```
---
id: 93
title: 'Did you prepare your talk?'
date: '2010-11-17T00:12:00+00:00'
author: 'Xavier Amatriain'
##layout: post
permalink: /did-you-prepare-your-talk/
header:
  teaser: https://amatriain.net/blog/images/ted-presenter.jpg
categories:
    - Uncategorized
tags:
    - conferences
    - 'presentation skills'
    - research
---
```

I don’t consider myself to be a great presenter. As a matter of fact, every time I finish a presentation, I find myself thinking about how many things I screwed up and could have done much better. However, whenever I attend a conference I face the cruel reality: my presentations are way better than most research presentations. If I am really not that good, it can only mean one thing: researchers generally suck at presenting their work. (This is in fact one of the reasons I am against organizing research conferences around oral presentations. But this is another discussion I will leave for another post).

So, if you have any doubts of whether you could be in that category of good researcher/poor presenter, you can do a quick test: Watch the video below. If you think your last presentation is well summarized in the video, you definitely fit into the group. Even if you don’t, you might find some tips or advice of interest to you in the rest of this post.

<div style="text-align: center;"><object height="385" width="480"><embed allowfullscreen="true" allowscriptaccess="always" height="385" src="http://www.youtube.com/v/yL_-1d9OSdk?fs=1&hl=es_ES" type="application/x-shockwave-flash" width="480"></embed></object></div>OK, so what are the three basic rules to make a decent presentation? Easy: (1) Prepare yourself, (2) prepare yourself, and (3) prepare yourself.

At this point, you might already be tempted to stop reading because you disagree with what I am saying. I have found several reasons why people disagree with something as obvious as the fact that making a good presentation requires preparation, but I think all of them are summarized in the two following:

<span style="font-weight: bold;">(a)</span> <span style="font-weight: bold;">I’m a natural</span>: Maybe you are the kind of self-assured person that thinks that has great presentation skills and those shine best the more you improvise. I was pretty close to this myself some time ago. But if you fit into this category, there is a very easy test you can do: tape yourself on video on several presentations. If you still think you are great and need no preparation or further skills, congratulations! But chances are that then you realize how many things you have been doing wrong and how much you can improve. All great presenters I know stress the fact that preparation is key, period.

<span style="font-weight: bold;">(b)</span> <span style="font-weight: bold;">I’m a researcher, not a TV Star</span>: On the other extreme, you might be aware of your limitations but might think that this is not such a big deal. You are a researcher and live in the world of formulas, theories, or code. You could care less about what people get from your talks and you would be happy standing up and doing the chicken, chicken, chicken presentation. And this is not an exaggeration: I have seen junior researchers that are still editing slides a couple of hours before their scheduled presentation in a top conference. My take on this is the following: if you think presentation skills are not part of what is required in a researcher, you are wrong.

[![](http://localhost:8080/wordpress/wp-content/uploads/2010/11/ted-presenter.jpg)](http://1.bp.blogspot.com/_xAtUP4Gu6Zk/TO2uqVrAn-I/AAAAAAAAAJw/tje5_aXwlbI/s1600/ted-presenter.jpg)
So by this point I will suppose that you are convinced of the importance of preparing research presentations. Ideally, you have also taped yourself and found that there are many things to improve. The question is what to do next. Obviously, I cannot pretend to summarize a presentation skills course in a post. There are thousands of resources out there in the form of books, videos, or similar that you will find without problem. But I do think that I can pinpoint a few issues that are important and tricks that might help.

First, I think it is important to separate two kinds of “preparation”: (1) mid/long-term preparation aimed at improving your skills, and (2) short-term preparation for your next presentation.

<span style="font-weight: bold;font-size:130%;">Improving your skills</span>

Again, you can find many books and resources on how to do this. But some of the things that you should at least consider:

(1) Tape yourself:

This will make you aware of the weak points and where you need to focus your efforts

(2) Enjoy the stage:

Some people have a really hard time every time they go onto stage, and this shows. There are many things you can do to learn techniques and improve on this that go from playing in a band to taking some acting and performance lessons (I did this and found it very useful and enjoyable)

(3) Read about it:

No need to become obsessed. But reading a couple of books or watching some videos giving you tips is not going to hurt. And remember, this is part of your expected skill set as a researcher. If you want a starting point, I can recommend you read a short 12 page essay on “[How to give an academic talk v4.0](http://pne.people.si.umich.edu/PDF/howtotalk.pdf)” by Paul N. Edwards from U. Michigan.

(4) Rehearse the techniques:

It is very good if you have situations where you can rehearse what you learn from the previous. Actually, many of the techniques can be applied in “real” life (e.g. when talking to your boss). Others require of a more realistic setting. I have been lucky to use the courses at the university as a rehearse playground for improving my skills

(5) If you need help, look for it:

I have seen many cases of researchers with severe communication problems when presenting. Maybe I sound too harsh here, but I don’t think this is acceptable. If you really want to be a researcher but don’t think you can get to an acceptable level of presenting either (a) have some co-author present for you or (b) find some professional help. And this latter would be my preferred option. It is not so hard nowadays to find coaches or places that can help you out and if you agree this is an ability that you need in your job (and, again, you should agree), it is worth that you invest on it.
<span style="font-weight: bold;font-size:130%;">
</span>[![](http://localhost:8080/wordpress/wp-content/uploads/2022/06/chairs.jpg)](http://3.bp.blogspot.com/_xAtUP4Gu6Zk/TO2vZ7AcGoI/AAAAAAAAAKA/ElhgjU5si20/s1600/chairs.jpg)
<span style="font-weight: bold;font-size:130%;">
Preparing your next presentation</span>

Regardless of whether you manage to improve your general presentation skills or not, you will have to face your next presentation sooner or later. When preparing the talk, you should focus on its two main components: the slides, and the talk itself.

<span style="font-weight: bold;">The slides</span>

Again, there are many resources out there on how to prepare your slides, the style, the design.. I was lucky to attend a course on Zen presentation style by the guys at [Presentaciones Artesanas](http://www.presentacionesartesanas.com/). Zen-style presentations are [the kind of slides](http://www.presentationzen.com/presentationzen/2009/05/making-presentations-in-the-ted-style.html) you will see at TED, for instance. I try to bear some of the techniques in mine but (a) I am not a professional presenter (that is, although presentations are important in a researcher’s life, I have other things to do), and (b) sometimes, transmitting scientific rigor in a very graphical style is not easy (actually, [according to Tufte](http://www.wired.com/wired/archive/11.09/ppt2.html), even Powerpoint should be banned from scientific publications). However, I do recommend to understand some of the design concepts behind the Zen style and maybe use some of them as a basis.

Once you have found your style, you will need to do the following tasks:

(1) Know your audience:

Before you start preparing the presentation, take some time to understand who you will be talking to. It’s not the same to do a talk at a conference than pitch your work to business people, present to a prospective employer, or, like I did last week, try to convince high-schoolers of how cool Computer Science is.

Even if you are only focusing on research presentations at conferences, they are not all the same! Sometimes you will be giving a talk in a setting where everybody is an expert in what you are talking about, while in other occasions only a tiny fraction of the audience is working in your same field. In my case, I won’t use the same kind of approach if I am presenting at a Recsys conference where everybody knows about Recommender Systems than at a generic one like WWW, where I can only assume that most of the audience does not know the topic in depth.

It is also important to look at the program schedule. The name of your session and the talk immediately before and after yours is going to give you more information about who might be sitting in. If you are presenting in a conference with multiple tracks, the talks scheduled at the same time as yours will give you some hint about who is *not* going to be attending yours.

(2) Find “the message”

Find a simple take-away message that you want to get through to your audience. In many cases it will be something along the lines of “look how important and interesting my research is, please go ahead and look more into it by reading the paper… and don’t forget to cite it in your next publication”. But in other to transmit that idea you need to make your point. Therefore, find the answers to: (a) what problem does your work solve, (b) what makes your work different from other solutions, and (c) why should anybody care about it. These three questions should help you find the message. Stick to one/two ideas and refer the audience to the paper for more details. Trying to squeeze in too many messages in too little time is a recipe for disaster.

Some researchers like to add another secondary message thread: (d) I am really smart and what I did is so complicated you might not even grasp it… I particularly dislike this kind of presentations and find them pretentious and boring (maybe because I am not so smart). But hey, I know some people have made quite a career of this so you should be aware.

(3) Prepare a script

Once you have identified the “main message,” you are ready to prepare the script of the slides. I usually start off by having a bunch of empty slides with only the title on them. By having this, I can see if I might be going over time, need to sort things out differently… The script will depend on the kind of talk and time you have to speak. But in general, it will have a structure such as:

1. Introduce context and situation
2. Formulate problem and why it is important to solve
3. Main message (Solution to the problem, consequences, details on the solution…)
4. Summary on problem and solution
5. Future work and things to do

The script is important, but be ready and willing to change it. You are likely not to get it perfect from the start and as soon as you start adding more detail you will see a clearer picture. Don’t make the “sticking to the plan” hit you back.

(4) Make the visuals

Maybe you think this is the least important part of your presentation. In my experience, I have come to value the visuals very much. Actually, most of the time I spend in preparing some presentations is looking for appropriate visuals that back up and re-enforce the “main message”. The less familiar the audience is with your topic, or the less hardcore researchy it is, the more time you will want to spend choosing appropriate visuals. Some well-chosen pictures will make your message more sticky. And you might find some images that are so powerful that might make you go back to your script and twist it a bit.<span style="font-weight: bold;"></span>

The talk

Once you have the slides more or less ready, you can start preparing the talk itself. Bear this in mind: if you prepare the slides but not the talk, your presentation is likely to suck. Some ideas and tips that can help you in the process:

(1) Tape yourself

As I mentioned before, taping yourself is one of the best tools I have found for improving your presentation skills. It is also an amazing tool for preparing your next talk. If you watch a video of yourself rehearsing the talk, you will be able to analyze what you are explaining wrong, where you are wasting your time, what jokes don’t make sense… Besides, it is a perfect timing tool: not only you will get the exact duration of your talk in the video but also how you distributed it. This will allow you to make sure that you are devoting the right amount of time to getting the “main message” through.

(2) Test technical issues as many times as possible

Don’t need to mention [Murphy’s law](http://en.wikipedia.org/wiki/Murphy%27s_law), I suppose, but if something can go wrong, it will. No matter how much I check things over and over, there are always technical issues that catch me by surprise.

In my last talk, I had a couple of videos that I knew could be problematic. I spent a lot of time in making sure they were working in the presentation. I even tried my laptop with a secondary monitor to make sure. I asked the host well in advance to make sure that I had the possibility of connecting the audio output of my laptop. And on the day, I went to the hall and tested the audio. I was even going to test the videos but the audience was already half in, so I preferred to keep the surprise and tested with some random music instead (big mistake here!). What happened? Videos did not show, so I had to improvise opening them with another program and that did not work very well either because of a limitation on the projector’s resolution, I think.

In my case, I like to put myself in situations of technological risks and I like the feeling of doing a complex live demo in a presentation that I know can fail (I guess is like the adrenaline rush I used to have when playing in a concert). But I have to admit that the safest advice is to keep technical challenges as simple as possible. And be anal in checking many times those that you know are likely to fail.

Of course, checking technical details means, among other things, that you need to be in the room for your talk well in advance and test the presentation in the same conditions you are going to it later (even if that means missing one of the coffee breaks in the conference!).

(3) Be ready for improvising

And my last piece of advice may seem to contradict the rest. I have been talking about the importance of preparing many details of the talk. However, a presentation should always leave room for improvisation and adaptation. There is nothing worse than the feeling that the speaker has learned the conference by heart, and is not making any attempt to connect with the audience and the context. Besides, there might be elements during the talk that might force you to improvise: a technical issue, different audience that you expected, a reaction from somebody…

You should be able to put any kind of external element into your presentation while not losing the main message. I don’t like having to skip slides since it gives the impression that you are in a rush to finish, but many times there is no alternative: You might have lost precious time in trying to play that video, or maybe went too far in the introduction and now you need to cut short. It is again very important that you have a clear picture of what the “main message” is and improvise by skipping those slides that are not needed to understand it.

[![](http://localhost:8080/wordpress/wp-content/uploads/2022/06/microphone.jpg)](http://2.bp.blogspot.com/_xAtUP4Gu6Zk/TO2vKopEwmI/AAAAAAAAAJ4/kljMWuizEUk/s1600/microphone.jpg)
I hope that some of this advice is useful in your next presentations. But I would like to hear from you: how do you prepare your talks? Any tips or suggestions you want to share in the comments?

```
---
id: 92
title: 'On Trust Networks and Gamification. Or How Quora can overcome its Hype and embrace long-term Success'
date: '2011-01-31T21:26:00+00:00'
author: 'Xavier Amatriain'
##layout: post
permalink: /on-trust-networks-and-gamification-or/
header:
  teaser: https://amatriain.net/blog/images/14-01.png
categories:
    - Uncategorized
tags:
    - 'game dynamics'
    - quora
    - 'Social Networks'
    - trust
---
```

If you are reading this blog I am pretty sure that you know quite a lot about [Quora](http://www.quora.com/) by now. If not, you should sign on and try it a bit before you continue reading the post.

I have to admit it, the first time I saw Quora I thought it looked like a watered-down version of [stackoverflow](http://http//stackoverflow.com/) , only with a much broader scope. The ability to follow was nice but… “big deal”, I thought. However, I was missing the important point of the seamless integration between Quora and existing OSN, namely Twitter and Facebook. I always say that for an OSN to succeed it needs to ride on all the previous successful ones (including email if you allow me to stretch the definition of OSN that far), but I missed that part in quora until its hype began. Having quick connection to Twitter and Facebook, allowed Quora to overcome the always feared cold-start problem. You sign to Quora and in no time you are “connected” to all your “friends” and can start following their questions, their answers, votes… cool!

Well, so it seemed. But in no time, just as quick as people starting hyping about the service they were complaining about it and predicting its failure. This [recent post at Techcrunch](http://techcrunch.com/2011/01/31/quora-quora-quora-quora-quora-quora-quora/) does a pretty good at summarizing and linking to the main Quora bitchmemes. Don’t miss the[original post by Vivek Wadhwa](http://techcrunch.com/2011/01/23/why-i-don%E2%80%99t-buy-the-quora-hype/) or some of the [threads](http://www.quora.com/What-can-be-said-to-Vivek-Wadhwas-criticism-on-TechCrunch-Why-I-Don%E2%80%99t-Buy-the-Quora-Hype) at Quora itself. You should also read [this very illustrative piece](http://scobleizer.com/2011/01/30/why-i-was-wrong-about-quora-as-a-blogging-service/) on how Scobble went from love to hate in a matter of weeks.

To summarize, the two biggest complaints are the following: (1) Quora will inevitably be overtaken by spam and there will be no way to find good content anymore; and (2) producers of good content (answers) will become tired of the system and progressively leave making problem (1) even more inevitable.

While I do agree that these (and many other) issues are very important, I don’t see them as inevitable and, in the following paragraphs, I would like to describe two ways to address them. But to start with, let me just state that believing that Quora can survive on being an inside-moderated network is not the answer. So what can be done?

<span style="font-weight: bold;font-size:130%;">Trust networks
</span>

In a trust network, nodes (users) have an associated trust value that is somehow used to decide how its contribution will be taken into account by the rest of users. For instance, in a recommender system, I can push content by neighboring nodes I trust while filtering out that coming from nodes with a lower trust value. In more sophisticated versions, trust is not a unique value but can be topic-specific. That is, my trust value can be very high for independent music but very low for classic literature. (If you are interested in the general topic of Trust and Social Networks you can read [Golbeck’s book](http://www.amazon.com/exec/obidos/ASIN/1848003552/j16t3i5j15-20) or any of her many publications or presentations available online)

So let’s go back to Quora now: why should they implement a trust network overlay? and, how could they implement a useful one? There are several reasons for why they should be doing so. But let us focus on the spam issue. You do not want for bad answers to get promoted by bad/evil users. The way around it is to not give these users the power to promote answers. And you can do this quite easily by assigning trust values to users. It would take 100 votes by “level 1” users to get an answer to the level of another one with just one vote by a “level 100” user. Of course, as I was mentioning before, this trust level could be topic-sensitive. Makes sense, doesn’t it?

But, there a number of issues that are still unsolved on how to implement this trust network. The first one is who decides to promote and demote users? My answer is quite simple: users themselves. Whenever your answer gets voted up/down so would your trust level. And again, how much this level would go up/down would depend on the trust level of the voting user.

The only important remaining issue to such an approach is how to deal with the cold-start issue. But the answer to this would come from the integration to other OSN I was mentioning at the beginning. If I were implementing this kind of system, I would give users an initial trust level based on their [TunkRank](http://tunkrank.com/) or their [Klout](http://klout.com/) Score.

<span style="font-weight: bold;font-size:130%;">Gamification
</span>
The other major issue that still needs to be tackled is how we guarantee that users do not become tired of the system and abandon it. I hope it is clear by now that the approach I described above would make things much more interesting for users interested in promoting their trust level. In fact, this is very close to what is known as [gamification](http://gamification.org/wiki/Gamification) (see also game dynamics or game mechanics for very related concepts). Attach a badge to given levels of trust for some topics and you can start competing with Foursquare check-ins.

The use of badges, or game dynamics in general in Q&amp;A sites is by no means new. Actually, stackoverflow, that I was referring to earlier in the post, delivers topical [badges](http://stackoverflow.com/badges). And obtaining the first badge on a given topic can be an important accomplishment worth noting in your resume. But stackoverflow did not come up with this idea out of the blue: levels of expertise in forums have been used for a long time (see the[Coffee Cups/Beans in Ubuntu forums](http://ubuntuforums.org/announcement.php?f=48), for instance).

I am not saying that implementing these two approaches would guarantee Quora’s success. But not implementing them will probably guarantee the opposite. We have all seen potential in Quora. Apart from the quick integration with existing OSN and the ability to follow, there is the real-time component that brings it closer to a Q&amp;A Twitter. If they don’t fix these potential issues, somebody else will come up with an improved version that can very well be the “next big thing” that some social media gurus were seeing in Quora just some weeks ago.
```
---
id: 91
title: 'Managing Research the Agile Way'
date: '2011-03-15T10:56:00+00:00'
author: 'Xavier Amatriain'
##layout: post
permalink: /managing-research-agile-way/
categories:
    - Uncategorized
tags:
    - agile
    - 'extreme Programming'
    - mamagement
    - research
    - scrum
---
```

I have discussed previously on this blog about how well the [Scientific Method adapts to Agile approaches](http://localhost:8080/wordpress/2008/06/agile-research.html). These ideas also took me to an unfinished effort to draft an [Agile Research Manifesto](http://localhost:8080/wordpress/2009/06/very-draft-agile-research-manifesto.html). However, by talking to several people with similar ideas, I realized that these attempts were largely interpreted as an intellectual exercise with little practical application. It is clearly my fault for not having explained that all of this in reality comes from many practical experiences. Some of these experiences go back to my PhD years when managing the [CLAM framework](http://clam-project.org/), as well as many undergrad student projects. As a matter of fact, during those days I published a practical guide for students on how to do their final project the “agile way” (I still keep the [webpage](http://xavier.amatriain.net/PFC/), in catalan, for historical reasons).

In any case, in this post I wanted to address the practical side of agile research management by giving you a flavor of how I try to manage projects.

