# technology_hiring_admissions_2026

https://medium.com/@GeoffreyGordonAshbrook/technology-hiring-admissions-f97f6212b4c2

Technology in Hiring & Admissions

On The State of, and Applications of Technology to,
Hiring and Job Seeking Around The Year 2025

2026.01.14-22, Geoffrey Gordon Ashbrook

# Introduction

To better understand the technical needs of hiring based on the evolving situation of hiring over time, let's start by likening hiring to an area that is more frequently encountered by most people: Journalism. Hiring practices and news and journalism are (as of 2026) still struggling to take form in the age of the internet. 2026 is exactly 30 years since the 1996 expansion of the world wide web. Some areas of the world have adapted to the internet and computer technology better than others, if after a transition period (music sales, for example, famously underwent an arguably painful transition before new models were developed).

Both journalism and hiring are areas that were 'disrupted' by the emergence of the world wide web internet in the late 1990's (earlier aspects of networks and internets aside), and are both areas where technology can realistically greatly help. Yet in both cases after more than a quarter century (and after thirty years of internet technology development) neither have evolved into clear, and clearly maintainable, forms that are surviving and thriving with the internet.

Many local newspapers, and TV news networks in general, are (as of 2026) either already closed or 'dying on the vine' (a term I think I first heard to describe the music CD shop industry). Journalism as a professional research and reporting discipline has largely been replaced as pipeline of information publishing by more entertaining non-journalistic infotainment clickbait (or 'rage bait') online that is too often in the form of extreme and radicalizing content (yes, that is a very terse oversimplification of debates around publishing-quality that have been heated since the Gutenberg press around 1440).

Hiring, before the internet, often followed a pattern where jobs were posted in news-papers to be applied for in-person or starting the process over the telephone (with a person). A few things have changed since then.

While technology will hopefully greatly support both journalism and hiring practices at some future time, both appear (in 2026) to be in a state more recognizable as a tragedy of the commons, where mass and crowd behaviors combine with a lack of infrastructure and education, so that most people suffer continual loss in cycles of non-ideal-behavior and distorted perceptions that maintain a lock on this undesirable equilibrium.

We will explore issues facing hiring and needed-solutions, examined through several contexts, including:
1. High level areas: Who needs what?
2. Prioritizing issues that can be most influenced (or controlled)
3. Immediate-Simpler Needs vs. Nuanced Future Needs (to help prioritize aiming for simpler goals first)
4. A case study of a full-stack software build and deployment for a hiring agency.
5. Some factors around trying to apply "AI" solutions

# High Level Areas: Who needs what?

From the perspective of HR or employers, and from the perspective of job applicants, what new tools, and or what features and functionalities, are needed?

## From an HR or Employer Perspective:

1. Tools or capacity to handle a quantity of applications that is too large for any feasible human staff to coordinate a manual inspection. (E.g. If you receive tens or hundreds of thousands of applications for one position, how is a reasonably-sized staff supposed to search, match, and filter that large number down to something more like the ten most favorable applicants to look at in more detail and interact with (e.g. interview)?)

2. Tools or capacity to fuzzy-compare varyingly-worded job-application ~files with a database of fields (e.g. the names of job titles and skills (note: a supplementary ability might be somehow having applicants clarify specific most-important items in a structure-data format)

3. Providing feedback for professional development: informing rejected applicants about what they are perceived to need, so that next-time the applicants can either gain those qualifications or better clarify their application materials (e.g. if they do already have a given qualification).

4. Helping to make a job description in a reasonable amount of time (which may involve coordination between departments). This is something that hiring agencies that I have worked with in the past have expressed: hiring managers and HR departments do not have the time, resources, and sufficient tools to produce clear and meaningful job description documentation (or to organize and decide on the plans and details that are to go into that documentation).

5. Helping to post a job in a reasonable amount of time: Even if you have all the materials describing a job, the formalities of publishing and publishing (e.g. to a variety of job posting locations) is onerous.

6. Helping to collect and filter applications in a reasonable amount of time (for example, the initial rough filtering for generally appropriate applicants).

7. Job 'Project Area' definition: https://github.com/lineality/project_areas_for_project_and_product_management

8. Decision management: How various parties coordinate decisions about the job specifications

9. Facilitating questions and tests in applications: Having information beyond a standard resume can be very useful when finding the few applicants you are looking for in an overwhelming haystack of applications.

10. Creating clear, coherent, accurate, job descriptions.

11. Incentivizing the applicant (for it to be legitimately worth their time to) put more time and helpful detail into the application. The employer wants to get useful, accurate, high-quality information from the applicant. But each applicant needs to fill out hundreds to thousands of applications and time simply does not exist for more than seconds or minutes to be spent on a given application. This is part of the lose-lose tragedy of the current situation. What features, such as guaranteed feedback, of the design and format of the application process, could make higher quality application data sustainable?

## From a Job-Applicant Perspective:

1. Creating applications that are accessible and possible to complete in a realistic amount of time:

Rules of thumb for Technical Job searches:

- A technical job applicant needs to submit generally between 500 and 1000 job applications before getting a job, from which they can expect to receive absolutely no information about why the rejected applications were rejected.

- A job post that is more than four days old has already been spammed with so many applications that it is not worth applying; after N thousand applications they stop looking at new ones, and the limit is usually reached in a few days (sometimes in a few minutes).

- Because job descriptions are so frequently incoherent and wrong, if there is anything remotely connected to your resume you should apply to the job. Do not pass over a job post simply because you do not meet one or more so-called "requirements."

- You can expect a delay of several weeks before hearing back.

- At the risk of being vague, never say anything negative about anything or anyone, ever.

Let's say there are roughly 20 Mon-Friday days in a month that a job applicant can spend job-searching. Let's say the applicant can spend four-six hours each day doing nothing but filling out applications (without even a minute pause), and assuming two months of job searching to stay-fresh in the hiring pool.
20 days (per month) * 2 months * 4-6 hours a day, to fill out 1000 applications = 9.6 to 14.4 minutes per application. Note, this is most likely an over-estimate of how much time can really be spent filling out an application. Most job applicants do not have a secretary who continually hands them applications to fill out for six hours straight.

What needs to happen in this (generously estimated) 10-15 minutes?

1. go to where the job is posted
2. create various searches for what job to find
3. check the basic location/type of the job
4. read the often several pages of job description

5. go to the application-site (usually involving creating a new account and login involving emails and passwords and confirmation numbers emailed, etc)
6. manually entering in the same basic name, address, materials-upload information
7. answering the standard gender, race, sub-race, military background, disability background questions
8. often re-entering numerous redundant resume/linked-in fields from scratch (not uncommonly, for your entire background and resume)
9. It is recommended to make at least one outreach attempt per application (e.g. someone doing hiring in the company, which is usually not easy to find in linkedin (let alone on the internet generally)

+

And all that is before the applicant says anything in particular about that job:

1. Why they want the job or some give message to the hiring manager
2. Specific requirement confirmation (e.g. do they really live in X country)
3. Write a cover-letter
4. Take an automated test
5. Do an online automated interview

## A Word Matching Problem

A submitted resume should be (must be) tailored to each job, based on not only specific job requirements but on specific terms and phrases (e.g. that will be used to filter out unqualified people). How long does it take to make a completely new and microscopically re-phrased resume?

