more comprehensive evaluations of interactive systems that incorporate automation.

Natural Language Translation

Unlike the more complex tasks of data wrangling or exploratory visual analysis, our initial motivating examples—spell checking and query autocomplete—involve a basic shared representation for both input and output: digital text. For those cases, recommendations can be achieved largely via lookup tables or frequency statistics. Although involving similar input/output domains, a much more difficult task—and a holy grail of AI research—is natural language translation. Language translation is a multibillion dollar industry, critical to the dissemination of science, policy, and culture among the people of the world.

AI researchers have long sought the goal of fully automatic high-quality translation, but with limited success (20). Machine translation (MT) is a valuable tool for gisting, but despite impressive progress in recent years, human translators remain the gold standard for accurate, nuanced, and culturally appropriate work. As early as the 1950s, researchers envisioned interactive systems that might balance the strength of manual and automatic approaches to natural language translation (6, 20), assisting human translators while also providing valuable training data for automated systems. However, that vision sat largely unrealized, partially due to the computing field's focus on fully-automated methods, but no doubt also due to insufficient accuracy by past translation systems: For a long time, MTs were more hindrance than help, providing insufficient value and making the computer a rather poor collaborator.

We sought to revisit the vision of interactive translation in the context of today's more accurate MT engines. In the predictive translation memory (PTM) project (21, 22), we investigated approaches to mixed-initiative language translation, seeking a virtuous cycle in which machines might automate rote tasks (improving efficiency) while human professionals provide nuanced translation guidance (improving quality). The PTM interface (Fig. 5) consists of an augmented text editor, in which freeform text is the primary shared representation. Text input is first initialized with a full MT. To make edits, users can either begin typing or hover over words to view alternative suggestions. Upon editing, PTM automatically retranslates parts of the text not yet touched by the user, taking user edits into account. These user edits are also used to make incremental training updates to the underlying translation model, enabling domain adaptation that can improve MT quality, even in the midst of a session.

The interaction design of the PTM system (21) required great care. Drawing on Horvitz's (11) principles, we sought to introduce aids by which MT systems could provide assistance, while avoiding alterations that might disrupt the fluency of text entry. These aids include per-word translation lookups upon mouse hover [allowing efficient direct invocation and termination