<div style="text-align: center;">[![The Plug-Hole](http://farm2.static.flickr.com/1077/754511201_3067a868d7.jpg)](http://www.flickr.com/photos/tonymangan/754511201/ "The Plug-Hole por ~~Tone~~, en Flickr")
<span style="font-size:78%;">(Picture by [~Tone](http://www.flickr.com/photos/tonymangan/))</span></div>
<span style="font-weight: bold;">The anatomy of a research project</span>

What am I talking about when I say a “research project”? Although they might be completely different in theme and even scope, all of the projects that I have in mind when explaining the agile management approach should share at least some of the following properties:

- <span style="font-weight: bold;">Small-sized team</span>: It is very likely that we are dealing with a one or two researchers team. A 3-4 people research team can already be considered large in my experience.
- <span style="font-weight: bold;">Very open and imprecise requirements</span>: Especially at the beginning, we might have a coarse idea or hypothesis to validate. However, the approach, method, and scope, are likely to be undecided until very late in the game.
- <span style="font-weight: bold;">High risk</span>: By definition, a research project has to be highly innovative and therefore… risky. Our goal is to minimize the cost of a failure and realize early on but not to remove failure since this is an intrinsic feature of risk.
- <span style="font-weight: bold;">Imprecise resources</span>: The fact that requirements are not clear and risk is high is usually accompanied by the fact that resources that can be allocated to the project are usually imprecise. If the project is highly successful and proves its interest in the first iterations, it can grow into something larger with more resources added to it. On the other hand, it is also very likely to be killed quickly if it does not yield promising initial results.

<span style="font-weight: bold;">The planning game</span>

I will usually start-off by devoting a couple of weeks to a <span style="font-style: italic;">Sprint 0</span> during which the main tasks will be:

- <span style="font-weight: bold;">Understand what has been done before</span>: Obviously, this requires lots of reading. However, it is good practice to also start writing at this same time, maybe in an informal wiki or the like.
- <span style="font-weight: bold;">Define the tools</span>: Unless you are in a very specific environment, tools are likely to change for every project. Sometimes it is not only about what is the best tool, but also about what the team is most familiar with. This is usually an important thing in most projects, but it is more so in a project that it is high risk in nature and should avoid spending lots of time/resources in adapting to new tools.
- <span style="font-weight: bold;">Define the initial scope</span>: There is no way you can have a complete picture of what is going to be the output of the project by this time. However, you should be able to list what you think will be the main steps and even some findings you anticipate. This list should be written like an ever changing Product Backlog (prioritized list of high-level features).

<div style="text-align: center;">[![Fifteen accounts of life, death, and everything that interferes.](http://localhost:8080/wordpress/wp-content/uploads/2022/06/2329783873_3dc3c6a550.jpg)](http://www.flickr.com/photos/babyowls/2329783873/ "Fifteen accounts of life, death, and everything that interferes. por Jenna Carver, en Flickr")</div><div style="text-align: center;"><span style="font-size:78%;"><span><span>(Picture by [Jenna Carver](http://www.flickr.com/photos/babyowls/))</span></span></span></div><span style="font-weight: bold;">Prioritizing</span>

<span>One of the most important activities that you end up doing when planning any project, be it at the initial phase or at any of its iterations, is prioritizing the different requirements, stories… Doing this in a group meeting is a great way to gain insights on the project and to be strategic. Prioritizing tasks is not much different from any cost/benefit analysis: you measure cost, you measure benefit, and then sort items according to benefit/cost ratio.</span>

In the case of project planning, I usually like to assign cost to “complexity”, and benefit to “interest”. In other words, the cost of a feature or story will be how difficult or complex we anticipate it is to implement it. And the benefit is how interesting or important it is for our final goal. Once you sort items using the interest/complexity ratio, you will find that easy-to-do yet interesting features float to the top, while complex and not so important sink down to the bottom.

Of course, the interesting discussions happen right in the middle. And especially when we have something that seems to be very important, but also very complex to achieve. In these cases, we feel tempted to jump right away at the problem and devote 100% of our energy to it. However, one of the agile principles is that things seem more complex when you don’t have enough understanding. If you put them off to later iterations, they will eventually become clearer and clearer and end up surfacing to the first positions on your priority list. I have found this sort of <span style="font-style: italic;">smart procrastination</span> to be extremely useful for agile research management.

<span style="font-weight: bold;">Iterate, Iterate, Iterate</span>

Once you have come up with your initial product backlog, it is all a matter of breaking the process down in short iterations – I usually plan for one-week. At the beginning of each iteration (or <span style="font-style: italic;">Sprint</span>), you look at your product backlog, pick some of the top stories and break them down into finer grain tasks. You do the prioritization game on this new list and come up with your next week’s scrum/iteration backlog.

When doing this finer-grain prioritization, I have found it very useful to use the estimated number of hours as the measure of “complexity”. Therefore, when picking the top tasks of our list, we will also have an estimate of how feasible it is to have them during this iteration and how much will be the relative effort put into each of them. And, if any task is estimated to be more than a day long, do yourself a favor and break it into several tasks.

Also, it is important that, especially during the first iterations, you realize that the continuation of the project might be at stake at each iteration (or at least the current approach). Therefore, when measuring the “importance” of tasks to prioritize, ask yourself how relevant will that task be to convince you and others that you are onto something or you need to change routes.<span style="font-weight: bold;"></span>

<span style="font-weight: bold;">Test-driven research</span>

<span>If you are familiar with agile methods, you will probably know how important testing is in an agile project. Tests not only guarantee the stability of the project but are actually a way to specify requirements in a more verifiable form. In a similar way, you can think of specifying many of your research hypothesis as a test. For example, you can turn your hypothesis that the effect of a given procedure on your data or population is significant by a verifiable assertion that for t-test(D_original, D_after_procedure)</span>-&gt; p is smaller 5%. There are many hypothesis and research tasks that can – and should – be written in this form before making it to your prioritized todo list. At least you should worry on how any of your results will be validated and how you can trust for them to be consistent and significant.

<div style="text-align: center;"> [![*splooch!* Gordon Freeman vs. Master Chief](http://localhost:8080/wordpress/wp-content/uploads/2022/06/3385957841_85bf7fcca6.jpg)](http://www.flickr.com/photos/dunechaser/3385957841/ "*splooch!* Gordon Freeman vs. Master Chief por Dunechaser, en Flickr")
<span style="font-size:78%;">(Picture by [Dunechaser](http://www.flickr.com/photos/dunechaser/))</span>
</div><span style="font-weight: bold;">Related approaches</span>

If you are interested in this kind of approaches, I recommend you read the article on the [SCORE method](http://cacm.acm.org/magazines/2010/10/99484-score-agile-research-group-management/fulltext), which is somewhat related to many of the things I am mentioning here. [Here](http://agile2003.agilealliance.org/files/P6Paper.pdf) you can read an interesting paper on doing test-driven research. Finally, I find the [Pomodoro](http://www.infoq.com/news/2009/09/Pomodoro) method a very interesting approach to individual time management. Since many research projects end up being quasi-individual, Pomodoro fits them pretty well.---

```
id: 90
title: 'The Science and the Magic of User Feedback'
date: '2011-03-18T15:25:00+00:00'
author: 'Xavier Amatriain'
##layout: post
permalink: /science-and-magic-of-user-feedback/
categories:
    - Uncategorized
tags:
    - 'explicit feedback'
    - noise
    - 'recommender systems'
---
```

That was the main title of a series of talks I gave in different labs and companies during my recent California tour. In this presentation, I talked about many of our recent projects related to how to interpret user feedback, in general, and in the particular case of recommender systems. I talked about our work on [measuring user rating noise](http://localhost:8080/wordpress/2009/04/i-like-it-i-like-it-not-or-how-miss.html), our follow-up in devising [algorithms to reduce this natural noise](http://localhost:8080/wordpress/2009/08/rate-it-again.html), and on how you can use[experts instead of crowds](http://localhost:8080/wordpress/2009/05/wisdom-of-few.html) to not only minimize this noise but address other issues in collaborative filtering.

I also gave a sneak preview of our results to the[music survey](http://localhost:8080/wordpress/2010/08/study-on-online-music-taste-call-for.html) I announced some time ago. [Denis Parra](http://www.sis.pitt.edu/%7Edparra/) and I have submitted this work recently and are hoping to get it accepted to tell you a bit more about how to map implicit to explicit feedback.

## Slides:

<iframe src="//www.slideshare.net/slideshow/embed_code/key/ajzix3INI8C6Rj" width="595" height="485" frameborder="0" marginwidth="0" marginheight="0" scrolling="no" style="border:1px solid #CCC; border-width:1px; margin-bottom:5px; max-width: 100%;" allowfullscreen> </iframe> <div style="margin-bottom:5px">

```
<strong> <a href="//www.slideshare.net/xamat/the-science-and-the-magic-of-user-feedback-for-recommender-
systems" title="The Science and the Magic of User Feedback for Recommender Systems" target="_blank">The
Science and the Magic of User Feedback for Recommender Systems</a> </strong> from <strong><a
href="//www.slideshare.net/xamat" target="_blank">Xavier Amatriain</a></strong> </div>

## Video:

<iframe src="https://player.vimeo.com/video/22353044?h=239c114666" width="640" height="360" frameborder="0"
allow="autoplay; fullscreen; picture-in-picture" allowfullscreen></iframe>
<p><a href="https://vimeo.com/22353044">Tech Talk: Xavier Amatriain (Telefonica) -- &quot;The Science and
Magic of User and Expert Feedback for Improving Recommendations&quot;</a> from <a
href="https://vimeo.com/talksatlinkedin">Talks at LinkedIn</a> on <a href="https://vimeo.com">Vimeo</a>.</p>
---
id: 88
title: 'Recommender Systems: We&#8217;re doing it (all) wrong'
date: '2011-04-07T20:59:00+00:00'
author: 'Xavier Amatriain'
##layout: post
permalink: /recommender-systems-were-doing-it-all/
header:
  teaser: https://amatriain.net/blog/images/fig2b.jpg
categories:
    - Uncategorized
tags:
    - error
    - 'recommender systems'
    - stats
---

A few days back, there was an interesting post by Judy Robertson in the Communications of the ACM blog. The
post, entitled “[Stats: We’re doing it wrong](http://cacm.acm.org/blogs/blog-cacm/107125-stats-were-doing-
it-wrong/fulltext)“, builds upon a paper from last year’s CHI conference in which they report that more than
90% of the HCI researchers used the wrong statistical tools when analyzing and reporting on likert scale
type of data. A Likert scale is a unidimensional scale on which the respondent expresses the level of
agreement to a statement – typically in a 1 to 5 scale in which 1 is strongly disagree and 5 is strongly
agree.

Here is an excerpt from the post that I think is worth highlighting:

> Likert scales give ordinal data. That it (sic), the data is ranked “strongly agree” is usually better than
“agree.” However, it’s not interval data. You can’t say the distances between “strongly agree” and “agree”
would be the same as “neutral” and “disagree,” for example. People tend to think there is a bigger
difference between items at the extremes of the scale than in the middle (there is some evidence cited in
Kaptein’s paper that this is the case). **For ordinal data, one should use non-parametric statistical
tests** which do not assume a normal distribution of the data. **Furthermore, because of this it makes no
sense to report means of likert scale data—you should report the mode**.

As Judy, I have to admit that I am not a stats expert myself either. But in the general case I would agree
with the previous: likert scale data is ordinal and cannot be treated as interval. However, whether treating
it as interval is <span style="font-weight: bold;">always</span> a mistake or can be accepted under some
circumstances is something that I am not sure and relates to the rest of this post.

So for instance, it is not uncommon to find references where they clearly state that likert data can be
treated as interval. For example, look at what they say in [this handbook]
(http://www.fao.org/docrep/W3241E/w3241e04.htm) edited by the FAO.

> Likert scales are treated as yielding Interval data by the majority of marketing researchers.

Or look at [the answer](http://stats.stackexchange.com/questions/10/under-what-conditions-should-likert-
scales-be-used-as-ordinal-or-interval-data) to the question of whether likert data can be treated as
interval in stackexchange.

So there might be some circumstances in which, depending on the analysis, likert could be treated as
interval… I guess. But not in the general case.

<span style="font-weight: bold;">Implications for Recommender Systems</span>

Now onto the big question: What does this have to do with Recommender Systems and how does it affect? To
start with, let me ask you the question: Does the likert (1 to 5) scale relate to anything we use in
recommender systems? You got it: <span style="font-weight: bold;">ratings</span> !

So our worry goes now to understanding whether ratings can be treated as interval or they should instead be
treated as ordinal data, just as they are in the general case of the likert scale. In order to defend that
ratings can be treated as interval, we should have some validation that the distance between different
ratings is approximately equal. However, just as in the case of likert scales, we know this is not the case.
```

Look at this figure from [our previous work](http://localhost:8080/wordpress/2009/04/i-like-it-i-like-it-not-or-how-miss.html) on measuring noise in ratings.
![](file:///home/xavier/Sandbox/data/articles/xamat_UMAP09/figs/fig2b.jpg)[![](http://localhost:8080/wordpress/wp-content/uploads/2011/04/fig2b.jpg)](http://4.bp.blogspot.com/-0BzBRalrIDo/TZ4xEitd5EI/AAAAAAAAAKs/1l2M5Og6dEo/s1600/fig2b.jpg)

Here we are plotting the probability of finding different kinds of inconsistencies between pairs of ratings. The probability that a user changes her rating between 2 and 3 is almost 0.35 while the probability she changes between 4 and 5 goes down to almost 0.1. This is a clear indication that users perceive that the distance between a 2 and a 3 is much lower than between a 4 and a 5.

<span style="font-weight: bold;">Consequences</span>

At this point, we can safely say that ratings are ordinal but not interval data. However, they are treated as a continuous interval scale in most of the recommender systems research! Let us stop to think a few of the consequences of ratings not being interval data.

<span style="font-weight: bold;">Distance Measures:</span> All the neighbor based methods in collaborative filtering are based on the use of some sort of distance measure. The most commonly used are Cosine distance and Pearson Correlation. However, both these distances assume a linear interval scale in their computations! We should conclude that using these distance measures with rating data is wrong. Other measures such as [Spearman’s rank correlation](http://en.wikipedia.org/wiki/Spearman%27s_rank_correlation_coefficient), do not assume this. But to be honest, I don’t remember having read many papers using Spearman.

<span style="font-weight: bold;">Error Measures:</span> This is my favorite one… The most commonly accepted measure of success for recommender systems is the Root Mean Squared Error (RMSE). But wait, this measure is explicitly assuming that ratings are also interval data! Similar error measures such as MAE also fall in the same trap… banned! So what could we use? Standard Information Retrieval measures such as Precision and Recall do not necessarily assume interval scale on the ratings, although their mapping to recommendation efficiency may also be questioned. Rank-based measures such as [Discounted Cumulative Gain](http://en.wikipedia.org/wiki/Discounted_cumulative_gain) (nDCG) seem like our best bet for now.

<span style="font-weight: bold;">Matrix factorization</span>: Most MF techniques in Recommender Systems are in fact optimizing for RMSE. Therefore, we should discard them as statistically incorrect for the same reasons stated above. There are interesting alternatives to this though, like the [PureSVD](http://research.yahoo.com/files/recsys2010_submission_150.pdf) method presented in Recsys last year, that do not optimize for RMSE but rather for ranking.

<span style="font-weight: bold;">Conclusion</span>

It is clear that explicit ratings, just like likert scale data, have to be treated like ordinal (and not interval data). However, most of the methods and measures currently in use in recommender systems assume in some sense that there is a continuous linear scale in the ratings. Of course I am not advocating for throwing all of this research to the trash (among other things, it would include much of mine), but I would advice for a drastic change in the way we approach these issues.

I am writing this post, especially in the hope to get feedback and reactions from you. So I am looking forward to the comments.

<span style="color: rgb(204, 0, 0); font-weight: bold;">Update:</span> This post was featured in Ycombinator Hacker News. So far it has received over 6K views and there is a somewhat interesting [comment thread](http://news.ycombinator.com/item?id=2423313) in Ycombinator.

(I’d like to thank and acknowledge the contribution of [Denis Parra](http://www.sis.pitt.edu/%7Edparra/), [Alexandros Karatzoglou](http://www.ci.tuwien.ac.at/%7Ealexis/Welcome.html), and [Rodrigo Oliveira](http://www.ic.unicamp.br/%7Eoliveira/) to this post through previous very fruitful discussions)
```
---
id: 86
title: 'Walk the Talk: On the Combination of Implicit and Explicit Feedback'
date: '2011-07-19T05:32:00+00:00'
author: 'Xavier Amatriain'
##layout: post
permalink: /walk-talk-on-combination-of-implicit/
header:
  teaser: https://amatriain.net/blog/images/up-box.png
categories:
    - Uncategorized
tags:
    - explicit
    - 'explicit feedback'
    - implicit
    - 'music recommenders'
    - 'recommender systems'
    - 'user modeling'
---
```

Last week, [Denis Parra](http://www.sis.pitt.edu/%7Edparra/) presented our paper entitled “Walk the Talk: Analyzing the Relation between Implicit and Explicit Feedback for Preference Elicitation” at the [UMAP conference](http://www.umap2011.org/). The paper won Denis the best-student paper award (Congratulations!).

The paper presents our initial work in analyzing the relation between implicit and explicit feedback. In short, the main question we wanted to answer is how does the self-reported preferences users give in a typical 5-star interface relate to what they actually do when looking at their consumption patterns. Our hypothesis was that there should exist simple models that relate both kinds of feedback. Finding a way to robustly convert implicit feedback into explicit ratings would open up the door to applying well-known methods with implicit feedback. But, much more importantly, we could then combine both kinds of input in a single model.

In order to test our hypothesis, we prepared an experiment in the music domain. We asked last.fm users to take a [survey](http://localhost:8080/wordpress/2010/08/study-on-online-music-taste-call-for.html) in which we queried them about how much they liked albums that were already in their listening history. With this data in hand, we could analyze the relation between implicit and explicit feedback and try to fit a simple model.

I recommend you read the [full paper](http://bit.ly/r1mvkK) if you want to get the longer story of our findings, but here is a brief summary:

- There is a strong correlation between implicit feedback and self-reported preference (see figure below)
- Variables such as recentness of interaction or overall popularity do not have significant effect. Note that in [a previous study](http://www.princeton.edu/%7Emjs3/salganik_watts08.pdf) by Salganik &amp; Duncan Watts, global popularity was found to affect users perceived quality. However, in that case and as opposed to ours, users were made aware of the popularity.
- Interaction effect: When listening to music, some people prefer to listen to isolated songs or albums. The way they interact with music, affects the way they report their taste.

[![](http://localhost:8080/wordpress/wp-content/uploads/2011/07/up-box.png)](http://1.bp.blogspot.com/-gtrwSvrIpEI/TifEaf_CfYI/AAAAAAAAANI/IzeMVpRZbaQ/s1600/up-box.png)
After our analysis, we then construct a linear model that takes into account these variables by performing a linear regression. Once we have built these models, we can evaluate their performance in a regular recommendation scenario by measuring the error in predicting ratings in a hold-out dataset.

This paper represents an initial but very promising line of work that we have already improved in several ways such as the use of logistic instead of linear regression to account for the non-linearity of the rating scale or the use of the regression model as a way to combine both implicit and explicit feedback. But I will leave those findings for a future post.

```
---
id: 84
title: 'Joining Netflix'
date: '2011-07-29T04:49:00+00:00'
author: 'Xavier Amatriain'
##layout: post
permalink: /joining-netflix/
header:
  teaser: https://amatriain.net/blog/images/2011-07-202B08.19.53.jpg
categories:
    - Uncategorized
tags:
    - netflix
    - 'recommender systems'
---
```

Three weeks ago, I started to work for Netflix. Everything has moved so fast with so many things to do and learn that it seems like I have already been here for a much longer time!