The issue of term-matching may be an especially important point and topic here. One of the key bottlenecks that exists in this flood of online job descriptions and applications is the use of exact match term filters. A classic headache for businesses is that haphazard and varying text strings filled into a field that must contain something exact (such as an id, an title, a category, an address etc.) is useless information that cannot be processed. Inputs for this type of field cannot be any string the user wants to put in; it must match the fixed options in the company's database (there are several terms used to describe this type of data entry form field, such as "Dropdown," "Select Field," "Selection Box," "Choice Field," "List Field", etc. This is why you may have noticed an increase in the number of online forms where there must be an exact-match or

option-selection (such as mailing address input). So, back to job applications, unless the applicant in their resume manages to somehow luckily put exactly the right characters in exactly the right order in exactly the right expected places in the resume, their resume (which may be a 100% match for the job to a human eye) will be rejected by the word-match-filters.
This is perhaps the key context for the rise of paid services, such as by Linkedin, to help with the very time-consuming and tedious task of using the exact wording of the job description to re-word your resume for just that specific job. Unless the resume makes it through the black-box magic word filter, no one will ever look at it. (In theory, this situation could be fixed or improved by better communication about what is being requested and what is being sent, and possibly a different format, as companies know they are turning away good candidates. But with tens of thousands of spam emails to filter out, the status quo is a low-bar muddle-through meh.

It would not be too difficult for the job application to include a standardized questionnaire with binary inputs for each requirement and wish-list item (like are seen sometimes on Linkedin-Easy-Apply forms). Instead, typo-filled job descriptions flood the internet and never-looked-at resumes spam their way back through the internet with zero feedback, zero improvement, and significant cost and loss incurred by all parties. Somewhere there is probably a golden business optimization opportunity buried in here (as with those who found a profitable way to sell music on the internet), and some cost-effective service will become a better standard. But after thirty years (to date), crickets.

People can disagree about the exact numbers (and surely those do vary depending on each business and admissions context) but the overall pattern and trend is the same. There is a disconnect between a kind of pre-internet expectation that a person can and should spend hours, days, or weeks preparing a professional application including research into the company, a specific relevant resume, a custom cover-letter, work history, work-sample, letters-of-recommendation for that specific job, specific interview preparation, etc., and a post-internet reality where the amount of time they can realistically allocate to each application is significantly less (often around three to ten minutes).

2. Getting feedback for professional development (same as described above)

3. **Required vs. Preferred:** Interpreting the often unclear parts of job descriptions:
There is a fundamental area of confusion around what skills are required that wastes time on both sides. Employers say they strictly require all possible qualifications when that is not true; some things they require, other things are merely good to have. Employees (having no idea what is actually required) say that they have all possible qualifications when that is not true. In this tragedy of the commons it becomes infeasible for either side to know what the other side is really trying to match to.

### Simple vs. Advanced Tools & Features
There are various ways of breaking down and categorizing (e.g. for prioritization and build scheduling) what different types of goals there are. One useful context may be separately focusing on features that the most immediate simple and concrete needs (as in 'putting out fires' type problem solving), on the one hand vs. longer term, bigger picture, and more nuanced stretch-goals on the other hand. This may also help to identify what types of technologies and techstacks maybe needed for different features and functionalities.

Examples of feature-areas for solving the most common, basic, workflow choke-points may include:
- Time spent to create a job-post
- Time/accuracy of applying for a job
- Time/accuracy to filter applications
- Time/accuracy of late stages including interview
- Reducing large numbers of applicants
- Hybrid data-search

(For each of these, defining a goal and measuring if you have the goal should be reasonably straight forward.)

Areas for stretch-goals that beyond the bare minimum may include:
1. Communication (Skills & Facilitation) Tools
2. Decision Making (Skills & Facilitation) Tools
3. Project (Skills & Facilitation) Tools including Project Areas Definition & Alignment
4. System-Health (Skills & Facilitation) and Tools including Sustainability and System-Collapse Prevention
5. Developing and retaining talent
6. Supporting productive definitions, methods, practices, and policies
7. "Project Areas" include points that often should be covered at least in interviews (in the interested of, and at the initiative of,

both parties) if not mentioned earlier in job descriptions and application materials, these include (paraphrase for this context):
- Process (e.g. workflow type, policies, procedures)
- Schedules (e.g. starting, stoping)
- Users/Stakeholders (e.g. Who are they?)
- Features, Needs & Goals (e.g. What are they?)
- Prioritizing and Goal/Target/Milestore Selection (e.g. What is the first Minimum Viable Product to build?)
- Feedback, Communication, Learning (e.g. using stakeholder feedback and identifying and acquiring needed skills)

Here it will probably take more work to define a set of goals, user-stories, and suites of features and subfeatures, and more time should be spent working with the users and stakeholders throughout development (for example for iterative feedback and UI testing).

See:
https://github.com/lineality/project_areas_for_project_and_product_management

## Prioritizing what HR Can (vs. cannot) Affect

As hiring issues are in some ways a large scale 'commons' problem, some of the issues that make hiring burdensome are not directly in the control of HR/Employer.

Aspects that a particular hiring-specialist can and cannot 'control':

### Can Affect/Control
- Project-area definition emphasis
- Clarifying, selecting, distinguishing, what you really require them to have (across specific work and technical skills, as well as project-management and team-communication skills), what is a luxury for them to have, what they can realistically be taught, and (realistically or defensively) what they are likely to need to be taught
- Best-practice feedback
- Feasibly-realistic time-duration of initial application: seconds-minutes
- Better and more efficient focus on projects, articles, during in-person conversations
- Efficient online exams for screening
- Be realistic about what you can and should teach the new employee

- 'Long term' security & maintainability of your infrastructure (to some degree): having ever-less technical debt, and building maintainable assets
- HR vs. Hiring Manager/Team conflicts or miscommunications

### **Cannot Affect/Control**
- National/international formatting standards: The simple fact that there is no national, international, or other standard for CV/resume formatting results in instant spaghettification of both role and person description information to systematic reading.
- National/international lack of feedback
- National/international lack of support from academia
- General software and IT industry 'long term' issues (e.g. security, maintainability, and profitability, long term data storage, etc.)
- Issues with the internet itself
- Schools do not teach practical skills (e.g. e.g. students with a masters in information science who cannot open a file)
- Schools do not teach communication skills
- Schools do not teach coordination skills
- Schools do not teach STEM literacy
- Schools do not teach project skills

From a project/product management perspective, the 'project' for which the prospective employee is being hired is often (as is routine) in a state of undefined, indeterminate, mush, leading to arbitrary, imaginary, vague, incoherently-evolving, and self-contradictory questions and requirements.

## **Count-Limiting & Standardizing Skill Areas:**
1. quantity of skills (and fixed list-sizes
2. Skill Identification: fixed vs. string-matching vs. fuzzy/vector-search

Linked-in only allows 50 skill areas. In some occupations that may be more than enough, but in IT that very quickly fills up and becomes a bottleneck. In the proliferation of languages, platforms, certifications, services, software packages, tech-stacks, technologies, cloud providers, etc. not to mention soft-skills, disciplines, business skills, STEM skills, etc., there is no way pick a set of 50 linkedin skills that have any chance of matching the diverse and continually changing job descriptions that come and go.

## Requiring 'human' oversight of sometimes tens of thousands of applications is not feasible.

The inevitable result of being so overloaded with applications with no adequate system to navigate and manage the volume of data is that the employer is arbitrarily discarding the most qualified people in favor of randomly selected inferior candidates, where the process has no feedback or mechanisms for improvement. While this is happening globally affecting not only employers but more broadly other areas such as academic admissions, the stubborn refusal to take leadership and face, or even acknowledge, the problem leads to an atmosphere of nihilistic apathy and learned helplessness that negatively affects everyone in all industries and sectors (either directly or indirectly). At the same time, the standard practices that could be less badly filtering applicants are apathetically rutted in a status quo of asking formulaic and often tautologically meaningless questions. A classic example is the very common trick-question: "Tell me an example about a situation in which you overcame an obstacle." This Stalinesque question is asked requiring a secret trick answer usually specific to that individual interviewer, often used to 'test' whether the employee knows that they need to answer with a nebulously positive cloud of words, describing nothing and no one in any way that could be considered negative. If the person answers literally and honestly (giving vital accurate feedback) they are automatically disqualified.