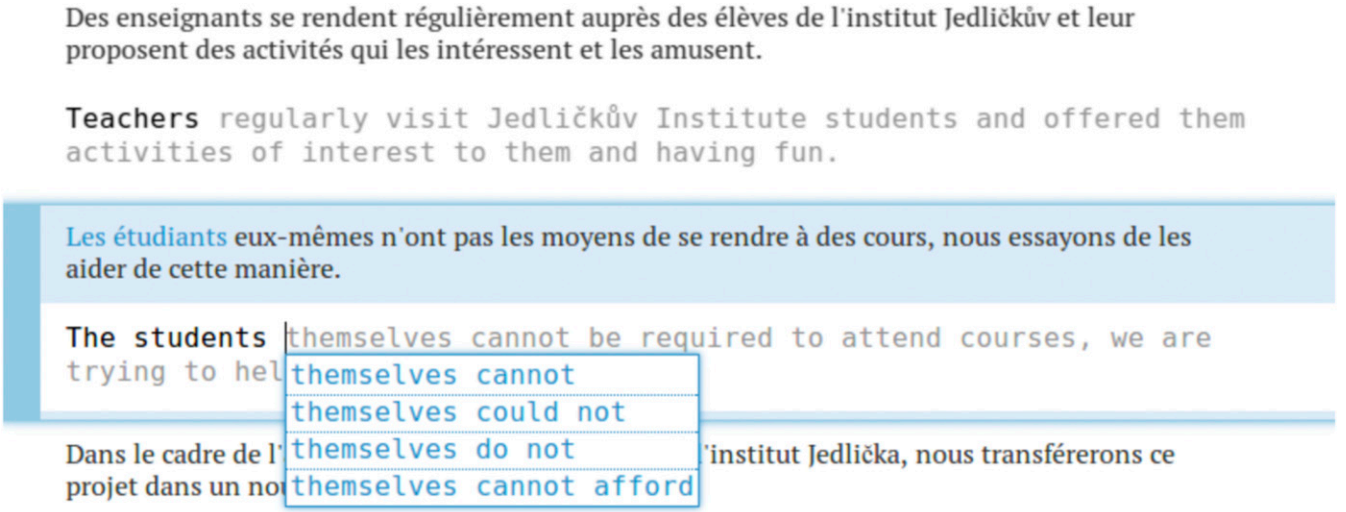

Fig. 5. The PTM tool for natural language translation. The interface shows the document context, with source inputs (here, French) interleaved with suggested target translations (here, English). Target translations are seeded via MT, but are user editable. The current focus sentence is indicated by a blue rectangle, with translated source words shaded. As a translator types, the machine suggests update in real time to reflect the changed context. Translator edits are also used to tune the translation engine to provide in situ domain adaptation.

(11)], visualizing translation coverage of the source text (maintaining working memory of recent interactions), and real-time updates to predicted translation completions in response to a user's partial translation (providing mechanisms for efficient interaction to refine results). Early user tests were critical for identifying and removing augmentations that proved distracting or overwhelming, such as visualizing raw source-target word alignments rather than just source coverage. We also tuned the translation engine to provide updates within 300 ms, as higher latencies led to user reports of a "sluggish" interface in which updates were out of sync with the user's translation process.

To assess our approach, we conducted a series of experiments involving professional translators for a number of target languages (e.g., Arabic, French, and German), translating texts from a variety of genres. As a baseline, we first compared purely manual translation with postediting (23), in which a human translator is given MT output and can revise (or delete) it as they see fit. Using ratings by human judges, we found that postediting led to reduced time and improved quality over purely manual translation, suggesting that translation technology has advanced sufficiently to provide value for high-quality translation tasks.

We then conducted a study comparing postediting and PTM. Interactive translation with PTM resulted in higher-quality translations (according to the BLEU+1 metric), with >99% of characters being entered via interactive aids (21). PTM was slightly slower than postediting, but this effect diminished over time as users became more accustomed to the new interface. Moreover, retuning the MT engine on the more fine-grained interactive PTM inputs led to significantly greater improvements than with postediting, resulting in fine-grained corrections to MT output (22). This research on PTM is now being commercialized by Lilt, a company providing adaptive language-translation services.

Despite these benefits, some study participants reported (not unlike Voyager) that the inclusion of automation affected their behavior and perceived agency. One participant felt that with MT support, "the translator is less susceptible to be creative," while another stated that MT "distracts from my own original translation process by putting words in [my] head." Indeed, we observed that translators using MT-powered aids produced more consistent translations, primed by MT output. These observations suggest trade-offs to examine in future evaluations: Individual translators may experience a mix of productive assistance and "short-circuiting" of thought; meanwhile, customers who hire a team of translators might prefer having the more consistent results.

Designing Shared Representations

The above projects investigate ways of integrating agency and automation by incorporating predictive models and feedback into interactive applications, in each case showing benefits in terms of user productivity, quality of outcome, and/or enhanced learning. By leveraging shared representations that can be authored and edited—ranging from simple text to specifications of data transformations or visualizations—people and algorithms can both contribute to, and adaptively learn from, solutions to shared problems. This approach sits in opposition to a perspective of pure automation, suggesting one avenue by which AI methods can be applied to helpfully reshape, rather than replace, human labor.

For each example application, careful interface design and evaluation was required to craft an effective "collaborative" interface. That said, the notion of collaboration used here differs in critical ways from the one common to social software for people, such as shared document editing. To safeguard human agency, the interaction models in Wrangler, Voyager, and PTM are fundamentally asymmetric: Automated methods suggest possible actions, which are then displayed for review and revision by the user, who remains the ultimate decision maker.