I am now working as the manager of a small team working on recommendations &amp; personalization in the company that promoted recommender systems research to major headlines thanks to the [Netflix Prize](http://www.netflixprize.com/). It also feels great to come to the company in an exciting time when it has just reached its 25th million customer and is starting its international expansion to [Latin America](http://blog.netflix.com/2011/07/netflix-is-coming-to-latin-america.html).

All the fuzz created around the Netflix Prize [might lead some](http://news.cnet.com/8301-17852_3-20078504-71/mit-prof-netflix-has-its-recommendations-wrong/) to believe that rating prediction is all there is to Netflix suggesting a given movie. However, I was happy to find out that rating prediction is only one of the many signals that my team uses in creating the final suggestions.

Awesome place, awesome people, and awesome time to be around. And, btw, [we are hiring](http://www.netflix.com/Jobs?id=7563), so let me know if you are interested in joining. (<span style="font-weight: bold;">Update</span>: it seems that the jobs link is currently not active outside US/CA… I’m working on getting this fixed)

[![](http://localhost:8080/wordpress/wp-content/uploads/2011/07/2011-07-202B08.19.53.jpg)](http://3.bp.blogspot.com/-EU1JNoF8ZXI/TjI9TqhAlmI/AAAAAAAAANQ/WKPl4DaPcds/s1600/2011-07-20%2B08.19.53.jpg)

```
---
id: 82
title: 'The Recommender Problem &#038; the Presentation Context'
date: '2011-09-26T06:41:00+00:00'
author: 'Xavier Amatriain'
##layout: post
permalink: /recommender-problem-presentation/
header:
  teaser: https://amatriain.net/blog/images/NetflixInterface.jpg
categories:
    - Uncategorized
tags:
    - interaction
    - 'presentation data'
    - 'recommender systems'
---
```

<span style="font-size: 12pt;">In the traditional formulation of the “Recommender Problem”, we have pairs of items and users and user feedback values for very few of those dyads. The problem is formulated as the finding of a utility function or model to estimate the missing values.</span>

In many real-world situations, feedback will be implicit<span style="font-weight: bold;">**</span> and binary in nature. For instance, in a web page you will have users visiting a url, or clicking on an add as a positive feedback. In a music service, a user will decide to listen to a song. Or in a movie service, like Netflix, you will have users deciding to watch a title as an indication that the user liked the movie. In these cases, the recommendation problem becomes the prediction of the probability a user will interact with a given item. There is a big shortcoming in using the standard recommendation formulation in such a setting: we don’t have negative feedback. All the data we have is either positive or missing. And the missing data includes both items that the user explicitly chose to ignore because they were not appealing and items that would have been perfect recommendations but were never presented to the user.

A similar issue has been dealt with in traditional data mining research, where classifiers need to be trained only using positive examples. In the ["Learning Classifiers from Only Positive and Unlabeled Examples"](http://www.cse.ucsd.edu/users/elkan/posonly.pdf) SIGKDD 08 paper, the authors present a method to convert unlabeled examples into both a positive and a negative example, each with a different weight related to the probability that a random exemplar is positive or negative. Another solutions to this issue is presented in the “[Collaborative Filtering for Implicit Feedback Datasets](http://research.yahoo.com/files/HuKorenVolinsky-ICDM08.pdf)” paper by Hu, Koren and Volinsky. In this work, the authors binarize the implicit feedback values: any feedback value greater than zero means positive preference, while any value equal to zero is converted to no preference. A greater value in the implicit feedback value is used to measure the “confidence” in the fact the user liked the item, but not in measuring “how much” the user liked it. Yet another approach to inferring positive and negative feedback from implicit data is presented in the paper I co-authored with Dennis Parra, and I presented in a ["previous pos](/_posts/walk-talk-on-combination-of-implicit.html). There, we argue that implicit data can be transformed to positive and negative feedback if aggregated at the right level. For example, the fact that somebody listened only once to a single track in an album can be interpreted as the user not liking that album.

In many practical situations, though, we have more information than the simple binary implicit feedback from the user. For unlabeled examples that the user did not directly interact with, we can expect to have other information. In particular, we might be able to know whether they were shown to the user or not. This adds very valuable information, but slightly complicates the formulation of our recommendation problem. We now have three different kinds of values for items: positive, presented but not chosen, and not presented. And this is only if we simplify the model. In reality, information related to the presentation can be much richer than this and we might be able to derive data like the probability the user actually saw the item or weigh in different interaction events such as mouse overs, scrolls…

| ![](/blog/images/NetflixInterface.jpg) |

At Netflix, we are working on different ways to add this rich information related to presentations and user interaction to the recommender problem. That is why I was especially interested in finding out that this year’s SIGIR best student paper award has been awarded to a paper that addresses this issue. In the paper [“Collaborative Competitive Filtering: Learning Recommender Using Context of User Choice"](http://www.cc.gatech.edu/%7Esyang46/papers/SIGIR11CCF.pdf), the authors present an extension to traditional Collaborative Filtering by encoding into the model not only the **collaboration** between similar users and items, but also the **competition** of items for user attention. They derive the model as an extension to standard latent factor models by taking into account the context in which the user makes the decision. That is, the probability I decide to select a given item depends on which are the other items I have as an alternative. Results are preliminary but promising. And, this work is definitely an interesting and appealing starting point for an area with many practical applications.

However, there are many possible improvements to the model. One of them, mentioned by the authors, is the need to take into account the so-called **position bias**. An item that is presented in the first position of a list has many more possibilities to be chosen than one that is farther down. This effect is well-known in the search community and has been studied from several angles. I would recommend, for instance to read some of the very interesting papers on this topic by Thorsten Joachims and his students. In the paper [“How Does Clickthrough Data Reflect Retrieval Quality?"]

(http://www.cs.cornell.edu/People/tj/publications/radlinski_etal_08b.pdf), for instance, they show how arbitrarily swapping items in a search result list has almost no effect. This proves that the positioning of the element can be a most important factor than how relevant the item is.

I would love to hear of other ideas or approaches to deal with this new version of the recommender problem that includes, and would encourage researchers in the area to address an issue of huge potential impact.

***Note**: I am using the word implicit here in the traditional sense in the recommendation literature. The truth is that a user selecting an item is in fact **explicit** information. However, it can be considered implicit in that the user is informing about the preferences indirectly by comparing the item to others in a context.

```
---
id: 81
title: 'Recsys 2011 &#8211; Notes and Pointers'
date: '2011-11-03T04:29:00+00:00'
author: 'Xavier Amatriain'
##layout: post
permalink: /recsys-2011-notes-and-pointers/
header:
  teaser: https://amatriain.net/blog/images/14-01.png
categories:
    - Uncategorized
tags:
    - conference
    - 'recommender systems'
    - recsys11
---
```

[![](http://recsys.acm.org/2011/images/Chicago_night1.jpg)](http://recsys.acm.org/2011/images/Chicago_night1.jpg)
I found [Recsys](http://recsys.acm.org/2011/index.shtml) this year of very high quality in general. There were many good papers and presentations. The [Industry track](http://recsys.acm.org/2011/industry_track.shtml) was also very high-quality, with very interesting talks from companies such as Twitter, Facebook, or eBay. Jon Sanders and I also gave two presentations explaining how recommendations have evolved since the Netflix Prize (more on this soon).

Here are my rough notes with pointers to some papers I considered especially interesting. I have grouped them in 5 categories that I think summarize the main topics in the conference: (1) Transparency and explanations, (2) Implicit Feedback, (3) Context, (4) Metrics and evaluation, and (5) Others. Note that the selection is completely biased towards my personal interests.

<span style="font-weight: bold;">(1) TRANSPARENCY &amp; EXPLANATIONS.</span> One of the recurring themes was the fact that user trust and perceived quality of the recommendations was very much influence not by accuracy alone, but by how transparent the system was, and the amount of “explanations” that were added.

- Daniel Tunkelang(LinkedIn) did a very interesting tutorial on “Recommendations as a Conversation with the User”, where he focused on these kinds of issues. See his slides in [his blog](http://thenoisychannel.com/2011/10/31/recsys-2011-tutorial-recommendations-as-a-conversation-with-the-user/).
- Neel Sundaresan (eBay) also stressed in his keynote that adding explanations can sometimes be more important than getting the recommendation right.
- In the paper “[Each to His Own: How Different Users Call for Different Interaction Methods in Recommender Systems](http://www.usabart.nl/portfolio/KnijnenburgReijmerWillemsen-recsys2011.pdf)“, the authors found that depending on how experts are users in the domain, they prefer different kind of recommendations and interaction models. For example, in one of the extremes, novices, prefer top-10 non-personalized to their personalized recommendations. In general a hybrid model of interaction is better than either implicit or explicit-only.

<span style="font-weight: bold;">(2) IMPLICIT FEEDBACK.</span> A lot of papers this years on using implicit consumption data instead of (or in combination with) ratings.

- The best paper, by Yehuda Koren and Joe Sill, addressed the issue of non-linearity in ratings. “[OrdRec: An Ordinal Model for Predicting Personalized Item Rating Distributions](http://labs.yahoo.com/node/640)” modifies the standard Matrix Factorization approach to adapt to the fact that user ratings are ordinal, but not numerical. The way they model ratings, with a set of thresholds, can be used in combination with any model, not only SVD-like approaches. This paper effectively addresses most of the issues I raised in my previous post “[We are doing everything wrong…](http://localhost:8080/wordpress/2011/04/recommender-systems-were-doing-it-all.html)“
- In “[Modeling Item Selection and Relevance for Accurate Recommendations: A Bayesian Approach](http://unical.academia.edu/NicolaBarbieri/Papers/803078/Modeling_Item_Selection_and_Relevance_for_Accurate_Recommendations)” they define the concept of a “Free probabilstic model” where they try to predict independently the probabilty of play and rating.
- In “Multi-Value Probabilistic Matrix Factorization for IP-TV Recommendations”, the authors present a Matrix Factorization model that allows for multiple observations of the same item. In particular, it is applied for IPTV recommendations where the fact that the user watched part of an episode is interpreted as negative feedback.

- “[Matrix Co-factorization for Recommendation with Rich Side Information and Implicit Feedback](http://www.cs.purdue.edu/homes/fangy/hetrec11-fang.pdf)” presents a combined Matrix Factorization model that includes ratings, content features, and implicit feedback. They use cosine item similarity for weighing negative examples.
- In “[Personalizing Tags: A Folksonomy-like Approach for Recommending Movies](http://www.slideshare.net/alansaid/personalizing-tags-a-folksonomylike-approach-for-recommending-movies/download)“, they use tags (or categories) as a very simple method of recommending movies: for each user compute average rating given to movies with a certain tag.

<span style="font-weight: bold;">(3) CONTEXT.</span> There were 2 workshops ([CARS](http://cars-workshop.org/) and [CAMRA](http://2011.camrachallenge.com/)), and several papers in the main conference, talking about how to add contextual information for the recommendations:

- “The Effect of Context-Aware Recommendations on Customer Purchasing Behavior and Trust” is an interesting paper, focusing on the evaluation side. They include an A/B test for measuring the effect of context-aware recommendations. Using context increased overall sales in $ but not in number. Therefore, users tend to spend more $ per item.
- In the [CAMRA](http://2011.camrachallenge.com/) workshop, many papers (such as “Temporal Rating Habits: A Valuable Tool for Rater Differentiation” or “Identifying Users From Their Rating Patterns”) were related to how to identify who the author of a rating in a household was, since this was one of the tasks for the contest.
- Also related to group recommendations, “Group Recommendation using Feature Space Representing Behavioral Tendency and Power Balance among Members”, tries to model what is a good recommendation for a group where each of the individuals does not have the same influence.

<span style="font-weight: bold;">(4) METRICS and EVALUATIONS: </span>There were several papers that offered different ways to measure accuracy for top-N ranked recommendations.

- “[Rank and Relevance in Novelty and Diversity Metrics for Recommender Systems](http://www.slideshare.net/pcastells/acm-recsys-2011-rank-and-relevance-in-novelty-and-diversity-metrics-for-recommender-systems)” presents an interesting framework that includes metric for measuring not only accuracy, but also novelty, diversity….
- “Item Popularity and Recommendation Accuracy” is an interesting work on how to remove popularity bias from accuracy metrics. A user study validates the fact that recall measure is correlated with user perceived quality of recommendation. Besides proposing a recall metric that removes popularity bias, he also proposes a popularity stratified training method that weights negative examples according to how popular they are.
- “[Evaluating Rank Accuracy based on Incomplete Pairwise Preferences](http://ucersti.ieis.tue.nl/files/papers/3.pdf)” proposes a measure called expected discounted rank correlation for the specific case of implicit feedback.

<span style="font-weight: bold;">(5) OTHERS</span>

- eBay and UCSC presented “[Utilizing Related Products for Post-Purchase Recommendation in E-commerce](http://users.soe.ucsc.edu/%7Ejwang30/index.files/recsys175-wang.pdf)“. The paper won the best poster award
- There were many papers on Social Recommendations. Just to name one, in “Power to the People: Exploring Neighbourhood Formations in Social Recommender Systems”, they did a user study to figure out how much users would like and trust recommendations coming from different user groups (those they decided, friends, everyone…). Interestingly, the method of choice did not make much difference… until you told the users what it was.
- In “Wisdom of the Better Few: Cold Start Recommendation via Representative based Rating Elicitation” they discussed how to select most imformative users and items for cold start. I was surprised to see that our “Wisdom of the Few” approach got paraphrased in a paper title.
- There were a couple of very interesting workshops on [Music Recommendations](http://womrad.org/2011/) and [Mobile Recommendations](http://pema2011.cs.ucl.ac.uk/) that I had to miss since I was attending others. But, they are definitely worth looking into if you are into music or mobile.

```
---
id: 80
title: 'Beyond the 5 Stars Round up'
date: '2012-04-17T05:56:00+00:00'
author: 'Xavier Amatriain'
##layout: post
permalink: /beyond-5-stars-round-up/
header:
  teaser: https://amatriain.net/blog/images/4-01.png
categories:
    - 'recsys'
tags:
    - netflix
    - 'netflix prize'
    - 'recommender systems'
---
```

Last week, I published a post on the Netflix tech blog. The post, entitled “[Netflix Recommendations: Beyond the 5 stars](http://techblog.netflix.com/2012/04/netflix-recommendations-beyond-5-stars.html)” describes how recommendations have evolved at Netflix since the Netflix Prize. If you haven’t read the post, do so now.

The post stirred quite some interesting reactions and many comments. Here is a list of some of the ones I picked up, grouped by “category”:

- Our post got picked up by the usual content aggregators:
- Hacker News posted a link to the post, and as always attracted [a great number of comments](http://news.ycombinator.com/item?id=3810058)
- Likewise, [Reddit](http://www.reddit.com/r/programming/comments/rxp2e/netflix_recommendations_beyond_the_5_stars_part_1/)also attracted some interesting comments.

- Some online media gave a summary of our post:
- Engadget was the first one to have a brief piece with several interesting comments: [Netflix explains its recommendation system, can’t find a reason for Adam Sandler’s last movie](http://www.engadget.com/2012/04/08/netflix-explains-its-recommendation-system-cant-find-a-reason/#disqus_thread)
- The CNN also had a longer article, with a detailed summary of our post: [Inside Netflix’s popular ‘recommendation’ algorithm](http://whatsnext.blogs.cnn.com/2012/04/09/inside-netflixs-popular-recommendation-algorithm/)
- Econsultancy reflects on our post by calling us [Netflix: the algorithm company](http://econsultancy.com/us/blog/9554-netflix-the-algorithm-company)

- Others focused on the figure we gave on the post related to the percentage of views coming from recommendations.
- The Verge picked up on this for their piece: [Netflix offers details on its recommendation engine, says it guides 75 percent of viewership](http://www.theverge.com/2012/4/8/2934375/netflix-recommendation-system-explained)
- Also PC Magazine focused on this issue, but gave a more detailed summary of the post: [75 Percent of Netflix Viewing Based on Recommendations](http://www.pcmag.com/article2/0,2817,2402739,00.asp)
- Finally, Business Insider, also focused on this issue in their [Netflix’s Recommendation Engine Drives 75% Of Viewership](http://www.businessinsider.com/netflixs-recommendation-engine-drives-75-of-viewership-2012-4)

- The last few reactions I picked up, focused on the reasons why we are not using the winning entry to the Netflix Prize .
- The Next Web, entitled their piece [Remember Netflix’s $1m algorithm contest? Well, here’s why it didn’t use the winning entry](http://thenextweb.com/media/2012/04/13/remember-netflixs-1m-algorithm-contest-well-heres-why-it-didnt-use-the-winning-entry/?awesm=tnw.to_1E23j)
- Techdirt focused only exclusively on this issue in their [Why Netflix Never Implemented The Algorithm That Won The Netflix $1 Million Challenge](http://www.techdirt.com/blog/innovation/articles/20120409/03412518422/why-netflix-never-implemented-algorithm-that-won-netflix-1-million-challenge.shtml)
- Both the previous articles are ok, but a bit misleading in that they seem to imply that not implementing the final solution to the prize might have been a loss for Netflix. But, the prize to the worst uninformed and ill-intentioned reaction to our post has to go to Forbes’ [What The Failed 1m Netflix Prize Tells Us About Business Advice](http://www.forbes.com/sites/ryanholiday/2012/04/16/what-the-failed-1m-netflix-prize-tells-us-about-business-advice/). It should be clear to anyone that the Netflix Prize was a huge success in all metrics. And, Netflix had already recovered the $1m long before the competition ended.
```
---
id: 78
title: 'More data or better models?'
date: '2012-07-03T06:22:00+00:00'
author: 'Xavier Amatriain'
##layout: post
permalink: /more-data-or-better-models/
header:
  teaser: https://amatriain.net/blog/images/TheUnreasonableEffectivenessOfData.png
categories:
    - Uncategorized
tags:
    - 'big data'
    - 'machine learning'
    - models
---
```

The discussion of whether it is better to focus on building better algorithms or getting more data is by no means new. But, it is really catching on lately. This was one of the preferred discussion topics in [this year’s Strata Conference](http://news.cnet.com/8301-13556_3-57389685-61/data-vs-models-at-the-strata-conference/), for instance. And, I do have the feeling that because of the Big Data “hype”, the common opinion is very much favoring those claiming that it is “all about the data”. The truth is that data by itself does not necessarily help in making our predictive models better. In the rest of this post I will try to debunk some of the myths surrounding the “more data beats algorithms” fallacy.

## The Unreasonable Effectiveness of a Misquote

Probably one of the most famous quotes defending the power of data is that of Google’s Research Director [Peter Norvig](http://en.wikipedia.org/wiki/Peter_Norvig) claiming that “We don’t have better algorithms. We just have more data.". This quote is usually linked to the article on "The Unreasonable Effectiveness of Data", co-authored by Norvig himself (you should probably be able to find the pdf on the web although [the original](http://googleresearch.blogspot.com/2009/03/unreasonable-effectiveness-of-data.html) is behind the IEEE paywall). The last nail on the coffin of models is when Norvig is misquoted as saying that "All models are wrong, and you don't need them anyway" (read [here](http://norvig.com/fact-check.html) for the author's own clarifications on how he was misquoted).

| ![](/blog/images/TheUnreasonableEffectivenessOfData.png) |

The effect that Norvig et. al were referring to in this article, had already been captured years before in the famous paper by Microsoft Researchers Banko and Brill \[2001\] "[Scaling to Very Very Large Corpora for Natural Language Disambiguation](http://acl.ldc.upenn.edu/P/P01/P01-1005.pdf)". In that paper, the authors included the plot below.

| ![](/blog/images/BankoAndBrill.png) |

That figure shows that, for the given problem, very different algorithms perform virtually the same. however, adding more examples (words) to the training set monotonically increases the accuracy of the model.