Another critical problem on the HR/Employer side is a lack of clarity of what is truly needed vs. what is ideally desired. That employers saying they 'require' something that they do not actually need fuels the problem of applicants both applying for jobs they are not qualified for and exaggerating their qualification.

A notable variation on this problem is the very frequent issue with technical positions where there is a large, round, standard 'years of required experience' requirement that is larger than the amount of time the technology has existed for. As this issue was explained to me, this often comes from the HR department having standardized requirements that they must add to job descriptions that include set durations of experience for set 'levels' of pay and rank. For example some companies will use systems of 'Lead' 'Senior' and 'Junior' (or various other levels: "middle", "supporting", "researcher"), or simply numbers: Engineer 1,2,3 (or some rank as 3,2,1). And in such HR systems there are commonly pre-set requirements for how many years of experience is required for a given rank or level: such as 10 years experience for level-B and 6 years of experience for level-B.

But very frequently (and this would be quite funny if it were not so problematic) the technology, tools, skills, and techstasks for which N years of skill are required are (predictably) younger than the required experience. It is not difficult to find a job post that requires ten years of experience in a one year old technology and a PhD in a field that does not exist yet. The job description is too often a frankenstein of segments of descriptions that no person or group owns or is responsible for, shaped by the hiring manager and hiring department and interlaced with standardized norms and requirements from HR such that the whole is incoherent and very often logically impossible.

Given such job descriptions, how are applicants to interpret and frame their qualifications, and how are the hirers to interpret these applications? This is especially tragic in engineering where STEM skills are the bottom line, and neither the job nor the person are being communicated about in a pragmatic way, leading to confusion, loss of time, wasted resources, and decisions resulting in liabilities not assets.

## Utilize Human Expression & Modeling

One of the most amazing abilities that people have is the ability to encapsulate or express a non-simple situation using a variety of media in ways that do not mechanically derive from regurgitated discrete observations.

We should use people's ability to, if unconsciously, model situations to which they are exposed. This is one of our species most amazing talents, yet it is poorly understood and not well utilized. People can do an amazing job of reflecting and conveying nuanced situations such as often arise in hiring. This is more an art than a science, to date, but integrating this with known STEM is part of the blend that can make things work well (blending arts and sciences as advocated by figures such as Steve Jobs).

## Utilize Learning

There is a lazy status quo where no one wants to facilitate learning, spending much more time looking for someone else to have already taught it. In reality a job candidate who is willing to learn and has good project/product management, communication, and initiative-leadership skills is infinitely more productive than an employee with a similar sounding background but who lacks those skills (especially when either the particular job skill can be learned within the normal onboarding time period or that due to workplace specific nuances the job needs to be effectively (re)learned anyway by whoever is hired (which is often the case)).

Complimentary to this, having diversity in your background in a similar boon. The broader your background, the more accustomed to learning you are, and the more interested in learning and problem solving you are, the more productive the project is.

I have been on many projects and builds at many companies and organizations. Every project involves learning some new tools and skills. As a manager and mentor (perhaps because of my background as a STEM educator) I regularly recommend (and purchase) online courses for my co-workers. You cannot work in STEM, especially computer science and data science, without life long learning (cheesey as that may sound, feel free to invent a hip way to say the same thing). Based on the reaction of my co-workers, it seems my emphasis on learning is uncommon.

It is exceptionally rare to see in a job description any representation of, or emphasis on, the primary and foundational skills of project management and learning that are actually needed for a successful build. This absence is needless self-harm by organizations against themselves. Just as employers should (if only for themselves) clearly distinguish between needed stills and luxury skills, they should also distinguish between sufficiently-easily-taught-skills and not-easily-taught-skills.

## A Paradox in Examining Experience

A very common self-imposed limitation by employers is that they, often (not always), strictly only ask about the type of proprietary work experience that is under NDA (non-disclosure agreement), meaning they can know the least about what that was.

In STEM, there are many avenues for work and experience including open source projects, volunteer work, internships, academic work, semi-academic courses, online publications (technical blogs), maker and hackathon events, certification-related experience, trade-group-events, etc. The more "open" a person's work-sample is, the more that can be learned from it.

## Expectations about Communication & Cultivating Effective Communication

Some parts of hiring involve impersonal data-entry, but other parts such as job description formulation and applicant interviews involve full human back and forth communication.

Let's compare common American and Japanese perspectives and assumptions about meeting and team-communication: By analogy, if too much in summary, Americans view people as houses with a front door and straight front walk where the only entrance is straight. In Japan people are seen as buildings with only an obscure side-door accessible only by a winding garden path, where any straight and bold approach will find no way in. In the US the world is expected to be clear, straight, and absolute. Reality, to the American, is not contextual. There is one truth and it is the same everywhere.

If oversimplified it is largely the case that Americans not only prefer direct, blunt, quick, one-shot communication but also believe that this is the true way that real world communication can and should occur. Likewise (or contrarywise) in Japan both the preference and the perception is that communication takes time, many approaches, many indirect approaches, and many events among networks of people who have different relationships, and that a combination of 1:1, 1: many, and Many:Many events, often involving an array of protocols (not unlike IT network protocols) and improvisations. In Japan the world is expected to be full of nuanced, subtle, and non-literal elements, many of which are unexpected and not automatically perceived (and for example only perceived by some people). In Japan, the world is assumed to be contextual: All the conditions and above rules are specific to each context: superficially the same words said over a meal, in a house, at a meeting in an office, at a public event, in a hot-spring, on the factory floor, are to be understood in that particular context.

Both the American (as in the USA) and the Japanese perspectives could each be seen as 'one-sided' if taken to extremes, such that the real natural world is not always best reflected by one or the other. Sometimes contextual differences are not significant, sometimes they are. Sometimes a quick and solitary communication is practical and effective, sometimes it is not.

Part of the background here is that IT projects in the west very often fail in ways that come (again and again) as a surprise.

https://www.economist.com/podcasts/2025/05/15/why-it-projects-so-often-go-wrong https://www.economist.com/business/2025/05/08/why-so-many-it-projects-go-so-horribly-wrong

It should in no way be a surprise when a group of people who are not effectively in communication with each other about what they are doing do not, by sheer luck, arrive at an accidental agreement of actions. Evidence of a lack of effective communication about a project often characterizes the hiring process as the project team is assembled.

Note: I have chiefly lived and worked in the USA. I had the great privilege of living and working in Japan for nine years in a cross-section of setting: large companies, small companies, pharmaceutical and nutraceutical companies, engineering companies, education-companies, municipalities, libraries, and schools from pre-k to night-school, and experienced some peculiar interviews at the University of Shikoku where some were both surprised and unhappy that I chose to help in the 3:11 tsunami affected region in the north rather than take their job-post offered).

# **Case Study:** Technology to Support Staffing, Peterson Partners and Mindbank.ai

To illustrate more concretely some of the technologies, features, and issues involved in developing tools to support staffing I will attempt to summarize my work with Mindbank and Peterson Partners (both, incidentally, are cases where several companies share the same name).

- https://www.ptechpartners.com/
- https://www.linkedin.com/in/nick-shah/
(a ceo and company with extensive hiring experience, especially for technical and academic related job placements, who were a pleasure to learn from and work with)
- https://www.mindbank.ai/workforce

Though Mindbank dissolved (hopefully eventually reforming) before this project was fully used and adopted, I did build and deploy (including testing and using) this full-stack (backend and frontend) hybrid search and match system from scratch, intended for use by the staffing company and thereby, to some degree, understand the various parts in issues involved.