The example applications above also illustrate different points in the design space of shared representations. In Wrangler and Voyager, the shared representation involves three components: (*i*) a DSL that represents potential user actions, (*ii*) a predictive model that performs search and recommendation over the space of DSL statements, and (*iii*) an interface that presents language statements and recommendations in an interactive graphical form. A DSL provides a shared medium in which both people and machines reason about and formulate actions. These underlying DSLs (Wrangle and Vega-Lite, respectively) were manually designed by experts with extensive domain knowledge, with ML techniques subsequently applied to provide recommendations. The graphical interface maps the formal language to a more usable form via visual representations of DSL statements

and recommendations, which are then amenable to user review and action.

PTM, on the other hand, does not expose an underlying task representation. Users directly interact with text and may accept or modify spans suggested by the MT engine. Of course, both the human translator and MT engine possess rich internal representations: The translator brings their language experience and world knowledge, while the MT system comprises a model trained on a large corpus of language data. However, exposing an internal representation such as a parse tree seems unlikely to aid translation work if human translators do not consciously reason in those terms. In addition, while the original PTM system used a traditional beam search algorithm, later versions were revised to better-performing neural network approaches. Due to the text-only shared representation, a complete overhaul of the MT model did not require substantial changes to the interactive aids. Determining the right level of discourse for a shared representation—from explicit task models to direct editing of output—critically depends on the task, data, and user mental models involved.

In addition to demonstrating real-world utility, commercial and open-source use has led us to expand both the automated and interactive aspects of these systems. ML improvements have led to increased prediction accuracy while also expanding coverage to more tasks, such as inference of input data types in Wrangler. At the same time, updated interfaces for graphical review, authoring, and revision have proven critical to improve usability and enhance the interactive guide–decide loop. In general, we have observed that users want more control, not less, requiring us to develop more fine-grained interactive specification tools. Overall, we have found that neither automated suggestions nor direct manipulation play a strictly dominant role. Rather, we have observed that a fluent interleaving of both modalities can enable more productive, yet flexible, work.

Future Prospects

How might we better construct and deploy systems that integrate agency and automation via shared representations? One open research area is to develop design tools that aid the creation of effective AI-infused interactive systems. While the case studies above are suggestive, each involves a bespoke effort backed by multiple years of research and development. Future user-interface toolkits might change the cost structure of prototyping, development, and evaluation by supporting not only input and output, but by including task-modeling support combined with inference, learning, monitoring, and model-management services. While the examples above focus primarily on single-user scenarios, social systems involving teams of people as well as AI algorithms are an important concern. Another critical component of such future tools is ongoing monitoring and model analysis: As interactive systems are imbued with AI and ML techniques, designers, developers, and potentially even end users will require methods with which to inspect models, perform error analyses, and track how models change over time.

The shared representations in the case studies above, which range from text-editing interfaces to DSLs, were engineered by people. It is possible that shared representations might also be learned—whether in whole or in part—by using data-driven methods. For example, techniques such as variational autoencoders, generative adversarial networks, and vector-space word embeddings involve the construction of latent spaces of reduced dimensionality from input data, often in an (indirect) attempt to capture useful "semantic" properties of the data. Examples include "semantic arithmetic" in word vector spaces (24) (e.g., finding vector offsets that capture pluralization or changes of gender) and capturing typographic design principles (25) (e.g., attribute vectors that capture the notion of a bold typeface, including subtle changes such as preserving internal letter space and baseline while increasing stroke width). Scientific research in fields as diverse as astronomy (26) and genomics (27) have used similar approaches for unsupervised feature learning to leverage an abundance of unlabeled data, potentially capturing novel scientific insights. New approaches might better map such spaces, identify useful representations contained therein (28), and also potentially constrain them to capture known features of interest. A related issue is to properly account for and adjust for bias due to limited or unrepresentative input data. In short, how might people not only work with algorithms via a priori shared representations, but construct new representations in a data-driven manner?