So, case closed, you might think. Well… not so fast. The reality is that both Norvig's assertions and Banko and Brill's paper are right… in a context. But, they are now and again misquoted in contexts that are completely different than the original ones. But, in order to understand why, we need to get slightly technical. I don't plan on giving a full machine learning tutorial in this post. If you don't understand what I explain below, read Andrew Ng's [Practical Advice for Machine Learning](http://cs229.stanford.edu/materials/ML-advice.pdf). Or, better still, enroll in his [Machine Learning course](https://www.coursera.org/course/ml).

## Variance or Bias?

The basic idea is that there are two possible (and almost opposite) reasons a model might not perform well.

In the first case, we might have a model that is too complicated for the amount of data we have. This situation, known as *high variance*, leads to model overfitting. We know that we are facing a high variance issue when the training error is much lower than the test error. High variance problems can be addressed by reducing the number of features, and… yes, by increasing the number of data points. So, what kind of models were Banko &amp; Brill's, and Norvig dealing with? Yes, you got it right: high variance. In both cases, the authors were working on language models in which roughly every word in the
vocabulary makes a feature. These are models with many features as
compared to the training examples. Therefore, they are likely to
overfit. And, yes, in this case adding more examples will help.

But, in the opposite case, we might have a model that is too simple to explain the data we have. In that case, known as *high bias*, adding more data will not help. See below a plot of a real production system at Netflix and its performance as we add more training examples.

| ![](/blog/images/PerformanceVsTrainingSize.png) |

So, no, **more data does not always help**. As we have just seen there can be many cases in which adding more examples to our training set will not improve the model performance.

## More features to the rescue

If you are with me so far, and you have done your homework in understanding high variance and high bias problems, you might be thinking that I have deliberately left something out of the discussion. Yes, high bias models will not benefit from more training examples, but they might very well benefit from more features. So, in the end, it is all about adding "more" data, right? Well, again, it depends.

Let's take the Netflix Prize, for example. Pretty early on in the game, there was [a blog post](http://anand.typepad.com/datawocky/2008/03/more-data-usual.html) by serial entrepreneur and Stanford professor [Anand Rajaraman](http://en.wikipedia.org/wiki/Anand_Rajaraman) commenting on the use of extra features to solve the problem. The post explains how a team of students got an improvement on the prediction accuracy by adding content features from IMDB.

| ![](/blog/images/Datawocky.png) |

In retrospect, it is easy to criticize the post for making a gross over-generalization from a single data point. Even more, the [follow-up post](http://anand.typepad.com/datawocky/2008/04/data-versus-alg.html) references SVD as one of the "complex" algorithms not worth trying because it limits the ability of scaling up to larger number of features. Clearly, Anand's students did not win the Netflix Prize, and they probably now realize that SVD did have a major role in the winning entry.

As a matter of fact, many teams showed later that adding content features from IMDB or the like to an optimized algorithm had little to no improvement. Some of the members of the [Gravity team](http://www.gravityrd.com/references/netflix-prize?lang=en), one of the top contenders for the Prize, published a detailed paper in which they showed how those content-based features would add no improvement to the highly optimized collaborative filtering matrix factorization approach. The paper was entitled “[Recommending New Movies: Even a Few Ratings Are More Valuable Than Metadata](http://dl.acm.org/citation.cfm?id=1639731&dl=ACM&coll=DL&CFID=122239967&CFTOKEN=16331362)“.

| ![](/blog/images/AFewRatings.png) |

To be fair, the title of the paper is also an over-generalization. Content-based features (or different features in general) might be able to improve accuracy in many cases. But, you get my point again: **More data does not always help**.

## The End of the Scientific Method?

Of course, whenever there is a heated debate about a possible paradigm change, there are people like Malcolm Gladwell or Chris Anderson that make a living out of heating it even more (don’t get me wrong, I am a fan of both, and have read most of their books). In this case, Anderson picked on some of Norvig’s comments, and misquoted them in an article entitled: “[The End of Theory: The Data Deluge Makes the Scientific Method Obsolete](http://www.wired.com/science/discoveries/magazine/16-07/pb_theory/)“.

| ![](/blog/images/EndOfScientificApproach.png) |

The article explains several examples of how the abundance of data helps people and companies take decision without even having to understand the meaning of the data itself. As Norvig himself points out in [his rebuttal](http://norvig.com/fact-check.html), Anderson has a few points right, but goes above and beyond to try to make them. And the result is a set of false statements, starting from the title: the data deluge does not make the scientific method obsolete. I would argue it is rather the other way around.

## Data Without a Sound Approach = Noise

So, am I trying to make the point that the Big Data revolution is only hype? No way. Having more data, both in terms of more examples or more features, is a blessing. The availability of data enables more and better insights and applications. More data indeed enables better approaches. More than that, it **requires** better approaches.

| ![](/blog/images/NoToAnderson.jpg) |

In summary, we should dismiss simplistic voices that proclaim the uselessness of theory or models, or the triumph of data over these. As much as data is needed, so are good models and theory that explains them. But, overall, what we need is good approaches that help us understand how to interpret data, models, and the limitations of both in order to produce the best possible output.

In other words, data is important. But, data without a sound approach becomes noise.

(**Note**: this post is based on a few slides included in my [Strata talk](http://strataconf.com/strata2012/public/schedule/detail/22364) a few months back. Those slides sparked an interesting debate, and follow up emails that prompted me to write these lines)

```
---
id: 77
title: 'Netflix @ Recsys 2012'
date: '2012-09-05T17:18:00+00:00'
author: 'Xavier Amatriain'
##layout: post
permalink: /netflix-recsys-2012/
header:
  teaser: https://amatriain.net/blog/images/14-01.png
categories:
    - Uncategorized
---
```

We are just a few days away from the [2012 ACM Recommender Systems Conference](http://recsys.acm.org/2012/images/header.png) (#Recsys2012), that this year will take place in Dublin, Ireland. Over the years, Recsys has become my favorite conference because of its unique blend of academic research and industrial applications. If you are not familiar with the conference, you might get a flavor by reading [my report from last year](http://localhost:8080/wordpress/2011/11/recsys-2011-notes-and-pointers.html). Needless to say that it is also dear to my heart because of my involvement as General Cochair in its [2010 edition in Barcelona](http://recsys.acm.org/2010/).

If you had to mention a single company that is identified with recommender systems and technologies, that would probably be Netflix. The Netflix Prize started a year before the first Recsys conference in Minneapolis, and it impacted Recommender Systems researchers and practitioners in many ways. So, it comes as no surprise that the relation between the conference and Netflix also goes a long way. Netflix has been involved in the conference through the years. And, this time in Dublin is not going to be any different. Not only Netflix is a proud sponsor of the conference, but you will have the chance to listen to presentations and meet some of the people that make the wheels of Netflix recommendations turn. Here are some of the highlights of Netflix’ participation:

- Both Harald Steck and myself are involved in organizing the workshop on “**[Recommender Utility Evaluation: Beyond RMSE](http://ir.ii.uam.es/rue2012/)**“. We believe that finding the right evaluation metrics is one of the key issues for recommender systems. This workshop will be a great event to not only discover latest research in the area, but also brainstorm and discuss on the issue of recsys evaluation. Unfortunately, I will miss the workshop because of my traveling schedule. But, Harald will be representing Netflix on the organization side.
- On that same workshop, you should not miss the keynote by our Director of Innovation Carlos Gomez-Uribe. The talk is entitled “[**Challenges and Limitations in the Offline and Online Evaluation of Recommender Systems: A Netflix Case Study**](http://ir.ii.uam.es/rue2012/keynote.html)“. Carlos will give some insights into how we deal with online A/B and offline experimental metrics.
- On Tuesday, I will be giving a 90 minute tutorial on “**[Building industrial-scale real-world Recommender Systems](http://recsys.acm.org/2012/tutorials.html#building)**“. In this tutorial, I will talk about all those things that matter in a recommender system, and are usually outside of the academic focus. I will describe different ways that recommendations can be presented to the users, evaluation through AB testing, data, and software architectures. I look forward to seeing you all there.

Besides Harald, Carlos and myself, you should also look forward to meeting other members of the personalization team at Netflix. Rex Lam, Justin Basilico, and Kelvin Jiang, will also be attending the conference. We are all looking forward to meeting old and new friends and interacting with the Recsys community during the conference. If you want to make sure we meet, feel free to send me an email (first name initial+last name at netflix.com) or contact through [twitter](https://twitter.com/xamat).

```
---
id: 75
title: 'Recsys 2012: A long (and likely biased) summary'
date: '2012-09-17T17:47:00+00:00'
author: 'Xavier Amatriain'
##layout: post
permalink: /recsys-2012-long-and-likely-biased/
header:
  teaser: https://amatriain.net/blog/images/RecsysPhoto.jpg
categories:
    - Uncategorized
---
```

After a great week in beautiful and sunny Dublin (yes, sunny), it is time to look back and recap on the most interesting things that happened in the 2012 Recsys Conference. I have been attending the conference since its first edition in Minnesota. And, it has been great to see the conference mature to become the premiere event for recommendation technologies. I can’t hide that this is my favorite conference for several reasons: perfect size, great community, good involvement from industry, and good side program of tutorials, workshops, and demos.

| ![](/blog/images/RecsysPhoto.jpg) |

This year I arrived a bit late and missed the first day of tutorials, and first day of the conference. But, was able to catch up after jumping right in with my 90 minute tutorial on “Building Industrial-scale Real-world Recommender Systems”

In my tutorial ([see slides here](http://www.slideshare.net/xamat/building-largescale-realworld-recommender-systems-recsys2012-tutorial)), I talked about the importance of four different issues in real-world recommender systems:

- Paying attention to user interaction models that support things like explanations, diversity, or novelty.
- Coming up with algorithms that, beyond rating prediction, focus on other aspects of recommendation such as similarity, or, in particular, ranking.
- Using results of online A/B tests, and coming up with offline model metrics that correlate with the former.
- Understanding the software architectures where your recommender system will be deployed.

I was happy to see that some of these issues not only were mentioned, but almost became conducting threads throughout the conference. Of course, this might be in the eye of the beholder, and others might have come back with the impression that the main topics were others (I recommend you read these two other Recsys 2012 summaries by [Daniel Tunkelang](http://thenoisychannel.com/2012/09/14/recsys-2012-beyond-five-stars/) and [Daniele Quercia](http://www.syslog.cl.cam.ac.uk/2012/09/14/recsys-2012-few-things-i-remember/)). In any case, grouping in topics will help me summarize the many things I found interesting.

### Online A/B Testing and offline metrics

I am glad to see that this has become a relevant topic for the conference, because many of us believe this is one of the most important topics that need to be addressed by both industry and academia. One of these people is Ron Kohavi, who delivered a great keynote on “[Online Controlled Experiments: Introduction, Learnings, and Humbling Statistics](http://www.exp-platform.com/Pages/2012RecSys.aspx)“, where he described his learnings of many years of AB Testing in Amazon and Microsoft. It is funny that I cited his KDD 2012 paper in two slides in my tutorial, not knowing that he was in the audience. I recommend you go through his slides, it was one of the best talks of the conference for sure.

The importance of finding relevant metrics was, as a matter of fact, the focus of a workshop we organized with Harald Steck (Netflix), Pablo Castells (UAM), Arjen de Vries, and Christian Posse (LikedIn). The title of the workshop was “Recommendation Utility Evaluation: Beyond RMSE”. Unfortunately, I was not able to attend. But, I do know the keynote by Carlos Gomez-Uribe, also from Netflix, was very well received. And, the workshop as a whole went very well with several interesting papers and even more interesting discussions. You can access the papers on the [website](http://ir.ii.uam.es/rue2012/).

A couple of papers in the main track of the conference also touched upon the importance of optimizing several objectives at the same time. In “[Multiple Objective Optimization in Recommender Systems](http://www.slideshare.net/mechanistician/recsys-2012-slides?ref=http://thenoisychannel.com/)“, Mario Rodriguez and others explain how they design LinkedIn recommendations by optimizing to several objectives at once (e.g. candidate that is good for the job + who is open to new opportunities). They report results from an AB Test run on LinkedIn. In “[Pareto-Efficient Hybridization for Multi-Objective Recommender Systems](http://www.slideshare.net/marcotulioribeiro54/presentation-recsys12)“, Marco Tulio Ribeiro and others from Universidade Federal de Minas Gerais &amp; Zunnit Technologies take the multi-objective a step further. In their case, they optimize the system to not only be accurate, but also present novel or diverse items.

Some other papers went beyond the academic experimental procedure and implemented real systems that were tested with users. A good example is “Finding a Needle in a Haystack of Reviews: Cold Start Context-Based Hotel Recommender System” by researchers from the Tel Aviv Yaffo College and Technicolor.

### Learning to Rank

Another hot topic in this year’s recsys was ranking (or Top-n Recommendations as some prefer to call it). It is good to see that after some time publicly speaking about the importance of ranking approaches, the community seems now to be much more focused on ranking than on rating prediction. Not only there was a whole session devoted to ranking, but actually many other papers in the conference dealt with the topic in some way or another.

I will start by mentioning the very good work by my former colleagues from Telefonica. Their paper “CLiMF: Learning to Maximize Reciprocal Rank with Collaborative Less-is-More Filtering” won the best-paper award. And, I think most of us thought that it was very well-deserved. It is a very good piece of work. Well motivated, evaluated, and, it addresses a very practical issue. It is great to see the Recsys team at Telefonica that I started be acknowledged with this award. You can access the paper [here](http://baltrunas.info/papers/Shi12-climf.pdf) and the slides [here](http://www.slideshare.net/kerveros99/climf-collaborative-lessismore-filtering).

In that same session, researchers from the Université Paris 6 presented “Ranking with Non-Random Missing Ratings: Influence of Popularity and Positivity on Evaluation Metrics”, an interesting study on the very important issue of negative sampling, and popularity bias in learning to rank. The paper discusses these effects on the AUC (Area Under the Curve) measure, a measure that is not very well-behaved, nor very much used in evaluating ranking algorithms. Still, it is a valuable first step in a very interesting line of work. It is interesting to point out that the CLiMF paper addressed the issue of negative sampling in a radically different way: only considering positive samples. Yet another interesting paper in that session was “[Sparse Linear Methods with Side Information for Top-N Recommendations](http://www-users.cs.umn.edu/%7Exning/papers/Ning2012.pdf)“, a model for multidimensional context-aware learning to rank.

Another ranking paper, “Alternating Least Squares for Personalized Ranking” by Gábor Takács from Széchenyi István University and Domonkos Tikk from Gravity R&amp;D, received an honorable mention. The main author coined an (un)popular sentence during his presentation when he invited anyone not interested in Mathematics to leave the room. An unnecessary invitation in a conference that prides itself for being inclusively multidisciplinary. In Recsys, psychologists are seating through systems presentations as much as mathematicians are seating through user-centric sessions, and that is what makes the conference appealing. In any case, the paper presents an interesting way to combines a ranking-based objective function introduced in last year’s kdd and the use of ALS instead of SGD to come up with another approach to learning to rank.

Two papers dealing with recommendations in Social Networks also focused on ranking. “[On Top-k Recommendation Using Social Networks](http://eeweb.poly.edu/faculty/yongliu/docs/topk_tr.pdf)” by researchers from NYU and Bell Labs, and “[Real-Time Top-N Recommendation in Social Streams](http://www.l3s.de/web/upload/documents/1/diaz_recsys2012.pdf)” by Ernesto Diaz-Aviles and other researchers from the University of Hannover. The same first author had an interesting short paper in the poster session: “[Swarming to Rank for Recommender System](http://www.l3s.de/web/upload/documents/1/sp197-diaz.pdf)“. In that poster he proposes the use of a Particle Swarm Optimization algorithm to directly optimize ranking metrics such as MAP. The method proposes an interesting alternative to the use of Genetic Algorithms or Simulated Annealing for this purpose.

Finally, the industry keynote by Ralf Herbrich from Facebook, also introduced the world of Bayesian Factor Models for large-scale distributed ranking. This method, introduced by the same author and others from MSR as “[Matchbox](http://www2009.eprints.org/12/1/p111.pdf)” is now used in different settings. For example, the poster “[The Xbox Recommendation System](http://www.eng.tau.ac.il/%7Enoamk/papers/KNPS12.pdf)” presented its applicability for recommending movies and games for the Xbox. And, in “[Collaborative Learning of Preference Rankings](http://research.microsoft.com/pubs/166623/RecSys-2012.pd)” the authors apply it to… sushi recommendation!

### User-centric, interfaces &amp; explanations

This was probably the third big area of focus of the conference, with many contributions in papers, tutorials, and workshops. The first day, there were actually two tutorials that would fall into this category. In “Personality-based Recommender Systems: An Overview”, the authors presented the idea of using personality traits for modeling user profiles. Among other things, they introduced their proposal to use PersonalityML, an XML-based language for personality description. Interestingly, in the industry session, we saw that this is actually a quite practical thing to do. Thore Graepel from Microsoft explained their experiments in using The Big Five personality traits for personalization. In the other tutorial, “[Conducting User Experiments in Recommender Systems](http://www.slideshare.net/usabart/tutorial-on-conducting-user-experiments-in-recommender-systems)“, Bart Knijnenburg gave a thorough overview of how to conduct user studies for recommender systems. He also introduced his model for using structural equations to model the effects to evaluate. Again, I missed this tutorial, but I was fortunate to hear a very similar presentation by him in Netflix.

In “[Inspectability and Control in Social Recommenders](http://bostandjiev.com/Content/publications/SocialRecStudy.pdf)“, Bart himself (and researchers from UCSB) analyze the effect of giving more information and control to users in the context of social recommendations. A similar idea is explored in the short paper “The Influence of Knowledgeable Explanations on Users’ Perception of a Recommender System” by Markus Zanker.

Two papers addressed the issue of how much information we should require from users. In “User Effort vs. Accuracy in Rating-Bbased Elicitation” Paolo Cremonesi and others analyze how many ratings “are enough” for producing satisfying recommendations in a cold-start setting. And, in “How Many Bits Per Rating?”, the Movielens crew try to quantify the amount of information and noise in user ratings from an information-theoretical perspecive. An interesting continuation to [my work on user ratings noise](http://localhost:8080/wordpress/2009/04/i-like-it-i-like-it-not-or-how-miss.html). However, as the first author himself admited, this is just initial work.

Other highlights of user-centric work that fell more on the UI side were the paper “TasteWeights: A Visual Interactive Hybrid Recommender System” by my friends at UCSB, as well as the many papers presented in the [Workshop on Interfaces for Recommender System](https://homepages.abdn.ac.uk/csc284/pages/InterfaceRS/).

### Data &amp; Machine Learning Challenges

If somebody thought that data and machine learning challenges would fade away after the Netflix Prize, this year’s Recys was a clear example that this is far from being the case. Many challenges have taken over after that: the yearly KDD Cups, Kagel, Overstock, last year’s MoviePilot challenge, the Mendeley Challenge… Recsys had this year a [Tutorial/Panel](http://recsys.acm.org/2012/tutorials.html#best) and a [Workshop](http://2012.recsyschallenge.com/) on Recommender Systems Challenges, both organized by Alan Said, Domonkos Tikk, and others. I could not attend the Tutorial since it was happening at the same time than mine. But, I was able to catch some interesting presentations in the Workshop. Domonkos Tikk from Gravity R&amp;D gave [a very interesting presentation](http://www.slideshare.net/domonkostikk/from-a-toolkit-of-recommendation-algorithms-into-a-real-business-the-gravity-rd-experience) on how they evolved from being a team in the Netflix Prize to a real-world company with very interesting projects. Kris Jack from Mendeley also gave two interesting talks on the Mendeley recommender systems. In [one of them](http://www.slideshare.net/KrisJack/mendeley-suggest-engineering-a-personalised-article-recommender-system), he explained how they make use of AWS and Mahout in a system that can generate personalized recommendations for about $60 a month. In [the other](http://www.slideshare.net/KrisJack/rec-sys12mendeleydatachallenges), he talked about their perspective on data challenges.