Mindbank's CEO's vision for a helpful hiring and job-seeking tool was a high level design that would allow people working for the staffing company (or in HR or hiring in any company) to integrate several types of information, for example that would allow a hiring manager to (as a high level user-story type goal) not only match a prospective applicant to fields in a job description, but match the personality and world-style of the job applicant to a whole person whom they wished to fill the shoes of. Obviously this is not meant to be a hyperbolic science-fiction or supernatural over-reach into the ridiculous, cloning: a complete adult complete with memories so as to fill a job post. Rather the goal is to incrementally and concretely expand the types of search and match that are possible (assuming you have a pool to search in, which is not a trivial assumption). Additionally, job seekers should also be able to search for open jobs, and those hiring should be able to search not only among people who have applied but looking for people who they think should apply so as to reach out to them.

For example, a standard set of search and match tools (using computer science but not statistical data science) is a routine structured search using a routine database as has been used since the early days of computing, where the data are exact (or perhaps slightly fuzzy) fields of a specific type: zip code, phone number, name, job title, name of school, type of academic degree, etc.

What was desired in this case, and what was built, was a 'hybrid' database, or a database (or a set of databases) that contains not just different structured types of data, but a 'vector-database' of both structured and unstructured sets of data that allow a very flexible natural-language search to query for not just variations in spelling (though that is a significant tool) but more advanced concepts such as personality traits, tone of voice, maturity levels, etc. One of the perhaps overly ambitious assumptions of this system is that there would be significant samples of different kinds of data from each person: essays on standard topics, answers to comparable questions, interview transcripts, psychological evaluation results, etc. The idea was not that a vector-database could somehow turn a resume into a map of the person's personality, but that IF you had diverse materials from both a set of candidates and from the people who they would be replacing and or working with, THEN (assuming you had such riches of material) you could do a hybrid 'search and match' combining or alternating between structures tables of fixed information and fuzzy-conceptual AI-matrix-vector analysis. There is also the additional assumption (not always unrealistic) that you have both analysis result "metrics" (a third type of data) and a rigorous way to do that analysis (or otherwise obtain the results).

This design used three different types of data in a 'hybrid' database set:
1. structured, tabular, data
- field-values as in database/spreadsheet
e.g. Last Name = Sandwich

2. unstructured 'vector' (concept-search) data
- articles
- answers (such as interview question answers or psychological evaluation answers)
- resume data
- misc blurbs
(could also include social media posts, interview recordings, github projects, worksamples, publications etc.)

3. metrics data (analysis results)
- e.g. personality/psychology test results

Staffing Hybrid Database Search-&-Match Goals:

1. Being able to match an applicant for a combination of:
A. Structured data (such as distance of residence to job)
B. Fuzzy-Search for skills in application (overlaps with structured search)
C. Concept-Search for topics or personality traits (such as assertiveness or mentoring-type activities)
D. Metrics-Scores from tests including personality tests

2. Being able to not just match a person to a skill-set (as is standard) but match more closely to an individual profile of personality and work-style. This could include matching a person to who they are replacing and or to who they will be working with.

3. Being able to (in an automated and time-efficient way) give clear feedback to rejected applicants as to specifically how they were rejected, so that the application can either
A. acquire a new needed skill/experience area, or
B. repair an incorrect communication in the application materials (e.g. if the applicant did in fact have the skill they were thought not to have had)

4. Modular and bi-directional: Job seekers should be able to see what jobs they should apply to just as job-posters should be able to see (search and match for) what applicants (or potential applicants) may be best for the position.

5. Chained-Search-Match
Another feature that is important for practicality (which was part of the deployed system) is allowing for compatible and modular integration of search results for each type of search for each set of data. For example when either someone looking for a job or an employer looking for a prospective applicant or comparing applicants, case by case they will be 'stacking' their search in different sequences using different aspects of search. Sometimes the search will be begin with a location and then go to a fuzzy skills match and then to work place description, and then to a personality type. Or the search may begin with personality metrics and then go to structured job skills search and then to fuzzy job experience search, etc. With each step of search-match-filter, the initially 'too big to deal with' set of jobs or people gets narrowed down until the results are narrow enough to be practical (e.g. from 20k down to 20).

6. As is perennially important in data science (if not often discussed or planned for) structured-unstructured data interfacing and integration is crucial. Having (as this system had) an engine for creating a core standardized structured database for use with unstructured input (just as location-addresses are, for obvious reasons, increasingly using, where unstructured location descriptions are input initially and then 'snapped' to an ideal standardized form and structure) is a key under-the-hood set of features (or sub-features).

A stretch goal of this process could be said to be creating 'digital twins' of both the applicant and of the position or team or project, so that analysis (and even interaction) can happen experimentally with the digital twin more easily (and in some cases with more numerical clarity) than with the real people. But here the rhetoric of 'agents' and 'twins' can easily become lofty, not-sufficiently-defined, and impractical. A more concrete use-case as described by Peterson Partners is that, especially for technical positions, there is an ever-smaller number of people who can ask more specialized questions (such as questions about a niche technology or specific work practices). As a common-sense matter of time and resources, the more filtering that the most generalist staffing employees can do greatly assists the later stages of selection and hiring. If and when job-post and person-description information can be more diversely and more accurately described, processed, and compared (where the standard is still people emailing and uploading vague PDF files that cannot be reliably processed at all) and searched, matched, and filtered, both the applicant and the hiring team will be able to make much better use of their time and get less-noisy results.

Such a hybrid search-and-match system can be, and has been, built. The building of such a system does not represent a significant challenge per se (all the pieces are standard parts of computer science). Greater challenges are the larger scale questions of where what data will come from and who needs to use this system and how. There are various significantly different directions such a system could develop into (internal corporate systems, public system, academic systems, social system, autonomous systems, assistant systems, IoT systems, private systems, public system, etc.), and all are non-trivial eco-systems with many participants and moving parts, and where mis-alignments about who needs to do what and other snags can snowball into development challenges. For example, the user-interface alone could become an unwieldy ever-ongoing scope-expanding-monster undertaking (or it could be small, quick, clean and easily maintained).

### Where Do You Look

Before the internet the options of where to look for prospective employees were probably fewer than future technologies will allow for. The choices may have been (somewhat nested): A. Reach out to people on an 'internal' list who might be interested (including hiring through internal connections) or B. Post the job publically (such as in a newspaper, or perhaps a trade publication), with events such as job-fairs being perhaps a grey area, or C. Get 'outside help' assistance, such as hiring another firm who more or less have the same options: internal connections or public-external outreach.

Anecdotally, I have heard that 70% of positions are filled without being publicly posted. Due to the ease of using an already 'internally' known person vs. the logistical nightmare of finding someone else through a public post, a preference for 'internal' hires makes sense.

In theory, while it would involve some amount of investment, more options may emerge. For example industry-specific bodies may help to curate known, vetted, people who work in the industry. The industry of 'certifications' such as CompTIA might be seen to be part of this.

## Having "Updated Candidate Pools"

Building up and updating a pool of candidates about which detailed information is kept is a significant investment, though less onerous if this is a supplementary marginal search only applicable in some cases (e.g. where there are known people who may relatively frequently go from job to job such as Front End Dev contractors).

## ESG, Diversity & Supply Chain Reporting, Regulation, and Management

From 2020-2022 I worked for a company specializing in managing diverse-hiring for supply-chain sub-contracting, which is yet another set of dimensions around "hiring" not yet mentioned here. Hiring is not always one person based on one job-description. Sometimes larger projects and contracts, and even portfolios contracts and projects, need to be managed and reported on with aspects such as governance, diversity, federal requirements, etc. Not to mention that sustainability and diversity are both hard-science STEM areas and at the same time in less defined and usually arm-wavy ways cultural 'political' fad and social-conflict areas.