Going forward, a critical question is how to evaluate systems across varied levels of agency and automation, with the ultimate goal of building up a corresponding engineering discipline. For a given application, might we predict an appropriate balance of agency vs. automation based on the nature of the task, such as the presence of fixed, shifting, or open-ended objectives? In what cases will we wish to give AI methods greater discretion of action than in the examples considered here, and how will these methods interpretably justify those actions? The case studies above include evaluations involving task time and quality measures, as well as more qualitative concerns regarding participants' perceived autonomy and creativity of action. These studies demonstrate people's preference to see themselves as the initiators of critical actions and to avoid interruptions as they formulate their intent. By what methods should we further assess and compare such systems, not only at the point of initial design, but throughout the deployment life cycle?

While some evaluation criteria will necessarily be problem-specific, some critical dimensions will be shared across applications, particularly those relevant to human experience and learning. For example, how might shared representations help promote interpretability, learning, and skill acquisition by more novice users, using AI methods to level up—rather than deskill—workers? To what degree will such systems promote behavior characterized by user control vs. more passive acceptance of algorithmic recommendations? Can appropriate designs prompt more critical engagement in the face of automated support? In some cases, should we accept having the computer "think for us," if it frees us to engage in higher-level reasoning or more creative practice? The examples above suggest that these questions do not lend themselves to simple formulations, but rather a space of design trade-offs ripe for further study.

ACKNOWLEDGMENTS. I thank Jessica Hullman, Arvind Satyanarayan, Dan Weld, and the anonymous reviewers for their helpful feedback. This article discusses collaborative research projects conducted with Anushka Anand, Riley Chang, Jason Chuang, Spence Green, Philip Guo, Joe Hellerstein, Bill Howe, Sean Kandel, Jock Mackinlay, Chris Manning, Dominik Moritz, Felix Ouk, Andreas Paepcke, Zening Qu, Sebastian Schuster, Sida Wang, and Kanit Wongsuphasawat.

1. Lazer D, Kennedy R, King G, Vespignani A (2014) Big data. The parable of Google Flu: Traps in big data analysis. *Science* 343:1203–1205.
2. Stewart J (2017) Don't freak over Boeing's self-flying plane—Robots already run the skies. *Wired*. Available at https://www.wired.com/story/boeing-autonomous-plane-autopilot/. Accessed June 9, 2017.
3. Glanz J, Suhartono M, Beech H (2018) In Indonesia Lion Air crash, black box data reveal pilots' struggle to regain control. *The New York Times*. Available at https://www.nytimes.com/2018/11/27/world/asia/indonesia-lion-air-crash-.html. Accessed November 29, 2018.
4. Li F-F (2018) How to make A.I. that's good for people. *The New York Times*. Available at https://www.nytimes.com/2018/03/07/opinion/artificial-intelligence-human.html. Accessed March 7, 2018.
5. Jordan MI (2018) Artificial intelligence—The revolution hasn't happened yet. Available at https://medium.com/@mijordan3/artificial-intelligence-the-revolution-hasnt-happened-yet-5e1d5812e1e7. Accessed May 22, 2018.
6. Bar-Hillel Y (1960) The present status of automatic translation of languages. *Adv Comput* 1:91–163.
7. Shneiderman B, Maes P (1997) Direct manipulation vs. interface agents. *Interactions* 4:42–61.