### Context-aware and location-based recommendations

This has become a traditional area of interest in Recsys. It has now matured to a point that it has it own session, and two workshops: “[Personalizing the Local Mobile Experience](http://loca.mobilelifecentre.org/)“, and the “[Workshop on Context-Aware Recommender Systems](http://cars-workshop.org/)“. But, besides having its own session in the conference, several other papers in others also deal with context-aware recommendations. I have already mentioned “Sparse Linear Methods with Side Information for Top-N Recommendations”, for example. Other interesting papers in this area were “Context-Aware Music Recommendation Based on Latent Topic Sequential Patterns”, on the issue of playlist generation, and “[Ads and the City: Considering Geographic Distance Goes a Long Way](http://www.cl.cam.ac.uk/%7Edq209/publications/trumper12ads.pdf)” for location-aware recommendations.

### Social

A similar area that has already matured over several Recsys is Social. It has its own session, and Workshop, “[Workshop on Recommender Systems and the Social Web](http://ls13-www.cs.uni-dortmund.de/homepage/rsweb2012/index.shtml)” , and trancends over many other papers. In this area, the paper that I have not mentioned in other categories and found interesting was “[Spotting Trends: The Wisdom of the Few](http://www.slideshare.net/daniele.quercia/recsys-14297597)“. One of the reasons I found the paper interesting is because it builds on our idea of using a reduced set of experts for recommendations, what we called “[The Wisdom of the Few](http://localhost:8080/wordpress/2009/05/wisdom-of-few.html)“.

### Others

And yes, I still have some interesting stuff from the poster session that I could not fit into any of the above categories.

First, the short paper “[Using Graph Partitioning Techniques for Neighbour Selection in User-Based Collaborative Filtering](http://ir.ii.uam.es/%7Ealejandro/2012/recsys.pdf)” by Alejandro Bellogin. Alejandro won the Best Short Paper Award, for a great piece of work and presentation. He described an approach to use the Normalized Cut graph clustering approach for grouping similar users, and improve neighborhood formation in standard kNN Collaborative Filtering.

I also liked the poster “[Local Learning of Item Dissimilarity Using Content and Link Structure](http://cse.iitkgp.ac.in/%7Epabitra/paper/recsys12.pdf)“, another graph-based approach, in this case to learn a similarity function.

Finally, in “When Recommenders Fail: Predicting Recommender Failure for Algorithm Selection and Combination”, Michael Ekstrand starts to tap into an extremely important question: when and why do some recommendation algorithms fail? This question has been informally discussed in the context of hybrid recommenders and ensembles. But, there is clearly much more work to do, and many things to understand.

____

Well, if you made it all the way to here, it means that you are really interested in Recommender Systems. So, chances are that I will be seeing you in next year’s Recsys. Hope to see you in Hong Kong!

```
---
id: 73
title: '10 Little lessons for life that I learned from running'
date: '2013-01-16T06:50:00+00:00'
author: 'Xavier Amatriain'
##layout: post
permalink: /10-little-lessons-for-life-that-i/
header:
  teaser: https://amatriain.net/blog/images/ChicagoFinisher.jpg
categories:
    - Uncategorized
---
```

(Sorry for allowing myself to depart from the usual geeky computer science algorithmic talk in this blog. I owed it to myself and my biggest hobby to write a post like this. I hope you bear with me.)

Around 3 years ago, I smoked, I was overweight, and only exercised occasionally. Being a fan of radical turns in my life, I decided one day to go on a week-long liquid diet, I stopped smoking, and I took up running, with the only goal in my mind to some time run the half marathon in my home town. Little did I know that the decision to run would change my life in so many ways. This last year 2012, I have run 3 marathons, 4 half marathons, and a 199 mile relay with a team of 12. But, beyond that, I am convinced that I owe part of my personal and professional success these past years to the fact that I am a runner.

This post is my little homage to running and to the many lessons I have found in my journey.

| ![](/blog/images/ChicagoFinisher.jpg) |

When I started running I had lots of problems. The main one was due to an old knee injury that hit back on me. I had an ACL surgery when I was 16, and ever since my right knee has not been the same. When my knee started hurting this time, I visited several doctors, some specialized in sports. All of them recommended I should give up running. Some told me straight out that I would never be able to run a marathon. It took me lots of visits to the chiropractor, and lots of quads exercises over months to get back to running. But, I overcame these initial hurdles, and went into running not one but several marathons.

***Lesson 1. Beginnings are hard:** Starting anything new in life will be hard. You will need to invest lots of energies, and at times you will want to give up. The more important and significant the change is, the more it will take from you.*

| ![](/blog/images/97632-082-020f.jpg) |

After I finished my first marathon in Santa Cruz, and when I thought all my knee problems were long gone, my knee started hurting again. This was nothing like what I had experienced when starting. Still, it could have been enough to stop me from trying again. But, it didn’t. I focused on recovering. Soon I was back on the road.

 Lesson 2. There will be ups and downs: *Once you have overcome the initial difficulties in starting something
 new, you will be tempted to think that everything else should be easy . But, life, like most running courses, will have hills with ups
 and downs.*

| ![](/blog/images/Los3Mosqueteros.jpg) |

It is hard to wake up at 6 am for the morning run. It is easy to stay in bed when your legs are still sore from yesterday’s training. It is tough to go out running when it is raining or freezing outside. It is even harder to decide not to stop when you hit the wall on mile 20 of a marathon. All these day to day small decisions end up adding up and making the difference between you improving and accomplishing your running goals.

***Lesson 3. The importance of those small decisions:** The small day to day decisions play a huge role in building your character. They will end up determining your long term success and the direction your life takes.*

| ![](http://localhost:8080/wordpress/wp-content/uploads/2022/06/Sacramento.jpg) |

When you are not at your best, it is even harder to face all these small decisions I mentioned. If you are down for some time because of an injury, it is tough to start again on your own. Having a group of friends that share your passion for running is extremely important. I am fortunate to have a large group of friends that push me to become better, and help me get up when I fall.

***Lesson 4. You are not alone – the power of social influence… and friends:** Whatever new adventure you start in life, it is important to have people around you that understand and support it. People that share your passion can make a difference when you need it.*

| ![](/blog/images/723255-1001-0002s.jpg) |

As much as I have appreciated having that extra support from friends and other fellow runners, there are many times I have felt the pressure of having to make a decision on my own. Many of those small decisions such as getting up off bed on a rainy day, for example. Nobody is going to make them for you. I have also felt alone in many of my training runs. And, of course, in mile 20 of a marathon, when everyone is giving their best but you can only see strangers around you. In all those moments it is important to be strong and be ready to carry on, on your own.

***Lesson 5. But, you will be alone:** No matter how many friends support you, you will have to face important decisions on your own, and carry your own weight.*

| ![](/blog/images/Relay.jpg) |

It is well known that “repetition leads to mastery”. This is even more so for activities that require developing physical strength and resistance. There is no other secret to becoming a better runner than to run, and run often. Putting on more miles is the goal. Everything else will come.

***Lesson 6. Repeat, repeat, repeat, repeat:** Repetition is the key to mastering most things in life. If you want to become good at doing something, ask yourself how you can invest thousands of hours in it (read about [the 10k hour rule](http://www.gladwell.com/outliers/outliers_excerpt1.html) in Malcolm Gladwell’s Outliers).*

| ![](/blog/images/WithAitor.jpg) |

As much as repetition is needed to improve, it is hard to do so without a goal in mind. During my time running I have learned the power of having concrete goals. Setting up goals that are achievable in the long run, but not too easy to get to. As I have progressed, I have learned to be more demanding. My current goals is to do a 3:30 marathon, and a 1:30 half. The first one is achievable, the second one will need much more work. But these goals will keep me going and focused for some time.

***Lesson 6. Set your goals:** Setting ambitious but achievable goals in life will help you push harder and will keep you focused and looking forward.*

| ![](/blog/images/IMAG0506.jpg) |

When I look back at the way I started running, I realize how many things I did wrong. I have learned so much since then. I have read books, watched movies and online videos, talked to people that know much more than I do. I have also learned from looking at the data that I generate from each of my trainings. I have also learned to listen to and understand my body I am fortunate enough that I love learning, and I have enjoyed every bit of this learning experience.

Lesson 7. Data and knowledge: *Use all the information around you to improve your life. Data about you can give you insights into how to become better. And any knowledge you gain from external sources can make a difference when taking a decision.*

| ![](/blog/images/ClothesReady.jpg) |

One of the reasons why beginnings are hard (Lesson 1) is that people that start running tend to overdue it by, for example, increasing distance and pace at the same time. This typically leads to injury, and frustration. One of the most important things to learn when starting to run is to understand your own limitations. Even when you do, you will be tempted to push to hard by continuing to run when your leg hurts, or by doing one too many races in a short period. I have done all of the above. But it is important to remember that everyone has their limits and forcing beyond them can result in long term problems.

Lesson 8. Everyone has their limits: *Pushing yourself hard is good. However, there is such a thing as pushing *too* hard. You need to understand where your limits are to push them further, but only little by little.*

| ![](/blog/images/HMB.jpg) |

No matter how hard it can get at some points, no matter how long it can take you, there is no doubt that you can do whatever you set your mind to. I don’t have any special conditions for running, and I have never had. I don’t think I will ever be a “great” runner. However, now I look back and laugh when I remember my unreachable goal a little over 3 years ago was “only” to run a half marathon. If someone like me, with little or no pre-existing conditions, family and work obligations, and very little time, can do it, so can you.

***Lesson 9. But, yes you can:** No matter how low you fall or how far your goal is, you can do it. Only think about the many people just like you who have done it before (e.g. Estimates are that around 0.1 to 0.5% of US Population has completed a Marathon). Why should you be any less?*

| ![](/blog/images/VasonaLakePedroAlbertoXavi.jpg) |

As a conclusion, let me stress that the fact anyone can run does not mean that running is easy, and it requires no effort. It is precisely the fact that it is hard and requires effort for a long period of time what makes it worthwhile. Like most good things in life.

***Lesson 10. All good things come hard:** Think about it, all worthy things in life require effort and dedication. Being healthy, fit, happy, having a career, or a family, they all require your energy and long time investment. Just go with it, and enjoy every bit of the journey.*
```
---
id: 71
title: 'Tools for Data Processing @ Webscale'
date: '2013-06-09T18:19:00+00:00'
author: 'Xavier Amatriain'
##layout: post
permalink: /tools-for-data-processing-webscale/
header:
  teaser: https://amatriain.net/blog/images/ParquetTwitter-Poster.jpg

categories:
    - Uncategorized
---
```

A couple of days ago, I attended the [Analytics @Webscale workshop](http://analyticswebscale.splashthat.com/) at Facebook. I found this workshop to be very interesting from a technical perspective. This conference was mostly organized by Facebook Engineering, but they invited LinkedIn, and Twitter to present, and the result was pretty balanced. I think the presentations, though biased to what the 3 “Social Giants” do, were a good summary of many of the problems webscale companies face when dealing with Big Data. It is interesting to see how similar problems can be solved in different ways. I recently described how we address at Netflix many of these issues in our [Netflix Techblog](http://techblog.netflix.com/2013/03/system-architectures-for.html). It is also interesting to see how much sharing and interaction there is nowadays in the infrastructure space, with companies releasing most of what they do as open source, and using – and even building upon – what their main business competitors have created. These are my barely edited notes:

Twitter presented several components in their infrastructure. They use Thrift on HDFS to store their logs. They have now build [Twitter Parquet](http://parquet.io/), a columnar storage database that improves storage efficiency by allowing to read columns at a time.

![](/blog/images/ParquetTwitter-Poster.jpg)
@squarecog talking about Parquet

They also presented their DAL (Data Access Layer), built on top of [HCatalog](http://incubator.apache.org/hcatalog/).

| ![](/blog/images/TwitterDAL-HCatalog-Slide.jpg)|

| ![](/blog/images/DALTwitter-Poster.jpg)|

Of course, they also talked about [Twitter Storm](http://storm-project.net/), which is their approach to distributed/nearline computation. Every time I hear about Storm it sounds better. Storm now supports different parts of their production algorithms. For example, the ranking and scoring of tweets for real-time search is based on a Storm topology.

|![](/blog/images/TwitterRankingStorm-Slide.jpg) |

Finally, they also presented a new tool called [Summingbird](http://lanyrd.com/2013/lambda-jam/scghxb/). This is still not open sourced, but they are planning on doing so soon. Summingbird is a DSL on top of [Scalding](https://github.com/twitter/scalding) that allows to define workflows that integrate offline batch processing from Hadoop and near-line from Storm.

| ![](/blog/images/TwitterSummingbird-Slide.jpg) |

| ![](/blog/images/Summingbird-Poster.jpg) |

LinkedIn also talked about their approach to combining offline/near-line/real-time computation although I always get the sense that they are much more leaning towards the former. They talked about three main tools: [Kafka](http://kafka.apache.org/), their publish subscribe system; [Azkaban](http://data.linkedin.com/opensource/azkaban), a batch job scheduler we have talked about using in the past; and [Espresso](http://data.linkedin.com/projects/espresso) a timeline-consistent NOSQL database.

| ![](/blog/images/Exa-scaleSystemsFacebook-Slide.jpg) |

Facebook also presented their whole stack. Some known tools, some not so much. [Facebook Scuba](https://www.facebook.com/notes/facebook-engineering/under-the-hood-data-diving-with-scuba/10150599692628920) is a distributed in-memory stats store that allows them to read distributed logs and query them fast. [Facebook Presto](http://gigaom.com/2013/06/06/facebook-unveils-presto-engine-for-querying-250-pb-data-warehouse/) was a new tool presented as the solution to get fast queries out of Exabyte-scale data stores. The sentence “A good day for me is when I can run 6 Hive queries” supposedly attributed to a FB data scientist stuck in my mind ;-).

Morse is a different distributed approach to fast in-memory data loading. And, [Puma/ptail](http://www.quora.com/Why-did-Facebook-develop-Puma-pTail-instead-of-using-existing-ones-like-Flume) is a different approach to “tailing” logs, in this case into HBase.

| ![](/blog/images/FacebookScubaPoster.jpg) |

Another Facebook tool that was mentioned by all three companies is [Giraph](https://github.com/apache/giraph). (To be fair, Giraph was started at Yahoo, but Facebook hired the creator Avery Ching). Giraph is a graph-based distributed computation framework that works on top of Hadoop. Facebook claims they ran a Page Rank on a graph with a trillion edges on 200 machines in less than 6 minutes/iteration. Giraph is another alternative to [Graphlab](http://graphlab.org/). Both LinkedIn and Twitter are using it. In the case of Twitter, it is interesting to hear that they now prefer it to their own in-house (although single-node) [Cassovary](https://github.com/twitter/cassovary). It will be interesting to see all these graph processing tolls side by side in this year’s [Graphlab workshop](http://graphlab.org/graphlab-workshop-2013/).

Another interesting thread I heard from different speakers as well as coffee-break discussions was the use of [Mesos vs. Yarn](http://www.quora.com/How-does-YARN-compare-to-Mesos) or even [Spark](http://spark-project.org/). It is clear that many of us are looking forward to the NextGen Mapreduce tools to reach some level of maturity.

```
---
id: 69
title: 'Recommendations as Personalized Learning to Rank'
date: '2013-07-23T18:50:00+00:00'
author: 'Xavier Amatriain'
##layout: post
permalink: /recommendations-as-personalized/
image: /blog/imagesTwoDimensionalRanking.png
header:
  teaser: https://amatriain.net/blog/images/TwoDimensionalRanking.png

categories:
    - Uncategorized
---
```

As I have explained in other publications such as the[Netflix Techblog](http://techblog.netflix.com/2012/04/netflix-recommendations-beyond-5-stars.html), ranking is a very important part of a Recommender System. Although the Netflix Prize focused on rating prediction, ranking is in most cases a much better formulation for the recommendation problem. In this post I give some more motivation, and an introduction to the problem of personalized learning to rank, with pointers to some solutions. The post is motivated, among others, by a proposal I sent for a tutorial at this year’s [Recsys](http://recsys.acm.org/recsys13/). Coincidentally, my former colleagues in Telefonica, who have been working in learning to rank for some time, proposed a very similar one. I encourage you to use this post as an introduction to [their tutorial](http://recsys.acm.org/recsys13/tutorials/#content-tab-1-1-tab), which you should definitely attend.

The goal of a ranking system is to find the best possible ordering of a set of items for a user, within a specific context, in real-time. We optimize ranking algorithms to give the highest scores to titles that a member is most likely to play and enjoy.

If you are looking for a ranking function that optimizes consumption, an obvious baseline is item popularity. The reason is clear: on average, a user is most likely to like what most others like. Think of the following situation: You walk into a room full of people you know nothing about, and you are asked to prepare a list of ten books each person likes. You will get $10 for each book you guess right. Of course, your best bet in this case would be to prepare identical lists with the “10 most liked books in recent times”. Chances are the people in the room is a fair sample of the overall population, and you end up making some money. However, popularity is the opposite of personalization. As I explained in the previous example, it will produce the same ordering of items for every member. The goal becomes is to find a personalized ranking function that is better than item popularity, so we can better satisfy users with varying tastes. Our goal is to recommend the items that each user is most likely to enjoy. One way to approach this is to ask users to rate a few titles they have read in the past in order to build a rating prediction component. Then, we can use the user's predicted rating of each item as an adjunct to item popularity. Using predicted ratings on their own as a ranking function can lead to items that are too niche or unfamiliar, and can exclude items that the user would want to watch even though they may not rate them highly. To compensate for this, rather than using either popularity or predicted rating on their own, we would like to produce rankings that balance both of these aspects. At this point, we are ready to build a ranking prediction model using these two features.

Let us start with a very simple scoring approach by choosing our ranking function to be a linear combination of popularity and predicted rating. This gives an equation of the form *score(u,v) = w1 p(v) + w2 r(u,v) + b*, where *u*=user, *v*=video item, *p*=popularity and *r*=predicted rating. This equation defines a two-dimensional space as the one depicted in the following figure.

| ![](/blog/images/TwoDimensionalRanking.png) |

Once we have such a function, we can pass a set of videos through our function and sort them in descending order according to the score. First, though, we need to determine the weights ***w1*** and ***w2*** in our model (the bias ***b*** is constant and thus ends up not affecting the final ordering). We can formulate this as a machine learning problem: select positive and negative examples from your historical data and let a machine learning algorithm learn the weights that optimize our goal. This family of machine learning problems is known as "Learning to Rank" and is central to application scenarios such as search engines or ad targeting. A crucial difference in the case of ranked recommendations is the importance of personalization: we do not expect a global notion of relevance, but rather look for ways of optimizing a personalized model.

As you might guess, the previous two-dimensional model is a very basic baseline. Apart from popularity and rating prediction, you can think on adding all kinds of features related to the user, the item, or the user-item pair.Below you can see a graph showing the improvement we have seen at Netflix after adding many different features and optimizing the models.

| ![](/blog/images/Ranking-FeaturesPerformance.png) |

The traditional pointwise approach to learning to rank described above treats ranking as a simple binary classification problem where the only input are positive and negative examples. Typical models used in this context include Logistic Regression, Support Vector Machines, Random Forests or Gradient Boosted Decision Trees.