"AI"

Throwing more vaguely understood 'technology' at an industry that has for a quarter century been ever more drowning in poorly understood technology is likely going to prolong the state of floundering in a quagmire of chaos. There is even more 'potential' for the future to be brighter, but there is no less a need for actual focused investment of time and resources and coherent coordination for development.There does not appear to be a precedent for spontaneously emerging free sustainable profitable self-maintaining tech that rescues areas such as journalism and hiring. What people are waiting for I do not understand, but waiting to be miraculously saved by someone else out of the blue is likely going to be very long if not indefinite.

As in the above case of the built but not utilized hybrid-database for hiring-matching, there are various AI/ML tools that can help with hiring that do not have any necessary connection to generative AI chatbots (for example: vector-"embeddings" are related to (and very useful) but not the same as fad-popular generative-chatbots).

(Note: While somewhat speculative, I strongly suspect that improved and expanded use of 'vector-embeddings' (or concept-matrix-vectors) will be both practical and an area of significant future new developments. The asymmetry between attention on generative-models (or generative uses of models) and direct vector use and analysis is extreme. While 'generative' tools, books, discussions, products, resources are proliferating so extremely that it likely indicates a kind of 'fad' or 'bubble' or 'AI Summer' boom, direct-vector resources are sparse, hard to find, or completely unavailable. I recommend keeping a keen eye on how you can use and develop direct use of 'vector-embeddings' (or concept-matrix-vectors).)

The topic of AI/ML use across the hiring practices of everyone in the world who does hiring is too vast of a topic to cover here in detail. (Am working on a report (draft text in progress here) on the topic of AI/ML adoption generally in the year 2024... which is taking too long to write as it is a massive topic).

A few specific highlights of important technical topics might include:

1. vector-embeddings and vector databases and hybrid databases
2. tech-stack sustainability, maintainability, and profitability
3. Clickbait fantasy addiction epidemic aside, there is still a significantly large problem of making AI/ML production-solutions that are good enough to use. Making an AI/ML tool that can be useful to a person using it is a very difficult and expensive and long process that people insist on assuming (incorrectly) to be quick, easy, and cheap. Making a fully autonomous AI/ML tool is so difficult that a given solution to a specific problem may not be ready in your lifetime.

- Try to think of STEM-solutions, not black-box-'AI' solutions.
- Try to think of STEM-solutions that very specifically does something that is clearly needed, as opposed to a magic silver bullet that vaguely does stuff that presumably replaces other-stuff. There are managers and business operators who somehow believe that there is a 'free ride' end-goal equilibrium, where you simply sit on static magic software forever and collect 'rent.' This pathological mindset is completely contrary to the realities of STEM and AI being both anything but passive, easy, self-maintaining, and 'hands-off' reliable. A delusional quest for free riches (by tech-illiterate bros) has nothing to do with the decades of intense work that STEM development requires.
- Do not expect smooth rapid progress.

- As in 1996-2001, there is a mix of real investment in what will be valid technologies along with bad investments in mirages, with also some accidentally good investments and incidentally failing investments (or investments that took longer to develop). And as in the early internet, there is probably still in 2026 web-spending for the sake of web-spending that is in no way profitable or productive: this could happen with AI, where it simply becomes a lemur cliff-jumping cultural mindset.
- As of 2026, the late 2022 renaissance of deep-learning AI still has no clear general or specific applications outside of people wanting to play with it (and that being so expensive to provide that the cost must be subsidized). Said differently, 90-95% of business investment in 'AI adoption for the sake of AI-adoption' is an unprofitable loss or a total failure, and perhaps more importantly no one knows what the supposed profitable 5% is.

- https://www.economist.com/finance-and-economics/2025/09/07/what-if-the-ai-stockmarket-blows-up

"*A recent study by researchers at the Massachusetts Institute of Technology finds that 95% of organisations are getting "zero return" from investments in generative AI.*"

This eight minute video is referenced here as being more allegorical than literal. Whether or not this represents a literal scenario (though it possibly does), this video does a decent job of conveying a high-risk company investment choice scenario to be avoided.
https://www.youtube.com/watch?v=TbIPUVkImGE

## cloud vs. local 'AI/ML' deployments:

An important decision for tech-stacks, AI or not, is whether you want to use your own cloud, a 3rd party cloud-service, not use the cloud, deploy 'on edge,' or other options. This needs to be decided case by case.

## Statefull-ness and Expectations of AI:

One of the main repeating problem-areas when people start trying to implement an AI solution for a business is that they have wrongly assumed that the AI-technology they were using was automatically 'stateful' (and also that it automatically converts between various data-types and definition-types of state data). (For much more analysis of project-object-state and AI, see object_relationship_spaces.

## AI Assistance Goals vs. AI Automation Goals: **Levels of Autonomy: the 1-5 system used for automobiles**

During many interviews I have been told about an AI-Automation project (since AI/ML has for years been my area of work). In most cases when I ask about the details whether the goal is fully autonomous automation or a tool to assist a person, the description then follows a common pattern of human inconsistency. Sometimes their narrative clearly describes one path exclusively, at other times the same narrative describes the other path (also exclusively), as though the person is brainstorming about very different directions they might want to go in. And while sometimes the person recognizes that they need to decide on the scope, most people boldly assert that deciding on a project scope is unimportant and (arms waving) rushes on to other topics.

As a compound issue (part goal-setting, part communication), it is very common that during interviews and discussions I will ask two questions:
1. How are you defining 'AI'? What exactly do you mean? (There are many technologies out there, some I have worked with more and can speak to, other less so.)
2. What are the specific features and functionalities that you are trying to build?

Often the answer has two parts, and too often a third part.
1. They have no explanation of what they mean by 'AI.'
2. The features they want to build do not require any 'AI.'
and
3. ~"Well, the boss says he wants us to use AI, so... we put the sign in the window."

As I am often not speaking directly with a senior manager (though sometimes I am), this, importantly, does not mean without a doubt that there is a complete lack of careful planning and goal-setting. However, this does show that there is at least a failure of communication. Good systems and practices of communication are critical for IT projects. If the people throughout most of the chain of communication below upper management have no idea what project is trying to do and how, that alone is a significant area for improvement. (Now, back to the topic of assistance vs. automation.)

Having done both types of projects (assistance tools and automation tools), I can say from experience that an human-assistance goal project is utterly different in scope and profoundly smaller and easier compared with a full-autonomy goal project. Not knowing that you do not know what type of scope your project has is a serious problem. Having a goal that is exponentially more difficult and costly than what you actually need to do is another serious problem.

Deciding what features are being built (and why, e.g. for whom and what their uses are) is best practice, is very important, and nearly always a firm necessity (if you try to skip this, the project will go off the rails).

To help with getting ready to think through where tools can best assist people vs. where you may truly need a process to be autonomous and unattended, let's go through the 0-5 scale for levels of autonomy for self-driving cars as a way to better understand some of the spectra, concepts, and terms. Going through examples like this should help you to better define and describe what you want your features and functionalities to be.

According to the US National Highway Traffic Safety Administration (or NHTSA) there are six levels of car-autonomy, which somewhat confusingly start with a 'level 0' that is (contrary to internet clickbait) NOT a 'no-assistance' level.

See:
https://www.nhtsa.gov/sites/nhtsa.gov/files/2022-05/Level-of-Automation-052522-tag.pdf

Here is a paraphrased (with more examples) outline of the NHTSA system.

### Level 0. Momentary & Rare Assistance to human action / human operator:

- alerts, warnings, specific event response
- e.g. check-engine light
- proximity beep
- air-bag deployment during a crash
- anti-lock breaks reacting to unusual breaking
- (analogy: highlight possibly-misspelled word)

(Note: There is a significant jump in how 'assistance' is defined between level 0 and level 1. This may be useful to study for including and balancing older and known technologies not popularly felt to be 'exciting new and mysterious AI!' yet are included in a clearer .STEM set of definitions of AI/ML/Data-Science.)