8. Shneiderman B (2007) Creativity support tools: Accelerating discovery and innovation. *Commun ACM* 50:20–32.
9. Terry M, Mynatt ED, Nakakoji K, Yamamoto Y (2004) Variation in element and action: Supporting simultaneous development of alternative solutions. *Proceedings of the ACM Human Factors in Computing Systems (CHI)* (Association for Computing Machinery, New York), pp 711–718.
10. Jacobs J, Brandt J, Mech R, Resnick M (2018) Extending manual drawing practices with artist-centric programming tools. *Proceedings of the ACM Human Factors in Computing Systems (CHI)* (Association for Computing Machinery, New York), vol 590, pp 1–13.
11. Horvitz E (1999) Principles of mixed-initiative user interfaces. *Proceedings of the ACM Human Factors in Computing Systems (CHI)* (Association for Computing Machinery, New York), pp 159–166.
12. Kandel S, Paepcke A, Hellerstein JM, Heer J (2012) Enterprise data analysis and visualization: An interview study. *IEEE Trans Vis Comput Graph* 18:2917–2926.
13. Kandel S, Paepcke A, Hellerstein JM, Heer J (2011) Wrangler: Interactive visual specification of data transformation scripts. *Proceedings of the ACM Human Factors in Computing Systems (CHI)* (Association for Computing Machinery, New York), pp 3363–3372.
14. Guo PJ, Kandel S, Hellerstein JM, Heer J (2011) Proactive wrangling: Mixed-initiative end-user programming of data transformation scripts. *Proceedings of the ACM User Interface Software & Technology (UIST)* (Association for Computing Machinery, New York), pp 65–74.
15. Heer J, Hellerstein JM, Kandel S (2015) Predictive interaction for data transformation. *Proceedings of the Seventh Biennial Conference on Innovative Data Systems Research (CIDR)*. Available at cidrdb.org/cidr2015/Papers/CIDR15_Paper27.pdf. Accessed November 29, 2018.
16. Wongsuphasawat K, et al. (2016) Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE Trans Vis Comput Graph* 22:649–658.
17. Wongsuphasawat K, et al. (2017) Voyager 2: Augmenting visual analysis with partial view specifications. *Proceedings of the ACM Human Factors in Computing Systems (CHI)* (Association for Computing Machinery, New York), pp 2648–2659.
18. Satyanarayan A, Moritz D, Wongsuphasawat K, Heer J (2017) Vega-lite: A grammar of interactive graphics. *IEEE Trans Vis Comput Graph* 23:341–350.
19. Moritz D, et al. (August 20, 2018) Formalizing visualization design knowledge as constraints: Actionable and extensible models in Draco. *IEEE Trans Vis Comput Graph*, 10.1109/TVCG.2018.2865240.
20. Green S, Heer J, Manning CD (2015) Natural language translation at the intersection of AI and HCI. *Commun ACM* 58:46–53.
21. Green S, Chuang J, Heer J, Manning CD (2014) Predictive translation memory: A mixed-initiative system for human language translation. *Proceedings of the ACM User Interface Software and Technology (UIST)* (Association for Computing Machinery, New York), pp 177–187.
22. Green S, et al. (2014) Human effort and machine learnability in computer aided translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Association for Computational Linguistics, Stroudsburg, PA), pp 1225–1236.
23. Green S, Heer J, Manning CD (2013) The efficacy of human post-editing for language translation. *Proceedings of the ACM Human Factors in Computing Systems (CHI)* (Association for Computing Machinery, New York), pp 439–448.
24. Pennington J, Socher R, Manning CD (2014) GloVe: Global vectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Association for Computational Linguistics, Stroudsburg, PA), pp 1532–1543.
25. Carter S, Nielsen M (2017) Using artificial intelligence to augment human intelligence. *Distill*. Available at https://distill.pub/2017/aia/. Accessed May 22, 2018.
26. Naul B, Bloom JS, Pérez F, van der Walt S (2018) A recurrent neural network for classification of unevenly sampled variable stars. *Nature Astron* 2:151–155.
27. Way GP, Greene CS (2018) Extracting a biologically relevant latent space from cancer transcriptomes with variational autoencoders. *Pac Symp Biocomput* 23:80–91.
28. Olah C (2015) Visualizing representations: Deep learning and human beings. Available at colah.github.io/posts/2015-01-Visualizing-Representations/. Accessed May 22, 2018.
29. Engelbart DC (1961) Program on human effectiveness. Available at web.stanford.edu/dept/SUL/library/extra4/sloan/mousesite/Archive/Post68/PrHumanEffectiveness.html. Accessed November 29, 2018.