There is a growing research effort in finding better approaches to ranking. The pairwise approach to ranking, for instance, optimizes a loss function defined on pairwise preferences from the user. The goal is to minimize the number of inversions in the resulting ranking. Once we have reformulated the problem this way, we can transform it back into the previous binary classification problem. Examples of such an approach are RankSVM \[Chapelle and Keerthi, 2010, [Efficient algorithms for ranking with SVMs](http://olivier.chapelle.cc/pub/ordinal.pdf)\], RankBoost \[Freund et al., 2003, [An efficient boosting algorithm for combining preferences](http://machinelearning.wustl.edu/mlpapers/paper_files/FreundISS03.pdf)\], or RankNet \[Burges et al., 2005, [Learning to rank using gradient descent](http://research.microsoft.com/en-us/um/people/cburges/papers/icml_ranking.pdf)\].

We can also try to directly optimize the ranking of the whole list by using a listwise approach. RankCosine \[Xia et al., 2008. [Listwise approach to learning to rank: theory and algorithm](http://research.microsoft.com/en-us/people/tyliu/icml-listmle.pdf)\], for example, uses similarity between the ranking list and the ground truth as a loss function. ListNet \[Cao et al., 2007. [Learning to rank: From pairwise approach to listwise approach](ftp://ftp.research.microsoft.com/pub/tr/TR-2007-40.pdf)\] uses KL-divergence as loss function by defining a probability distribution. RankALS \[Takacs and Tikk. 2012. [Alternating least squares for personalized ranking](http://wanlab.poly.edu/recsys12/recsys/p83.pdf)\] is a recent approach that defines an objective function that directly includes the ranking optimization and then uses Alternating Least Squares (ALS) for optimizing.

Whatever ranking approach we use, we need to use rank-specific information retrieval metrics to measure the performance of the model. Some of those metrics include [Mean Average Precision](http://en.wikipedia.org/wiki/Information_retrieval#Mean_average_precision)(MAP), [Normalized Discounted Cumulative Gain](https://en.wikipedia.org/wiki/Discounted_cumulative_gain) (NDCG), [Mean Reciprocal Rank](http://en.wikipedia.org/wiki/Mean_reciprocal_rank) (MRR), or Fraction of Concordant Pairs (FCP). What we would ideally like to do is to directly optimize those same metrics. However, it is hard to optimize machine-learned models directly on these measures since they are not differentiable and standard methods such as gradient descent or ALS cannot be directly applied. In order to optimize those metrics, some methods find a smoothed version of the objective function to run Gradient Descent. CLiMF optimizes MRR \[Shi et al. 2012. [CLiMF: learning to maximize reciprocal rank with collaborative less-is-more filtering](http://www.ci.tuwien.ac.at/~alexis/Publications_files/climf-recsys12.pdf)\], and TFMAP \[Shi et al. 2012. [TFMAP: optimizing MAP for top-n context-aware recommendation](http://www.ci.tuwien.ac.at/~alexis/Publications_files/tfmap-sigir12.pdf)\], optimizes MAP in a similar way. The same authors have very recently added a third variation in which they use a similar approach to optimize "graded relevance" domains such as ratings \[Shi et. al, "[Gapfm: Optimal Top-N Recommendations for Graded

Relevance Domains](http://arxiv.org/pdf/1307.3855.pdf)“\]. AdaRank \[Xu and Li. 2007. [AdaRank: a boosting algorithm for information retrieval](http://research.microsoft.com/en-us/people/hangli/xu-sigir07.pdf)\] uses boosting to optimize NDCG. Another method to optimize NDCG is NDCG-Boost \[Valizadegan et al. 2000. [Learning to Rank by Optimizing NDCG Measure](http://books.nips.cc/nips22/spotlight_show/Monday1.pdf)\], which optimizes expectation of NDCG over all possible permutations. SVM-MAP \[Xu et al. 2008. [Directly optimizing evaluation measures in learning to rank](http://research.microsoft.com/en-us/people/tyliu/sigir08-directoptimizeevalmeasure.pdf)\] relaxes the MAP metric by adding it to the SVM constraints. It is even possible to directly optimize the non-diferentiable IR metrics by using techniques such as Genetic Programming, Simulated Annealing \[Karimzadehgan et al. 2011. [A stochastic learning-to-rank algorithm and its application to contextual advertising](http://labs.yahoo.com/files/www2011.pdf%E2%80%8E)\], or even Particle Swarming \[Diaz-Aviles et al. 2012. [Swarming to rank for recommender systems](http://dl.acm.org/citation.cfm?id=2366001)\].

As I mentioned at the beginning of the post, the traditional formulation for the recommender problem was that of a rating prediction. However, learning to rank offers a much better formal framework in most contexts. There is a lot of interesting research happening in this area, but it is definitely worth for more researchers to focus their efforts on what is a very real and practical problem where one can have a great impact.

```
---
id: 68
title: 'Blog posts and Summer gigs'
date: '2014-06-20T21:03:00+00:00'
author: 'Xavier Amatriain'
##layout: post
permalink: /blog-posts-and-summer-gigs/
header:
  teaser: https://amatriain.net/blog/images/14-01.png

categories:
    - Uncategorized
---
```

I have recently heard complaints that this blog is [rather quiet lately](http://www.stackdriver.com/top-devops-influencers-blogs-follow/). I agree. I have definitely been focused on publishing through other sources and have found little time to write interesting things here. On the one hand, I find twitter ideal for communicating quick and short ideas, thoughts, or pointers. You should definitely [follow me](http://www.twitter.com/xamat) there if you want to keep up to date. On the other hand, I have published a couple of posts on the [Netflix Techblog](http://techblog.netflix.com/). A few months ago we published a post describing [our three-tier system architecture](http://techblog.netflix.com/2013/03/system-architectures-for.html) for personalization and recommendations. More recently we described our implementation of [distributed Neural Networks](http://techblog.netflix.com/2014/02/distributed-neural-networks-with-gpus.html) using GPUs and the AWS cloud.

The other thing I continue on doing very often is give talks of our work at different events and venues. In the last few months, for instance, I have given talks at LinkedIn, Facebook, and [Stanford](http://i.stanford.edu/infoseminar/).

This week I gave a talk and attended the Workshop on Algorithms for Modern Massive Datasets ([MMDS](http://mmds-data.org/)). This is a very interesting workshop organized by [Michael Mahoney](http://www.stat.berkeley.edu/~mmahoney/) every two years. It brings together a diverse crowd of people, from theoretical physicist and statisticians to industry practicioners. All of them are united by their work on large scale data-driven algorithms. You can find the slides of my presentation [here](http://www.slideshare.net/xamat/mmds-2014-talk-distributing-ml-algorithms-from-gpus-to-the-cloud).

So, what is next? If you want to catch some of my future talks, I will be giving a couple of public ones in the next few months.

First, I will be lecturing in the Machine Learning Summer School ([MLSS](http://www.mlss2014.com/)) at CMU in early July. I am really looking forward to joining such a great least of speakers and visiting Pittsburgh for the first time. I will be lecturing on Recommendation Systems and Machine Learning Algorithms for Collaborative Filtering.

Late August I will be giving a 3 hour long [Tutorial](http://www.kdd.org/kdd2014/tutorials.html) at KDD in New York. The tutorial is entitled “The Recommender Problem Revisited” and I will be sharing stage with [Bamshad Mobasher](http://www.cdm.depaul.edu/people/pages/facultyinfo.aspx?fid=653).

Finally, I was recently notified that a shorter version of the same tutorial has been accepted at [Recsys](http://recsys.acm.org/recsys14/), which this year is held in the Silicon Valley.

I look forward to meeting many of you in any of these events. Don’t hesitate to ping me if you will be attending.

```
---
id: 67
title: 'Introduction to Recommender Systems: A 4-hour lecture'
date: '2014-08-05T05:19:00+00:00'
author: 'Xavier Amatriain'
```

```
##layout: post
permalink: /introduction-to-recommender-systems-4/
header:
  teaser: https://amatriain.net/blog/images/14-01.png
categories:
    - Uncategorized
tags:
    - 'machine learning'
    - 'recommender systems'
    - recsys
---

A couple of weeks ago, I gave a 4 hour lecture on Recommender Systems at the [2014 Machine Learning Summer
School at CMU](http://mlss2014.com/index.html). The school was organized by Alex Smola and Zico Kolter and,
judging by the attendance and the quality of the speakers, it was a big success. This is the outline of my
lecture:

1. Introduction: What is a Recommender System
2. “Traditional” Methods
* Collaborative Filtering
* Content-based Recommendations
3. “Novel” Methods
* Learning to Rank
* Context-aware Recommendations
 * Tensor Factorization
 * Factorization Machines
4. Deep Learning
5. Similarity<
6. Social Recommendations
7. Hybrid Approaches
8. A practical example: Netflix
9. Conclusions
10. References

You can access the slides in Slideshare and the videos in Youtube, but I thought it would make sense to
gather both here and link them together.

Here are the slides:

<iframe allowfullscreen="" frameborder="0" height="356" loading="lazy" marginheight="0" marginwidth="0"
scrolling="no" src="//www.slideshare.net/slideshow/embed_code/37206312?rel=0" style="border-width: 1px;
border: 1px solid #CCC; margin-bottom: 5px; max-width: 100%;" width="427"> </iframe>

Here is the first session (2 hours):

[![First part](https://img.youtube.com/vi/bLhq63ygoU8/0.jpg)](https://www.youtube.com/watch?v=bLhq63ygoU8)

Here is the second session (2 hours):

[![Second part](https://img.youtube.com/vi/mRToFXlNBpQ/0.jpg)](https://www.youtube.com/watch?v=mRToFXlNBpQ)
---
id: 65
title: 'Ten Lessons Learned from Building (real-life impactful) Machine Learning Systems'
date: '2014-12-19T06:50:00+00:00'
author: 'Xavier Amatriain'
##layout: post
permalink: /ten-lessons-learned-from-building-real/
image: /blog/images/Screenshot2Bfrom2B2014-12-112B223A263A27.png
header:
  teaser: https://amatriain.net/blog/images/Screenshot2Bfrom2B2014-12-112B223A263A27.png
categories:
    - Uncategorized
tags:
    - 'big data'
    - 'machine learning'
    - models
    - 'recommender systems'
    - software
---

(This is a blogpost version of a talk I gave at [MLConf SF](http://mlconf.com/mlconf-sf/) 11/14/2014. See
below for original video and slides)

There are many good textbooks and courses where you can be introduced to machine learning and maybe even
learn some of the most intricate details about a particular approach or algorithm (See [my answer on Quora ]
(http://www.quora.com/What-are-the-best-talks-lectures-related-to-big-data-algorithms-machine-
```

learning/answer/Xavier-Amatriain)on what are good resources for this). While understanding that theory is a very important base and starting point, there are many other practical issues related to building real-life ML systems that you don’t usually hear about. In this post I will share some of the most important lessons learned in years of building large-scale ML solutions that power products such as Netflix and scale to millions of users across many countries.

And just in case it doesn’t come across clearly enough, let me insist on this once again: it does pay off to be knowledgeable and have deep understanding of the techniques and theory behind classic and modern machine learning approaches. Understanding how Logistic Regression works or the difference between Factorization Machines and Tensor Factorization, for example, is a necessary starting point. However, this in itself might not be enough unless you couple it with the real-life experience of how these models interact with systems, data, and users in order to obtain a really valuable impact. The next ten lessons are my attempt at trying to capture some of that practical knowledge.

## 1. More Data <strike>vs.</strike> and Better Models

A lot has been written about whether the key to better results lays in improving your algorithms or simply on throwing more data at your problem (see [my post](/_post/2012-07-03-more-data-or-better-models/) from 2012 discussing this same topic, for example).

In the context of the [Netflix Prize](http://www.netflixprize.com/), [Anand Rajaraman](http://anand.typepad.com/datawocky/anand-rajaraman.html) took an early stand on the issue by claiming that “more data usually beats better algorithms”. In [his post](http://anand.typepad.com/datawocky/2008/03/more-data-usual.html) he explained how some of his students had improved some of the existing results on the Netflix ratings dataset by adding metadata from [IMDB](http://www.imdb.com/).

![](/blog/images/Datawocky.png)
Fig 1. More data usually beats better algorithms

Although many teams in the competition tried to follow that lead and add extra features to improve results, there was little progress in that direction. As a matter of fact, just a year later some of the leaders of what would become the runner up team published [a paper](http://dl.acm.org/citation.cfm?id=1639731) in which they showed that adding metadata had very little impact in improving the prediction accuracy of a well-tuned algorithm. Take this as a first example of why adding more data is not always the solution.
![](/blog/images/AFewRatings.png)
Fig 2. Even a Few Ratings Are More Valuable than Metadata

Of course, there are different ways to “add more data”. In the example above we were adding data by increasing the number and types of features, therefore increasing the dimensionality of our problem space. We can think about adding data in a completely different way by fixing the space dimensionality and simply throwing more training examples at it. Banko and Brill [showed](http://dl.acm.org/citation.cfm?id=1073017)in 2001 that in some cases very different algorithms responded equally well by improving to more training data (see figure below)

![](/blog/images/BankoAndBrill.png)
Fig 3. Banko and Brill’s “famous” model performance curves

Google’s Research Director and renowned AI figure Peter Norvig is quoted as saying that “Google does not have better algorithms, just more data”. In fact, Norvig is one of the co-authors of “[The Unreasonable Effectiveness of Data](http://static.googleusercontent.com/media/research.google.com/en/us/pubs/archive/35179.pdf) where in a similar problem to the one in Banko and Brill (language understanding) they also show how important it is to have “more data”.

![](/blog/images/TheUnreasonableEffectivenessOfData.png)
Fig 4. The Unreasonable Effectiveness of Data

So, is it true that more data in the form of more training examples will always help? Well, not really. The problems above are complex models with a huge number of features which lead to situations of “[high variance](http://en.wikipedia.org/wiki/Bias%E2%80%93variance_tradeoff)“. But, in many other cases this might not be true. See below for example a real-case scenario of an algorithm in production at Netflix. In this case, adding more than 2 million training examples has very little to no effect.

![](/blog/images/PerformanceVsTrainingSize.png)
Fig 5. Testing Accuracy of a real-life production model

So, this leads to our first lesson learned, which in fact will expand over several of the following ones: it is not about more data versus better algorithms. That is a false dichotomy. Sometimes you need more data, and sometimes you don’t. Sometimes you might need to improve your algorithm and in others it will make no difference. Focusing exclusively on one or the other will lead to far from optimal results.

## 2. You might not need all your “Big Data”

This second lesson is in fact a corollary of the previous one, but I feel it is worth to mention explicitly on its own. It seems like nowadays everyone needs to make use of all their “Big Data”. Big Data is so hyped that it seems like if you are not using huge quantities of data you must be doing something wrong. The truth though, as discussed in lesson 1, is that there are many problems for which you might be able to get similar results by using much less data than the one you have available.

Think for example of the Netflix Prize where you had 0.5 Million users in the dataset. In the most favored approach, the data was used to compute a Matrix of 50 factors. Would the result change much if instead of the 0.5 M users you used, say 50 Million? Probably not.

A related, and important, question is how do you determine what subset of your data to use. A good initial approach would be to random sample your original data to obtain as many samples you need for your model training. That might not be good enough though. Staying with the Netflix Prize example, users might be very different and not homogeneously distributed in our original population. New users, for example, will have much fewer ratings and increase sparsity in the dataset. On the other hand, they might have a different behavior from more tenured users and we might want to make our model capture it. The solution is to use some form of [stratified sampling](http://en.wikipedia.org/wiki/Stratified_sampling). Setting up a good stratified sampling scheme is not easy since it requires us to define the different strata, and decide what is the right combination of samples for the model to learn. However, as surprising as it might sound, a well-defined stratified sampled subset might accomplish even better results than the original complete dataset.

Just to be clear, I am not saying that having lots of data is a bad thing, of course it is not. The more data you have, the more choices you will be able to make on how to use it. All I am saying is that focusing on the “size” of your data versus the quality of the information in the data is a mistake. Garner the ability to use as much data as you can in your systems and then use only as much as you need to solve your problems.

## 3. The fact that a more complex Model does not improve things does not mean you don’t need one

Imagine the following scenario: You have a linear model and for some time you have been selecting and optimizing features for that model. One day you decide to try a more complex (e.g. non-linear) model</span> <span style="vertical-align: baseline; white-space: pre-wrap;"> with the same features you have been engineering. Most likely, you will not see any improvement.

After that failure, you change your strategy and try to do the opposite: You keep the old model, but add expressive features that try to capture more complex interactions. Most likely the result will be the same and you will again see little to no improvements.

So, what is going on? The issue here is that simply put more complex features require a more complex model, and vice versa, a more complex model may require more complex features before showing any significant improvement.

So, the lesson learned is that you must improve both your model and your feature set in parallel. Doing only one of them at a time might lead to wrong conclusions. </div>

## 4. Be thoughtful about how you define your training/testing data sets

If you are training a simple binary classifier, one of the first tasks to do is to define your positive and negative examples. Defining positive and negative labels for samples though may not be such a trivial task. Think about a use case where you need to define a classifier to distinguish between shows that users watch (positives) and do not watch (negatives). In that context, would the following be positives or negatives?
- User watches a movie to completion and rates it 1 star
- User watches the same movie again (maybe because she can’t find anything else)
- User abandons movie after 5 minutes, or 15 minutes… or 1 hour
- User abandons TV show after 2 episodes, or 10 episode… or 1 season
- User adds something to her list but never watches it

As you can see, determining whether a given example is a positive or a negative is not so easy.

Besides paying attention to your positive and negative definition, there are many other things you need to make sure to get right when defining your training and testing datasets. One such issue is what we call *Time Travelling*. Time traveling is defined as usage of features that originated after the event you are trying to predict. E.g. Your rating a movie is a pretty good prediction of you watching that movie, especially because most ratings happen AFTER you watch the movie.

In simple cases as the example above this effect might seem obvious. However, things can get very tricky when you have many features that come from different sources and pipelines and relate to each other in non-obvious ways. *Time traveling* has the effect of increasing model performance beyond what would seem reasonable. That is why whenever you see an offline experiment with huge wins, the first question you might want to ask yourself is: “Am I time traveling?”. And, remember, Time Traveling and positive/negative selection are just two examples of issues you might encounter when defining your training and testing datasets. Just make sure you are thoughtful about how you define all the details of your datasets.

## 5. Learn to deal with (the curse of) the Presentation Bias

![](/blog/images/Screenshot2Bfrom2B2014-12-112B223A263A27.png)
Fig 6. Example of an Attention Model on a page

<div></div><span id="docs-internal-guid-a151ba49-0d50-b49d-dad4-e8c3299922b6"><span id="docs-internal-guid-704b7fa8-3d2c-0727-b62e-9a09a90f347d"></span></span><span id="docs-internal-guid-a151ba49-0d50-b49d-dad4-e8c3299922b6"></span><span id="docs-internal-guid-a151ba49-0d51-4591-bc30-f4bcaf178ac9"></span><span id="docs-internal-guid-a151ba49-0d51-8d0d-69f3-57381be5df4c"></span>
<span style="font-family: inherit;"><span style="line-height: 1; white-space: pre-wrap;">Let’s face it, users can only click and act on whatever your algorithm (and other parts of your system) has decided to show them. Of course, what your algorithm decided to show is what it predicted was good for the user. Let’s suppose that a new user comes in and we decide to show the user only popular items. The fact that a week later the user has only consumed popular items does not mean that’s what the user like. That’s the *only* thing she had a chance to consume!</span></span>

As many ([including myself](/_posts/2011-09-26-recommender-problem-presentation/)) have mentioned in the past, is important to take that into account in your algorithms and try to somehow break this “Curse of the Presentation Bias”. Most approaches to addressing this issue are based on the idea that you should “punish” items that were showed to the user but not “clicked on”. One way to do so is by implementing some presentation discounting mechanism (see [this KDD 2014 paper](http://www.cs.ubc.ca/~peil/papers/kdd2014.pdf) by the LinkedIn folks).