### Level 1. Single-Task Continuous and Routine Assistance to human action / human operator:

- marginal breaking help alone (single type of help)
- marginal acceleration help alone (single type of help)
- marginal steering help alone (single type of help)

(analogy: spelling suggestings, continuous auto-save)

### Level 2. Multi-Task Continuous Assistance to human action / human operator.

- marginal breaking help (in addition to other assistance types)
- marginal acceleration help (in addition to other assistance types)
- marginal steering help (in addition to other assistance types)

(analogy: spelling suggestings, continuous auto-save)

### Level 3. Optional Complete Continuous Control of All Systems, with human present for oversight and hand-over (human taking control)

- during windows of optimal and least-risk contexts, AI can take more or all actions alone, but overall there is a human driving.

### Level 4. Full Complete Continuous Control of All Systems Only Within a specific context (location/time/conditions/etc.) (no human required to be present IF in the specified context)

- 'context' limitations can be: space location, weather conditions, specific time-windows, etc.
- autonomous cart within factory, often full of sensors to support and contain cart
- parking lot device outside of business hours (e.g. 2-am to 4-am)
- An autonomous shuttle along one 'geo-fenced' street during lunch-hour in sunny warm dry clear weather.

### Level 5. Full Complete Continuous Control of All Systems in Any Context (no human required to be present in any context)

- all contexts (time, place, conditions, etc.)
- access to all roads (public/private)
- access at all times
- access under all conditions

Note this government regulation wording is significantly different from, and more detailed and nuanced than, popular internet posts that often use a ranking like this:

0. No assistance or autonomy
1. simple assistance (no autonomy)
2. advanced assistance (no autonomy)
3. very limited autonomy in specific contexts
4. moderate autonomy in more contexts
5. full autonomy (all contexts)

Note: It is interesting and curious that the distinction between levels 1 and 2 does not occur after level 2. Assistance in one area vs. assistance in multiple areas is a useful distinction. But there is no level for continuous control in one area vs. continuous control over multiple areas. When you are mapping out your own systems and goals you may want more granularity about how many areas are being assisted or running-unsupervised, as well as perhaps more granularity about 'unsupervised' as there may not be any processes that you will run 'in the dark' without any human or other external attention ever.

- Synthetic Data & Testing
https://github.com/stemnetbenchmarks

- Loss-Function Design and Outcome Based Learning:
https://docs.google.com/document/d/1O1xk6CxuubRqNqdh03ETlhtqrW5EFzGoiiRMseV8s2A/edit?usp=sharing

- Embedding-vectors and tokenizers (Perseid Family of Models)
https://github.com/lineality/gutenberg_babble/tree/main/perseids

- Pretraining vs. experiential learning, and model fragility (TWIML)
Model Fragility and New Issues since 2022
https://twimlai.com/podcast/twimlai/is-it-time-to-rethink-llm-pre-training/

- Broader AI Adoption Review (early draft)
https://docs.google.com/document/d/1tTDNP2X0EyECPu6Tar0oHK9q9tlRg1IaTIdAsB6GQwg

## Testing & Interfacing and Testing Structured/Tabular Data Search with Generative and Vector-Embedding Sub-symbolic Model Technology

Wherever possible you want to create, and automate, and regularly use, meaningful tests that evaluate and give diagnostics on each area of your systems.

If you try to use a generative tool to help with or perform structured-queries, you should have a benchmark that is modular, scalable, and that you can modify for your specific needs. Here is an open source structured query benchmarking tool that I put on github: https://github.com/stemnetbenchmarks/sythetic_biology_datasets_python

## A Family Tree of NLP Technologies

A mature pipeline that handles text and language will usually be a family tree of technologies of assorted types, both older and newer, symbolic and subsymbolic, statistical and GOFAI, high level abstractions and language, and low-level 'bare metal' optimized code and languages, etc.

I highly recommend "Natural Language Processing in Action: Understanding, Analyzing, and Generating Text with Python", by Hobson Lane, Hannes Hapke, et al. (1st Edition) https://www.amazon.com/Natural-Language-Processing-Action-Understanding/dp/B07X37578L/ (there is a second edition out now, but the first edition is such a classic, and the audible is very well done).

## Be clear about your definitions and goals. Be wary of Jargon.

Whether or not you have really found a ready-made solution from someone else, you should always very clearly understand your use-case and exactly what you need to get done.

In many cases you are far better off creating a do-one-thing-well solution based specifically on what your project is doing and how.

Be very, very, careful when trying to use a third party service or an approach with an attractive academic jargon name for a task that it may not actually match. I would extend this warning to even technologies that are often assumed to be commonplace, such as "SQL" databases and spreadsheets. Data Science, AI/ML, and computer science has evolved (if slowly) since the 1970s, and often not "out of, from, and with SQL and spreadsheets", but "despite and outside of SQL and spreadsheets." In the spectrum of commonly-used technologies in 2026, some of these are 'bad habit,' 'junkfood,' and 'cargo cult' technologies that are wise to avoid. Case by case, a given technology may be exactly what you do need, or may not be.

If you follow best practice, define your Project Areas, stay aligned (as in Agile), and move transparently and with feedback in iterative steps you should be able to catch the problems that otherwise (that usually) lurk out of sight until they spiral out of control and prevent your build from working.

### More Discussion of Technical Points for Data-Science AI/ML Builds

https://github.com/lineality/object_relationship_spaces_ai_ml
https://medium.com/@GeoffreyGordonAshbrook

## Are Good Solutions Out There?

There hopefully are hiring-tools that CRM or other services such as Monday.com that are very effective and helpful. There must be good practices out there, even if they are specific to a slice of an industry.

# STEM

There are many advances in areas of STEM that have been achieved in the last century that should be able to make hiring and admissions significantly more productive and efficient, from better understanding of individual and organizational decision making and cognitive psychology, to better understandings of various skills and diversities are assets to companies and institutions, to better understandings of the best practice, good governance, process and project skills that personal should have. Somewhere out there are vital formulations of crucial and hard won lessons learned that could help prevent the cycles of blindly repeated mistakes that cripple organizations.

Decisions, Coordination, Project-Area Definition & Alignment, and preventing system collapse:
One example area that is specific to alignment on project areas is the Uma Project. While technology supporting communication and decisions (and the study of factors around that) may relate only indirectly to hiring, the topics of project area definition alignment, coordinated decision making, and preventing system collapse, do quickly become priorities and parts of the overall problem space: deciding and communicating what exactly the job

description and work areas are, and coordinating a personnel decision that is a future asset and not a liability.

In addition to my background in biology, linguistics, psychology, learning and education, data science, agent-based technology, and AI-Machines learning, my research has been more specifically into study system collapse the role of projects, productivity, and social-values (such as Boy-Scout Values) in STEM.

The 'T is for Task' article (about Project Uma) will hopefully serve as an applied-topic introduction to the intersecting STEM areas involved in maintaining healthy systems.

## The Vital Importance of STEM over the significant liability of Nihilism

It will take work to break the cycle of problems that we are not aware of. Projects, including hiring, failing because people are not defining what they are doing, not communicating and aligning about what they have not defined (not giving and getting feedback about that), not making proper decisions or coordinated decisions, and then when things keep failing (tautologically because there was no actual define decided on aligned on and coordinated process that could possibly have succeeded) people double down on superstitious nihilism: the world magically fails so don't bother. This is wrong, this is bad, this is self harming. The problem is not magic-evil, or 'the human condition' or 'the wheel of samsara' or 'minorities and foreigners' or 'women and nature,' or a lack of strong-man-violence, or other scapegoats blamed by the bad-equilibria that people historically and erroneously settle on.

If the clouding of nihilism becomes a problem, I recommend using or making your own tools such as those of Definition Behavior Studies, to maintain a defended perception of what can be done and so that you can best choose, decide, and take initiative on, what should be done.

# Appendix 1: Critical Thinking Around a Fearmongering-Article on AI in Hiring

It has been speculated that "AI" may have any of various types of disruptive effects on hiring. This is not one of The Economist's better vetted articles:

https://www.economist.com/business/2026/01/12/job-applicants-are-winning-the-ai-arms-race-against-recruiters

I would recommend comparing the topic of AI in hiring the AI in elections. There is a significant doom-and-gloom clickbait industry that rewards any conspiracy prognostication about horrible things, but the reality is much more boring and much less scary.

1. While this article portrays AI tailored application materials as being inherent illegitimate (conflating it with North Korean Industrial Espionage, apparently just for the sake of saying something unrelated that sounds scary) anyone can see (as most people are on linkedin) that AI tailored application materials are a standard (Microsoft-owned) Linkedin service that they encourage everyone (which is quite nearly literally everyone in the world) to use (for a fee).

It is not clear what exactly this article is 'dooming' about (or sadly maybe that's how newspapers have to pay the bills in the age of the internet). There is no discussion at all about what might be, or if anything is, wrong with these technology-supported applications.

2. Broadly speaking generative AI simply is not good enough to create more than slop, which could credibly be the real problem (not North Korean spy applications, for heaven's sake). If job descriptions and applications and interviews all devolved into AI-slop, that would indeed be problematic (but that topic is not mentioned in the article).

3. For comparison, three other areas where people love to make dire predictions about the impact of AI are copyright, journalism, and elections.

In all these cases we can probably imagine, if vaguely, a future where a virtual world somehow overwhelms reality. Maybe election events could be drowned in fake election events. Maybe newspapers could be drowned in fake newspapers. Maybe AI-created products could drown out human-created products.

In reality between 2022 and 2026:
1. There are no emergent profitable, useful, AI solutions that any known company has found to do anything.
2. Many elections have happened between 2022 and 2026. And there are no known elections where it was impossible to carry out the election because of "AI."

3. There are no publishing or copyright revolutions or take-overs due to AI.
4. It has not become impossible for journalists to distinguish between world events and fake-ai-events.

In these cases, and likely with hiring as well,

1. There is a market for doom-and-gloom headlines, no matter how vague or fictional they are.

2. STEM technologies exist in a very diverse spectrum. While people love to fear that a 'frankenstein science monster' will do something bad, in reality 'technology' and 'AI' are very diversely distributed in what people do, to the point that it is often not meaningful to imagine that 'technology' and 'AI' are essences separate from 'natural-human-ness.'

This may also be a product of an irresponsible way in which the 2022 Chat-GPT technology was presented to people, which is making even practical business adoption difficult since expectations are science-fiction fantasy that do not connect with reality. In various ways, Open-AI is deliberately creating an entertainment-illusion of statefulness without disclosing that it is an illusion for self-entertainment. Even on more technical levels, they designed their API interface to pretend to divide the 'context history' into system-instructions, ai-role and human-user-role, inputs, a division that is by and large a fiction for the aesthetics of the human user interface.

Other parts of the separation of the fears from reality are that the AI tools that we have do not work the way people imagine, and fear, that they do, in three or four specific ways:
- Concept-vectors do not learn from experience (they are mostly frozen).
- Concept-vectors are 100% reactive: they do not take actions on their own volition.
- Concept-vectors cannot input or output 'stateful' information.

and fourth:
- The tech simply isn't that good. Though in theory this could change, from 2022-2026 we have increasing evidence that AI can "do" some "tasks" reasonably well, but it is both unable to do most tasks and 'scaling up' (throwing more money at the same methods) is not

making GPT-AI any better. In other words: AI simply isn't good enough to be harmful in the ways feared.

The fearmongering term "AI" suggests an autonomous super-intelligence which (as of 2026) does not exist. Whereas the STEM term 'AI/Machine-Learning' is a very, very, broad range of technologies that includes nearly all of computer science, applied math, statistics and probability.

We have an intuitive fear of an evil monster making a picture or a document, in a world where we feel that reality can be cleanly divided into a 'natural human mind created world' on the one hand and a 'just like human but evil artificial-mind world' on the other hand. (As a side note, this dichotomous anti-modernity fear-fantasy is significantly similar to parts of the paranoid NAZI cult and ideology around Germany that chiefly led to WWII, and as history sometimes repeats itself we may want to be vigilant about such populist paranoia. See:
- https://www.amazon.com/Rise-Fall-Third-Reich-History/dp/B003X4R6GQ/
- www.amazon.com/Bloodlands-Europe-Between-Hitler-Stalin-ebook/dp/B00B3M3VE6
- https://www.amazon.com/Black-Earth-Holocaust-History-Warning-ebook/dp/B00R04OVOI

) But that fear-fantasy does not correspond to reality. There is no such thing as, for example, a PDF with text and pictures that is pure-human-natural-good and that was made without any technology or math or statistics or heuristics. Nor is there a learning, autonomous, active, (identical to a person), evil-AI that will jump out of a parallel dimension and start flooding the world with evil PDF documents created using evil-conspiracy-technology that is separate from 'human-natural-pureness-pdf' methods. Such a world-view is not only a fictional-fantasy, it is incoherently nonsensical and infeasible.

The phrase "a resume made using AI" suffers the same definition problems as the phrase 'an image made using AI.' The words imply a split world that sounds attractive to many people: AI vs. Non-AI. For some reason we are drawn to believe in a Manichean dichotomy between pure-natural-resumes and evil-technology-resumes. But these reified polarizations of fantasy do not match up with reality. Virtually no digital image exists that does not use some technology that is statistical, or heuristic, or a form of 'AI/ML.' Likewise it would be very difficult to find or make a digital document that did not use some technology that falls into some category of AI/Machine Learning/Data Science.

Likewise, people imagine a teleological progression of technology from old-fashioned lower-technology to new future-tastic higher-super-AI technology. But the timeline of STEM is fascinatingly very unlike that expectation. In most cases, however astonishing this may be, AI-ML branches were developed before computers and digital technologies. Bayesian Machines learning in the early 1700's, Laplace's statistical work in the last 1700's, Ronald Fisher's statistical work in the 1900's, Regression (an ill-defined term with diverse meanings, sadly) work by Newton in the 1700's, Gauss and Legendre in the early 1800's, and in different ways by Galton in the 1800's, Turing's and Shannon's statistical work in the early 1900's, sub-symbolic artificial neural networks in the early 1940's, Kepler's methodology for systematic pattern learning in the 1500's. Etc.

- spell checking is AI
- grammar checking is AI
- auto-correct (such as capitalization) is AI
- hard-drive read-write is AI
- language translation is AI
- file search is AI
- web-search is AI
- graphics optimizations are AI
- image-rendering is AI
- most things involving statistics are "AI"
- most things involving STEM are "AI"

And even if we imagine a job-application or resume that is created and submitted by AI autonomously, what part of that task do we so strongly not want to be automated? That task is form-filling data-entry from one place to another in the same known set fields over and over again. For example, when a 'person' fills out a job application on a web-browser, the fields are automatically populated (an automated 'AI' process, filling in and completing the application) by the browser. Does name-field form-filling by a browser fill people with existential terror? Not... usually. When the terrorizing accusation is made "an application filled out by an AI!", is it suggesting that the fields such as last name, first name, were invented out of thin air (adding a new field called "banana fish", or filled out without reference to reality (inserting an incorrect name)? Do we not want better automated data-entry (do so many people secretly love doing hours of the same data entry)?

How many parts of image publishing software (like photo-shop) are not a form of AI/ML? How many parts of document and PDF publishing are not a form of AI/ML? How many parts of audio and video processing and publishing software and hardware are not some form of AI/ML? You could probably find one or two, but STEM and computer science technologies are (by definition) very common in STEM and computer science systems.