Another way to address the issue is to use viewed but not clicked items as negatives in your training process. This, in principle, makes sense: if a user searched for a query and ended up clicking in result number three it means the first two results were bad and should be treated as negatives… or not? The problem with this is that although the first two items were likely worse than the third one (at least in that particular context), this does not mean they were any worse than item in position 4, let alone item in position 5000, which your original model decided was no good at all. Yes, you want to remove the presentation bias, but not all of it since it responds to some hopefully well-informed decisions your model took in the first place.

So, what can we do? First thing that comes to mind is to introduce some sort of randomization in the original results. This randomization should allow to collect unbiased user feedback so as to whether those items are good or not (see some of the early publications by Thorsten Joachims such as [this one](http://www.cs.cornell.edu/People/tj/publications/radlinski_joachims_06a.pdf) or take a look at the idea of [result dithering](http://www.slideshare.net/tdunning/which-algorithms-really-matter) proposed by Ted Dunning).

Another better approach is to develop some sort of “attention model” of the user. In this case both clicked and non-clicked items will be weighted by the probability that the user noticed them in the first place depending on their location on the page (see [some of the recent work by Dmitry Lagun](http://scholar.google.com/citations?user=95PaR-QAAAAJ&hl=en) for interesting ideas on this area.

Finally, yet another and well established way to address presentation bias is by using some sort of explore/exploit approach, in particular multi-armed bandits. By using a method such as [Thompson Sampling](http://en.wikipedia.org/wiki/Thompson_sampling), you can introduce some form of “randomization” on the items that you are still not sure about, while still exploiting as much as you can from what you already know for sure (see Deepak Argawal’s [Explore/Exploit approach to recommendations](http://www.ueo-workshop.com/wp-content/uploads/2013/10/UEO-Deepak.pdf) or one of the many publications by [Thorsten Joashims](http://www.cs.cornell.edu/People/tj/) for more details on this).

## 6. The UI is the only communication channel between the Algorithm and what matters most: the Users

![](/blog/images/Screenshot2Bfrom2B2014-12-112B223A293A37.png)
Fig 7. The UI is the algorithm’s connection point with the user |

From the discussion in the previous lesson it should be clear by now how important it is to think about the presentation layer and the user interface in our machine learning algorithmic design. On the one hand, the UI generates all the user feedback that we will use as input to our algorithms. On the other hand, the UI is the only place where our algorithms will be shown. It doesn’t matter how smart our ML algorithm is. If the UI hides its results or does not give the user the ability to give some form of feedback, all our efforts on the modeling side will have been in vain.

Also, it is important to understand that a change in the user interface might require a change in the algorithms and vice versa. Just as we learned before that there is an intimate connection between features and models, there is also another to be aware of between the algorithms and the presentation layer.

## 7. Data and Models are great. You know what is even better? The right evaluation approach.

![](/blog/images/Screenshot2Bfrom2B2014-12-112B223A303A42.png)
Fig 8. Offline/Online Innovation Approach

This is probably one of the most important of the lessons in this post. Actually, as I write this I feel that it is a bit unfortunate that this lesson might seem as “just another lesson” hidden in position 7. This should be a good place to stress that these lessons in this post are not sorted from more to less important, they are just grouped in topics or themes.

So, yes, as important as all the other discussions about data, models, and infrastructure may be, they are all rather useless if you don’t have the right evaluation approach in place. If you don’t know how to measure an improvement you might be endlessly spinning your wheels without really getting anywhere. Some of the biggest gains I have seen in practice have indeed come from tuning the metrics to which models were being optimized.

Ok, then what is the “right evaluation approach”? Figure 8 illustrates an offline/online approach to innovation that should be a good starting point. Whatever the final goal of your machine learning algorithm is in your product you should think of driving your innovation in two distinct ways: offline and online.

![](/blog/images/Screenshot2Bfrom2B2014-12-112B223A323A32.png)
Fig 9. Offline Evaluation

<span style="font-family: inherit; vertical-align: baseline; white-space: pre-wrap;">First, you should generate datasets that allow to try different models and features in an offline fashion by following a traditional ML experimentation approach (see Figure 9): You train your model to a training seat, you probably optimize some (hyper)parameters to a validation set, and finally measure some evaluation metrics on a test set. The evaluation metrics in our context are likely to be IR metrics such as precision and recall, ROC curves, or ranking metrics such as [NDCG](http://en.wikipedia.org/wiki/Discounted_cumulative_gain), MRR, or FPC (Fraction of [Concordant Pairs](http://en.wikipedia.org/wiki/Concordant_pair)). Note though that the selection of the metric itself has its consequences. Take a look at Figure 10 for an example of how the different ranking metrics weight different ranks being evaluated. In that sense, metrics such as MRR or (especially) NDCG will give much more importance to the head of the ranking, while FPC will be weighting more on the middle of the ranks. The key here is that depending on your application you should choose the right metric.</span>

![](/blog/images/Screenshot2Bfrom2B2014-12-112B223A323A48.png)
Fig. 10. Importance given to different ranks by typical ranking metrics

Offline experimentation is great because once you have the right data and the right metric it is fairly cheap to run many experiments with very few resources. Unfortunately, a successful offline experiment can only be generally used as an indication of a promising approach worth testing online. While most companies are investing in finding better correlation between offline and online results, this is still, generally speaking, an unsolved issue that deserves more research (see [this KDD 2013 paper](http://dl.acm.org/citation.cfm?id=2488215), for example).

In online experimentation the most usual approach is to do A/B testing (other approaches such as [Multiarmed Bandit Testing](https://support.google.com/analytics/answer/2844870?hl=en) or [Interleaved Testing](http://www.cs.cornell.edu/people/tj/publications/chapelle_etal_12a.pdf) are becoming more popular recently but are beyond the scope of this post). The goal of an A/B test is to measure difference in metrics across statistically identical populations that each experience a different algorithm. As with the offline evaluation process, and perhaps even more here, it is very important to choose the appropriate evaluation metric to make sure that most if not all decisions on the product are data driven.

Most people will have a number of different metrics they are tracking in any AB test, but it is important to clearly identify the so-called Overall Evaluation Criteria (OEC). This should be the ultimate metric used for product decisions. In order to avoid noise and make sure the OEC maps well to business success it is better to use a long-term metric (e.g. customer retention). Of course, the issue with that is that you need time, and therefore resources, to evaluate a long-term metric. That is why it is very useful to have short-term metrics that can be used as initial early reads on the tests in order to narrow down worthwhile hypothesis that need to wait until the OEC read is complete.

If you want more details on the online experimentation piece there are many good reads, starting with the many good articles by Bing’s Ronny Kohavi (see [this](http://ai.stanford.edu/~ronnyk/2013%20controlledExperimentsAtScale.pdf), for example).

## 8. Distributing algorithms? Yes, but at what level?

There always comes a time in the life of a Machine Learning practitioner when you feel the need to distribute your algorithm. Distributing algorithms that require of many resources is a natural thing to do. The issue to consider is at what *level* does it make sense to distribute.

We distinguish three levels of distribution:
- Level 1. For each independent subset of the overall data
- Level 2. For every combination of the hyperparameters
- Level 3. For all partitions in each training dataset

In the first level we may have subsets of the overall data for which we need to (or simply can) train an independently optimized model. A typical example of this situation is when we opt for training completely independent ML models for different regions in the world, different kinds of users, or different languages. In this case, all we need to do is to define completely independent training datasets. Training can then be fully distributed requiring no coordination or data communication.

In the second level, we address the issue of how to train several models with different hyperparameter values in order to find the optimal model. Although there are smarter ways to do it, let’s for now think of the worst-case grid search scenario. We can definitely train models with different values of the hyperparameters in a completely distributed fashion, but the process does require coordination. Some central location needs to gather results and decide on the next “step” to take. Level 2 requires data distribution, but not sharing since each node will use a complete replica of the original dataset and the communication will happen at the level of the parameters.

Finally, in level 3 we address the issue of how to distribute or parallelize model training for a single combination of the hyperparameters. This is a hard problem, but there has been a lot of research put into it. There are different solutions with different pros and cons. You can distribute computation over different machines splitting examples or parameter using, for example, [ADMM](http://stanford.edu/~boyd/admm.html). Recent solutions such as the [Parameter Sever](http://parameterserver.org/) promise to offer a generic solution to this problem. Another option is to parallelize on a single multicore machine using algorithms such as [Hogwild](http://www.eecs.berkeley.edu/~brecht/papers/hogwildTR.pdf). Or, you can use the massive array of cores available in GPU cards.

As an example of the different approaches you can take to distribute each of the levels, take a look at [what we did](http://techblog.netflix.com/2014/02/distributed-neural-networks-with-gpus.html) in our distribution of Artificial Neural Networks over the AWS cloud (see Figure 11 below for an illustration). For Level 1 distribution, we simply used different machine instances over different AWS regions. For Level 2 we used different machine in the same region and a central node for coordination. We used Condor for cluster coordination (although other options such as StarCluster, Mesos, or even Spark) are possible. Finally, for level 3 optimization, we used highly optimized CUDA code on GPUs.

![](/blog/images/DistributedANN-Final.png)
Fig 11. Distributing ANN over the AWS cloud

## 9. It pays off to be smart about your Hyperparameters

<div><span id="docs-internal-guid-b7ec9469-3d35-8cb0-546e-5738b3abc26a"><span style="font-family: inherit;"><span style="line-height: 1; white-space: pre-wrap;">As already mentioned in the previous lesson, one of the important things you have to do when building your ML system is to tune your hyperparameters. Most, if not all, algorithms will have some hyperparameters that need to be tuned: learning rate in matrix factorization, regularization lambda in logistic regression, number of hidden layers in a neural network, shrinkage in gradient boosted decision trees… These are all parameters that need to be tuned to the validation data.
</span></span></span>
<span style="font-family: inherit;"><span style="line-height: 1; white-space: pre-wrap;">
</span></span>

<span style="line-height: 16px; white-space: pre-wrap;">Many times you will face situations in which models need to be periodically retrained and therefore hyperparameters need to be at least fine-tuned. This is a clear situation where you need to figure out a way to automatically select the best hyperparameters without requiring a manual check. As a matter of fact, having an automatic hyperparameter selection approach is worthwhile even if all you are doing is the initial experimentation. A fair approach is to try all possible combinations of hyperparameters and pick the one that maximizes a given accuracy metric on the validation set. While this is, generally speaking, a good idea, it might be problematic if implemented directly. The issue is that blindly taking the point that optimizes whatever metric does not take into account the possible noisiness in the process and the metric. In other words, we can’t be sure that if point A has an accuracy that is only 1% better than point B, point A is a better operating point than B. </span>
<span style="line-height: 16px; white-space: pre-wrap;">
</span>
<span style="line-height: 16px; white-space: pre-wrap;">Take a look at Figure 12 below, which illustrates this issue by showing (made up) accuracy results for a model given different values of the regularization parameter. In this particular example the highest accuracy is for no regularization, plus there is a relatively flat plateau region for values of lambda between 0.1 and 100. Blindly taking a value of lambda of zero is generally a bad idea since it points to overfitting (yes, this could be checked by using the test dataset). But, beyond that, going to the “flat region”, is it better to stick with the 0.1 value? By looking at the plot I would be inclined to take 100 as the operating point. This point is (a) non-zero, and (b) noise-level different in terms of accuracy from the other non-zero values. So, one possible rule of thumb to use is to keep the highest non-zero value that is noise level different in terms of the optimizing metric from the optimal point.</span></div><div><span style="font-family: inherit;"><span style="line-height: 1; white-space: pre-wrap;">
</span></span></div>

![](/blog/images/Screenshot2Bfrom2B2014-12-112B223A363A34.png)

| **Fig 12.** Example of model accuracy vs. regularization lambda |

<span style="font-family: inherit;"><span style="line-height: 1; white-space: pre-wrap;">I should also add that even though in this lesson I have dsf about using a brute-force grid search approach to hyperparameter optimization, there are much better things you can do which are again beyond the scope of this post. If you are not familiar with Bayesian Optimization, start with [this paper](http://papers.nips.cc/paper/4522-practical-bayesian-optimization-of-machine-learning-algorithms.pdf) or take a look at [Spearmint](https://github.com/JasperSnoek/spearmint) or [MOE](http://engineeringblog.yelp.com/2014/07/introducing-moe-metric-optimization-engine-a-new-open-source-machine-learning-service-for-optimal-ex.html).</span></span>

## 10. There are things you can do Offline and there are things you can’t… and there is Nearline for everything in between

<div><span style="line-height: 16px; white-space: pre-wrap;">In the lessons so far we have talked about the importance of data, models, UI, metrics… In this last lesson I thought it was worth to focus on systems and architecture. When the final goal of your ML model is to have impact on a product, you are necessarily going to have to think about the right system architecture. </span>
<span style="line-height: 16px; white-space: pre-wrap;">
</span>

<span style="line-height: 16px; white-space: pre-wrap;">Figure 13 depicts a three level architecture that can be used as a </span><span style="line-height: 16px; white-space: pre-wrap;">blueprint for any machine learning system that is designed to have a customer impact. The basic idea is that it is important to have different layers in which to trade off latency vs. complexity. Some computations need to be as real-time as possible to quickly respond to user feedback and context. Those are better off in an online setting. On the other extreme, complex ML models that require large amounts of data and lengthy computations are better done in an offline fashion. Finally, there is a Nearline world where operations are not guaranteed to happen in real-time but a best effort is performed to do them as “soon as possible”.</span></div><div><span style="font-family: inherit;"><span style="color: #666666; line-height: 1; white-space: pre-wrap;">
</span></span></div>

![](/blog/images/MachineLearningArchitecture-v3.jpg)](/blog/images/MachineLearningArchitecture-v3.jpg)
Fig 13. This three level architecture can be used as a blueprint for machine learning systems that drive customer impact.

Interestingly, thinking about these three “[shades of latency](https://gigaom.com/2013/03/28/3-shades-of-latency-how-netflix-built-a-data-architecture-around-timeliness/)” also helps breaking down traditional machine learning algorithms into different components that can be executed in different layers. Take matrix factorization as an example. As illustrated in Figure 14, you can decide to do the more time-consuming item factor computation in an offline fashion. Once those item factors are computed, you can compute user factors online (e.g. solving a closed-from least squares formulation) in a matter of milliseconds in an online fashion.

![](/blog/images/Screenshot2Bfrom2B2014-12-112B223A393A21.png)
Fig 14. Decomposing matrix factorization into offline and online computation

If you are interested in this topic take a look at our original [blog post](http://techblog.netflix.com/2013/03/system-architectures-for.html)in the Netflix tech blog.

## Conclusions

The ten lessons in this post illustrate knowledge gathered from building impactful machine learning and general algorithmic solutions. If I had to summarize them in 4 short take away messages those would probably be:

1. Be thoughtful about your data
2. Understand dependencies between data and models
3. Choose the right metric
4. Optimize only what matters

I hope they are useful to other researchers and practicioners. And, would love to hear about similar or different experiences in building real-life machine learning solutions in the comments. Looking forward to the feedback.

## Acknowledgments

Most of the above lessons have been learned in close collaboration with my former Algorithms Engineering team at Netflix. In particular I would like to thank [Justin Basilico](https://twitter.com/JustinBasilico) for many fruitful conversations, feedback on the original drafts of the slides, and for providing some of the figures in this post.

## Original video and slides

[![Original Vvideo of the talk](https://img.youtube.com/vi/WdzWPuazLA8/0.jpg)]
(https://www.youtube.com/watch?v=WdzWPuazLA8)

<iframe allowfullscreen="" frameborder="0" height="355" loading="lazy" marginheight="0" marginwidth="0" scrolling="no" src="//www.slideshare.net/slideshow/embed_code/41571741" style="border-width: 1px; border: 1px solid #CCC; margin-bottom: 5px; max-width: 100%;" width="425; text-align: center;"> </iframe>

```
---
id: 16
title: "2015: My Year in\_Running"
date: '2015-12-26T00:00:00+00:00'
author: xamat
##layout: post
permalink: /2015-my-year-in-running-1682ea7310b0/
header:
  teaser: https://amatriain.net/blog/images/20-01.png
reading_time:
    - ''
    - ''
categories:
    - Uncategorized
---
```

2015 has been a great year for me in many ways, but running has been particularly good. This year I ran 2000 miles (3200 km), compared to the 1850 last year. This is even more remarkable if I think about the fact that I switched jobs to a more demanding position at [Quora](http://www.quora.com) and I added almost 1 hour of daily commute. So, yes, you can.

![](/blog/images/20-01.png)

2,000 miles!</figcaption></figure>But, besides the amount of mileage I was able to put in, there are other memorable things I'd like to share.

### Races

This year I ran 6 official races: one 50 Miler, 3 full Marathons, and 2 Half Marathons. You can see all my race results in my [athlink page](http://www.athlinks.com/athletes/167221250).

#### The North Face Ultra Challenge 50 Miles

I ran [this beautiful race](https://www.thenorthface.com/get-outdoors/endurance-challenge/california.html) again this year. This was my second time after trying it out last year. The race is located in the Marine County Headlands, a beautiful, and hilly, countryside, very close to San Francisco. There are breathtaking views of the Pacific as you make your way through the 50 miles and the 10,000 elevation feet.

Having ran it before made me much more confident I would finish. I also had a few lessons I had learned the year before. In particular, I was sure I was not going to be wasting so much time in the aid stations.

With all that in mind, and with the help of my friend Alberto, who ran it all the way with me, I was able to finish in 11:36. Almost 1.5 hours less than last year.

#### The SF Marathon

The [San Francisco Marathon](http://www.thesfmarathon.com/?utm_source=google&utm_medium=cpc&utm_content=twoonesix&utm_campaign=scenic_race) is a hard and hilly marathon. The first part is beautiful since you run across the Golden Gate and back. The second half is painful although it is supposed to be slightly downhill from there on.

I run this year for the second time. I wasn't focusing on making a good time. However, I managed to do 3:34, which is way better than the first time.

![](/blog/images/20-02.jpeg)

#### The Surfer's Path Marathon

This was the first marathon I ever ran. It is a [beautiful race](http://www.surferspathmarathon.com/) along the Santa Cruz coast. It was fun to run it again this year. I felt really great that day. So much, that I actually matched my marathon PR of 3:24 even without meaning to do it. Actually, if I had known I was so close to beating my PR I would have pushed harder on the last few miles. Nevertheless, it is still great that I was able to technically beat my PR by a few seconds given that this marathon is much harder than the St. Georges Marathon, where I PR'd the year before.

![](/blog/images/20-03.jpeg)

#### The Zombie Runner's Halloween Marathon

I "had to" run [this marathon](http://www.coastaltrailruns.com/h_halloween.html)c simply because it was in my back yard. This marathon actually went through the trail that I use for my daily runs, so it was hard to say no, especially when a couple of friends were also running. It was a really tough “hybrid” trail/asphalt marathon with a pretty significant elevation gain. Very good training for the ultras, but my all time worst time in a marathon.