I wrote this article/report (link below) two years ago about the frothing fear that AI would take over publishing, the arts, journalism, and elections, with all the polite caveats and hedges that some unpredictable future catastrophe could not be ruled out.
https://medium.com/@GeoffreyGordonAshbrook/should-we-label-images-text-made-by-ai-916df9ac100a
But all the boring details that I outlined years ago, and the still-not-ending state of the world at that time, have not changed at all in the past two years. The nature of software, or AI, or publishing, have not changed. (Maybe the world will suddenly end next week, but so far my reporting is looking very accurate.)

Again, part of the problem is the deceptive salesmanship by OpenAI that they have some new technology that involves no human input or operation. That may be their future goal, but that simply does not exist in the world up to 2026. It is a goal and dream that more people will be able to use word-processing, spreadsheets, audio-production, video-production, image-publishing, etc. software, but it takes months or years to learn how to use all the features in these types of software.

I am not arguing applications of technology are always helpful or that bad outcomes can not happen (the destruction of journalism looks pretty bad to me). But hand-wringing over automated application form-filling (and nonsensically trying to associate that with North Korean Spy networks?!) is ridiculous. If we misplace our priorities and do not communicate accurately, even well intentioned cautions are unlikely to be practical.

If you want to know, or label, or limit, the use of a very specific brand of technology, for example images made using "Company-XYZ Product AI-123, version 456", that could be useful or at least meaningful. But vaguely saying 'AI/ML' is broader and more unclear than most people understand (schools really need to do a better job).

It may be nice that more people are interested in and passionate about the pop-fad of Chat-GPT, but the term "AI" is not inherently very meaningful. At the time when the term "AI" was invented in 1956 (https://en.wikipedia.org/wiki/Dartmouth_workshop) the term was created to refer to any technology doing or helping with something that a person can do but that at some time in the past there wasn't a technology to do or help with that. This is a very, very broad definition. Again, the term was invented to describe and help with finding a way to use STEM to do something (anything) that previously only people could do. Some of the first examples of this pattern were simply doing math: add, subtract, multiply, divide. Only people could do math, until technologies were developed to assist. The ENIAC project (an early proto-computer, not digital) was carried out to see IF machines could do physics calculations. Until it was done, people did not believe it could be done by anything except an expert human. ENIAC was in the 1940's, and the term AI was created in 1956, but the ENIAC 'test' is a great illustration of what is intended by the term 'AI': some people thought automated calculation could perform useful physics calculations; other people thought that only people could perform physics calculations. This dispute or difference of hypothesis can be defined, and tested: can something that can 'now' only be done by a human mind using some occult ('unseen') means be done with known and reproducible STEM techniques, A.K.A. "artificially"?

See:
https://www.amazon.com/Proving-Ground-Untold-Programmed-Computer/dp/1538718294/

One of Alan Turing's first tasks when building the UK's early digital computer system was to show that computers could do floating point arithmetic (see the must-read Hodges biography). Integer math, sure; by the 1950's people knew that was easy for computers. But floating point? Is that possible?! Only people can do floating-point math! As Turing demonstrated, digital computers can (though still cumbersome to this day) definitely use floating point datatypes.

Over the years this line of what STEM can do has moved further and further. In 2026, the meaning of 'AI' is often unclear, especially since it is in some ways a relative term, especially in popular perception. Early AI researchers expressed frustration that the public only accepted future goals as 'fun AI.' ~'Once you showed something could be done, no one cared.' (Paraphrased) The term 'AI' has consistently been volatile, sometimes embraced, sometimes shunned, sometimes used as a STEM term, sometimes used for art. If you study the field you will read about 'AI Summers' and 'AI Winters' where public hopes or fears influenced how much funding was given (and what people called their research). The fascinating term, useful but always a bit frustrating, poses many questions: Should something before 1956 be described as 'AI'? Is something 'AI' after it is shown to work? There are no absolute rules for using the term, and each time period feels differently about it. (It is a very interesting history that I highly recommend exploring.)

Two highly recommended non-technical overviews of AI are:

"Artificial Intelligence: A Guide for Thinking Humans" by Melanie Mitchell, 2019
https://www.amazon.com/Artificial-Intelligence-Guide-Thinking-Humans/dp/0241404827/

"A Brief History of Artificial Intelligence: What It Is, Where We Are, and Where We Are Going" by Michael Wooldridge,
https://www.amazon.com/Brief-History-Artificial-Intelligence-Where/dp/B088MMPZ49/

Another much different kind of interdisciplinary overview (also recommended) is:
"Possible Minds: Twenty-Five Ways of Looking at AI" Edited by John Brockman, 2019
https://www.amazon.com/Possible-Minds-audiobook/dp/B07MQX54TW/

And a longer list of recommended reading can be found at:
https://github.com/lineality/recommended_background_reading_CS_ai_ml_ds

Sometimes when people use the term 'AI,' as with the irresponsible writers who wrote the above misleading, fearmongering article about evil-artificial-resumes, they are invoking a fictional and fantastical malevolent stateful autonomous boogieman from horror movies and fiction, which as of 2026, is purely fiction. Yet they are problematically presenting this pure-fantasy as fact-reporting. The well documented human obsession with ever more rage-fueling negative extremist fantasy is a serious problem, and that human-problem could become entwined with 'AI tools' that are trained to generate or parrot fear-frenzy inducing content (and that would be a problem). But the ancient tendency of people to get lost in fantasies of horror to the point where they do not know what is real is not a problem that comes from a future-fantasy 'AI.'

The above Economist article should have better used The Economist's ample resources to help clarify the situations and needs of those hiring and those seeking positions (possibly including admissions and recruiting more generally, outside of what is technically 'hiring'). The article you are reading now lacks the brevity, wit, style, editing, data visualizations, and other keen insights that The

Economist can bring to bear. But hopefully this report will be of humble use in understanding and helping to solve some important problems.

# References and Further Digging

More Recommended Reading:
- https://github.com/lineality/recommended_background_reading_CS_ai_ml_ds

https://www.amazon.com/Alan-Turing-Enigma-Inspired-Imitation/dp/069116472X/v

https://en.wikipedia.org/wiki/Spaghettification

https://en.wikipedia.org/wiki/Manichaeism

https://www.teksystems.com/en/insights/article/fascinating-history-of-job-hunting

There are a number of topics touched upon here that I have written about in more depth (and various others that were not mentioned here such as the domain-specific tokenizer design optimization).

"Thinking Fast and Slow"
- https://www.amazon.com/Thinking-Fast-Slow-Daniel-Kahneman/dp/0374533555
- https://en.wikipedia.org/wiki/Thinking,_Fast_and_Slow

"Noise"
- www.amazon.com/Noise-Flaw-Human-Judgment/dp/B08LNYM39M/

For more details and efforts in the interest of productivity and stopping needless cycles of self-inflicted failures, see:

Definition Behavior Studies:
- https://github.com/lineality/definition_behavior_studies

Uma:
- https://github.com/lineality/uma_productivity_collaboration_tool

"T is for Task" Essay (draft)
-
https://github.com/lineality/uma_productivity_collaboration_tool/blob/main/t_is_for_task.txt

- https://docs.google.com/document/d/1F2nzmLXsgbMctHFsHV63fKpa2QIKd7cjFZBANdrluxw/edit?usp=sharing

Coordinated Decisions:

- https://github.com/lineality/Networked_Voting_and_Decisions_Including_One_Time_Pads

AI: 2024 in review

- https://docs.google.com/document/d/1tTDNP2X0EyECPu6Tar0oHK9q9tlRg1IaTIdAsB6GQwg/

Object Relationship Spaces

- https://github.com/lineality/object_relationship_spaces_ai_ml

Project Areas

- https://github.com/lineality/project_areas_for_project_and_product_management

Needs and goals analysis tools

- https://github.com/lineality/needs_goals_assessment_disambiguation

Author's Github: https://github.com/lineality