#### The Halves

I did not focus at all in “short distances” this year. Just ran a couple of halves (one was the Kaiser Permanente in SF and the other one in Los Gatos). Actually, for both of them I was somewhat injured and took them easy. Very far from my Half PR of 1:31. I know I would really need to focus on training for beating this.

#### The Relay

This year was my 3rd year in a row running [The Golden Gate Relay](http://www.therelay.com/). This is a really fun 200 mile team relay going from Calistoga in Napa Valley all the way to Santa Cruz. It is always lots of fun not only to run, but also to spend time with (sweaty) friends in a van.

This year, our team (Korrikalaris) really crushed it. We were 6th, out of more than 150 teams, and finished in 22:57. We were less than a minute away from making top 5!

### The Grand Canyon R2R2R

This was probably the highlight of my year in running. After a failed attempt last year, we were determined to do it this time around. And, everything went just as planned. There was water in all the water points in the canyon (which was very fortunate since we did not get that last year and just a day after we ran there were water problems). We finished the adventure in around 15 hours.

![](/blog/images/20-04.jpeg)

[The R2R2R](http://www.impactmagazine.ca/features/35-feature-articles/549-rim2rim2rim) is a bucket list run for all distance runners. It is around 48 miles and 11000 feet of elevation gain. You start in the south rim, go down to the canyon, run across, run up the north rim, run down, run across the canyon again, and finish with a brutal climb to the south rim again. All this surrounded by beautiful scenery.

There are many details that you need to figure out in order to run this. One of them, as I already mentioned, is water supply. Another one is weather. There are actually only two small windows during the year when you can run it, because of the huge difference of temperature between the rims and the canyon. Last year we attempted to do it in May, this time around in October. The weather was almost perfect since it was cloudy and mild all around. It did rain during the last 3-4 hours and that wasn’t so fun. But, we preferred a bit of rain to heat or cold.

### Writing about running

Having joined Quora at the beginning of the year, I started to be much more active in writing on the site. I have written about many things, running included. Actually some of my answers on running have become quite popular. [Here] (https://www.quora.com/profile/Xavier-Amatriain/answers) are links to some of them.

### Injuries

I have been pretty lucky this year with injuries. Otherwise, of course, I wouldn’t have made it to the 2000 miles. However, that doesn’t mean I have been injury-free all year. I started the year with problems in my ribs due to a mountain bike incident at the end of last year. I also had calf problems in one of my halves. This has been the only race that I had to walk for a couple of miles. Most of the year I had issues with my [psoas muscle](https://en.wikipedia.org/wiki/Psoas_major_muscle) that I managed to keep under control with stretches and massages. Finally, the last part of the year, I experienced the infamous [plantar fasciitis] (http://www.mayoclinic.org/diseases-conditions/plantar-fasciitis/basics/definition/con-20025664). I will give some tips and tricks on how to overcome some of these in the Quora questions about them. You should look forward to my answers.

Still, I have to be grateful that I have been able to mostly keep on running all year and I have not even had to take a sick day in all of 2015!

### Looking forward to 2016

So, what’s next? I don’t have big plans for next year. I will be running the awesome Big Sur Marathon in March. I am also looking forward to running the Relay again this year, this time around probably with the Quora team. Besides that, I might focus on trying to beat my marathon PR in a faster marathon. If I feel really, really well I might attempt to beat my Half PR, but that is much more unlikely.

In any case, I do look forward to a happy running 2016. See you on the roads!

```
---
id: 21
title: "10 More lessons learned from building real-life Machine Learning systems - Part\_I"
date: '2016-01-05T00:00:00+00:00'
```

```
author: xamat
##layout: post
permalink: /10-more-lessons-learned-from-building-real-life-machine-learning-systems-part-i-b309cafc7b5e/
header:
  teaser: https://amatriain.net/blog/images/19-01.png
reading_time:
    - ''
    - ''
categories:
    - Uncategorized
---
```

Over a year ago, following an original [presentation](http://www.slideshare.net/xamat/10-lessons-learned-from-building-machine-learning-systems) at MLConf, I wrote a blog post entitled "[10 Lessons Learned from building ML systems](http://technocalifornia.blogspot.com/2014/12/ten-lessons-learned-from-building-real.html)". At that point, I was leading the Algorithms Engineering team at Netflix and those lessons reflected lessons we had learned there over the last few years. When you do a post/presentation like that, you don't really know how it is going to be received. Some things might be obvious to many while others might be controversial and some will not agree. It turns out though that it was very well received and referenced elsewhere (e.g. <http://machinelearningmastery.com/lessons-learned-building-machine-learning-systems/> or <http://techjaw.com/2015/02/11/10-machine-learning-lessons-harnessed-by-netflix/>).

With that in mind, a year later, I now decided to follow up with 10 new lessons that built upon the original ones. The present two-part blog post includes new lessons not only learned directly at Quora but also from talking to many people at different companies.

It is worth noting that although most of these lessons are intended to be general, they do focus on user-facing internet-scale machine learning applications. All of them are definitely valid for applications such as search or recommendation. If you are working on areas like image processing or speech recognition they might be somewhat less applicable to you (would love to hear in the comments if that is the case).

### 10 More Lessons

Without further ado, let's get into the first 5 new lessons.

#### Lesson 1. implicit signals beat explicit ones (almost always)

Many have talked about how implicit feedback is, at least in some cases, more useful than explicit feedback. For example, I wrote about this at length on a two-part [post](http://techblog.netflix.com/2012/04/netflix-recommendations-beyond-5-stars.html) in the Netflix tech blog some time ago. There, we acknowledged that while the famous 5 stars at Netflix were a good initial simplification of the recommendation problem, there is much more to gain from other information such as what users actually decide to watch or not.

<figure>![](http://localhost:8080/wordpress/wp-content/uploads/2022/06/13qBYZdwjb38e5ripN3qmoA.png)
<figcaption></figcaption></figure>Another famous example of the same effect is Youtube [moving away from their 5-star ratings](http://techcrunch.com/2009/09/22/youtube-comes-to-a-5-star-realization-its-ratings-are-useless/) into a much simpler thumbs up/thumbs down model (See [here](http://youtube-global.blogspot.com/2009/09/five-stars-dominate-ratings.html) for the original blog post).

![](/blog/images/19-01.png)

So, is implicit feedback **always** more useful. And, if so, why is that?

Let's start by describing a bit better what implicit/explicit feedback is. I define implicit feedback as "information gathered from actions not directly recognized as giving feedback by the user". These include any action where the user is selecting, clicking, or making a choice. In other words, all those actions in which the user is not (consciously) informing the system about any preference.

One benefit of implicit data is that it is usually more dense and representative of all users and scenarios. Any product usage log is easily converted into implicit feedback. It does not require any specific action by the user. As such, it is not biased towards, for example, users who are more vocal and decide to express their preferences. Also, it does not require the user to be an expert on the product or the domain. As soon as you start using the product, you are giving implicit feedback.

Implicit feedback is also more representative of user behavior vs. user reflection. In other words, I might decide not to upvote an answer on Quora by, say, Donald Trump because I disagree with his point of view. However, that does not mean I am not interested in reading it. As a matter of fact, that piece of content might be more interesting for me to read than one that I will upvote to show sympathy or encouragement to the author. In many ways, implicit data is usually better connected to the final objective function of the product and the business. Because of this, it will also be usually better correlated with AB test result metrics.

As a quick thought experiment, let's take a look at what implicit data and explicit data tells us about films released in 2014. We'll take box office as an implicit measure of people "preferring" a film enough to decide to go and watch it. We'll take IMDb rating as an explicit measure of how much users like a film. If you take at the two lists below you will see what would be our top (unpersonalized) recommendations according to box office, and according to average rating. Which one would you choose? Of course, you can argue that rating average is not a good metric because niche content with very few, but good, ratings will be unfairly promoted. And, that is true. But, precisely this makes some of my previous points of how explicit data is affected by sparsity and user bias.

![](/blog/images/19-02.png)
![](/blog/images/19-03.png)

That said, it is not always the case that implicit feedback directly correlates well with long-term retention. A counter-example is clickbait content. You can get people to click on things that they “really did not want to click on” by showing flashy images or controversial statements, for example. Users might be “tricked” into clicking this, but later decide to leave your product because they feel they are wasting their time or do not get enough out of it. There are many examples of sites and products that have fallen into the trap of optimizing short-term metrics to then end up failing as a long-term business. Too much focus on only optimizing for implicit feedback might have that side effect. (You can read about short-term/long-term effects in AB testing in many of [Ronny Kohavi’s papers](http://www.exp-platform.com/Pages/default.aspx)).

The solution is, as many times, somewhere in the middle. You can actually combine different forms of implicit and explicit data in your ML models to account for short-term engagement but also for long-term retention. If you take a look at a Quora answer, for example, you can see that there are many actions the user can do on it: they can expand, upvote, downvote, comment, and share the answer. They can even click on the originating question. We take all of these into account in our models and features and tune them to the business long term goal and mission that is represented by the AB test metrics we care about.

![](/blog/images/19-04.png)

#### Lesson 2. your model will learn what you teach it to learn

Machine learning models don’t have a will of their own. They will just learn whatever you show them. In particular, machine learning models will respond to:

- Training data (e.g. implicit and explicit)
- Target function (e.g. probability of user reading an answer)
- Metric (e.g. precision vs. recall)

As a made-up thought experiment, related to the previous IMDB example, think about a scenario where we want to: “*Optimize probability of a user going to the cinema to watch a movie and rate it “highly” by using purchase history and previous ratings. Use NDCG of the ranking as final metric using only movies rated 4 or higher as positives.”* The information here does a pretty good job of specifying all the different constraints that we want to place on the model and what it needs to learn. Leaving any of them underspecified might lead to completely different, and maybe wrong, results. For example, if we underspecified the final metric and simply said “Use NDCG of the ranking” we might be tricking ourselves to only optimizing for the probability of going to the movie even though we are considering ratings both in the objective function and the training data.

Let’s go to another more practical example: the Quora homepage Feed:

![](/blog/images/19-05.png)

In this case, our training data will be a combination of implicit and explicit feedback as we mentioned above. The target function will try to capture the value of showing a story to a user by using a weighted combination of the different actions the user can take on the story. The weights will represent the contribution of that particular action to long-term company goals and will need to be tuned by combining product decisions with AB testing. We will predict the probability of each action and then compute expected value: v_pred = E\[V | x \] = ∑a va p(a | x). As for the final metric, we can use any ranking metric (or better, the ranking metric that we have measured better correlates to AB test metrics).

#### Lesson 3. supervised vs unsupervised learning

I find that many machine learning practitioners I talk to are not aware of how powerful and useful unsupervised learning is. Unsupervised learning can, for example, be used to reduce sparsity and fight against the curse of dimensionality. It can also be used to turn raw data into features that can be fed into other models.

As a matter of fact, a very common practical approach to many problems is to combine an initial step of unsupervised learning with a later step of supervised learning. A simple example of this is to combine some form of clustering with basic knn in order to implement collaborative filtering (see[Xue et. al](http://www.cs.ust.hk/~qyang/Docs/2005/Sigir05Xue.pdf), for example). The idea is simple: nearest-neighbor approaches are not efficient and rely on computing costly multi-dimensional distance functions. Clustering optimizes the approach by identifying users by their cluster and effectively reducing one of the dimensions (number of users) of the problem.

![](/blog/images/19-06.png)

Even if you think about Matrix Factorization, you can interpret it as either supervised or unsupervised (some would call it semi-supervised). It is supervised because there are labels that you are trying to predict in a similar way as you would be doing in a regression. However, MF can also be used and interpreted as purely a dimensionality reduction approach a la PCA or as a form of clustering (this is particularly true for Non-negative Matrix Factorization, which is often used for clustering). Actually, even if you consider a standard MF approach in which training is supervised since you are trying to minimize training error to labels, the final usage can be “unsupervised”. Instead of using the label prediction output directly, you can take in the learned factors as features for another model. This is an example of using unsupervised learning for “feature engineering”.

![](/blog/images/19-07.png)

Even Deep Learning does in its own way combine unsupervised and supervised learning. Take a look at this slide from [Yann LeCun’s tutorial at ICML’13](http://www.cs.nyu.edu/~yann/talks/lecun-ranzato-icml2013.pdf) where the different (Deep) Learning approaches are classified as Supervised or Unsupervised.

![](/blog/images/19-08.png)

In fact, Stacked Autoencoders are an interesting way to[pretain Deep ANN](http://www.jmlr.org/papers/volume11/erhan10a/erhan10a.pdf).

![](/blog/images/19-09.png)

#### Lesson 4. everything is an ensemble

When people think about what won the [Netflix Prize](https://en.wikipedia.org/wiki/Netflix_Prize), they usually think about models such as [SVD++](https://www.quora.com/Whats-the-difference-between-SVD-and-SVD++) or [RMBs](http://www.machinelearning.org/proceedings/icml2007/papers/407.pdf). The truth is that the prize was won by an ensemble. Initially, the Bellkor team was using Gradient Boosted Decision Trees to combine dozens of predictors. The final ensemble used Neural Networks to combine the 103 methods.

![](/blog/images/19-10.png)

In fact, most practical real-life machine learning applications use an ensemble approach. The question is “why wouldn’t you?”. The ensemble is going to be at least as good as the best of your methods (provided you can assume that your methods are not entirely correlated). Also, ensembles are a great way to combine very different approaches. For example you can combine content-based and collaborative filtering into a single prediction through an ensemble. And, you can use many different models in the ensemble layer (logistic regression, GBDTs, Random Forests, ANNs…).

![](/blog/images/19-11.png)

As a matter of fact, ensembles are the way to turn any model into a feature! You don’t know if you should be using Factorization Machines, Tensor Factorization, or Recurrent Neural Networks? Why not use all of them? Treat each model as a feature and feed them into an ensemble so it figures out the relative merit of each of them. Actually, it might be that a particular model is better at predicting a particular sub-area of your problem space. A non-linear ensemble will be able to figure that out!

#### Lesson 5. the output of your model will be the input of another model (and other design problems)

We have seen already how ensembles can be used to turn any existing model into a feature that will be fed into another. Chances are that if you build a useful model, someone will find a way to feed it into another. This is great for reusability, but it can easily turn things into a mess.

What this means is that if you are building a model that is likely to be used in different contexts and as input to others, you need to design it in a way that it is robust to this and it is ready to accept data dependencies. For example, if you are building a ranking model, it might be wiser to publish the rank number instead of the score. This way, a change in score distribution or scale will not affect downstream dependencies.

Furthermore, as Leon Bottou described in his recent [ICML keynote](http://icml.cc/2015/invited/LeonBottouICML2015.pdf), you should be aware of feedback loops in you machine learning system setup.

![](/blog/images/19-12.png)

Data dependencies and feedback loops though are just examples of a broader concern: we would like to treat our machine learning systems in a principled way just as we treat our software systems. Is that possible? The answer is yes… and no. Yes, because you should apply best software engineering practices to your machine learning systems. Concepts such as encapsulation, abstraction, cohesion, or low coupling, all have some parallel in machine learning. Sculley et al. from Google touch upon some of these principles in their (now popular) [paper](http://static.googleusercontent.com/media/research.google.com/en//pubs/archive/43146.pdf) on the “high-interest credit card”.

![](/blog/images/19-13.png)

However, the problem for machine learning systems and software is that there are no well known and documented design patterns. This makes reusable and principled design much more complicated than it should. This seems to me as a very interesting area for research since there is a real need and interest and not many answers (see how [this Quora question](http://qr.ae/RbmOVt) has no answer yet).

![](/blog/images/19-14.png)

I look forward to your feedback.

Onto [Part II](https://medium.com/@xamat/10-more-lessons-learned-from-building-real-life-machine-learning-systems-part-ii-93fe7008fa9#.rjcmgkgt2).
---
id: 36
title: "10 More lessons learned from building real-life Machine Learning systems — Part_II"
date: '2016-01-14T00:00:00+00:00'
author: xamat
##layout: post
permalink: /10-more-lessons-learned-from-building-real-life-machine-learning-systems-part-ii-93fe7008fa9/
header:
  teaser: https://amatriain.net/blog/images/18-01.png
reading_time:
    - ''
    - ''
categories:
    - Uncategorized

In [my previous post](https://medium.com/@xamat/10-more-lessons-learned-from-building-real-life-ml-systems-part-i-b309cafc7b5e#.olhmtobts), I introduced the ten new lessons and described the first five. Let’s directly dive into the final 5.

#### 6. the pains and gains of feature engineering

The main properties of a well-behaved machine learning feature are:

- Reusable
- Transformable
- Interpretable
- Reliable

What do these properties actually mean?

- **Reusability:** You should be able to reuse features in different models, applications, and teams
- **Transformability:** Besides directly reusing a feature, it should be easy to use a transformation of it (e.g. log(f), max(f), ∑ft over a time window…)
- **Interpretability:** In order to do any of the previous, you need to be able to understand the meaning of features and interpret their values.
- **Reliability:** It should be easy to monitor and detect bugs/issues in features

Let’s take a look at a practical example of feature engineering: Quora’s answer ranking. How can we figure out the features that are needed for developing a good model for answer ranking? First of all we need to understand how we define a “good” answer at Quora. Fortunately, we have a pretty detailed description in the [Quora Answer Policies](https://www.quora.com/What-are-the-main-policies-and-guidelines-for-answers-on-Quora). In that description we will find qualifiers such as:

- truthful
- reusable
- provides explanation
- well formatted
- …

![](/blog/images/18-01.png)

    So, how can we translate those dimensions we care about into features that we can feed into our machine learning model? Well, the trick is to think what data we have in the product that relates to some of those properties. We can use features relating to the quality of the writing, interaction features such as upvotes or comments, and finally user features such as the expertise and trustworthiness of the user in the topic.

#### 7. the two faces of your ML infrastructure

Whenever you develop any machine learning infrastructure, you need to keep in mind two different modes in which this infrastructure might be used:

- **Mode 1**: ML experimentation. In this mode we value flexibility, easiness of use, and reusability.
- **Mode 2**: ML production. In this mode we value all of the above plus performance &amp; scalability

Ideally, we want both modes to be as similar as possible. So, how can we combine them?

One possible thought is to favor experimentation and only invest in productionizing once something shows results. This might mean, for example, to have machine learning researchers using R, and later ask engineers to implement things in production in their programming language of choice. Another extreme is to do the opposite: favor production and have researchers struggle to figure out how to run experiments. For example, you could decide to have highly optimized C++ code and have ML researchers experiment only through data available in logs or the database.

The reality is that both of the options above **do not work**. They are not efficient, they are wasteful in resources, and they end up leading to at least one of the modes not functioning correctly.

The trick is to implement intermediate solutions that can address needs on both modes. One example is to have ML researchers experiment on iPython Notebooks using Python tools (scikit-learn, Theano…), use the same tools in production whenever possible, and implement optimized versions only when needed. Another option is to implement abstraction layers on top of optimized implementations so they can be accessed from more friendly experimentation tools

#### 8. why you should care about answering questions (about your model)

The value of a machine learning model is the value it brings to the product. Product owners and stakeholders in the process have expectations on how the product should behave and they want to verify them. It is important to be able to answer questions about why the model is doing something or why something failed. Model debuggability can actually bridge the gap between product design and machine learning algorithms.

I would argue that model debuggability is so important that it can end up determining, or at least influencing, the particular model to use, the features to rely on, or the tools used for the implementation.

![](/blog/images/18-02.png)
![](/blog/images/18-03.png)

As an example, we at Quora have a debug tool that allows us to analyze why we are seeing (or not seeing) a particular story in our homepage feed. This tool reports the score of not only individual stories but also of the features that refer to it. It also allows to compare different stories and understand what are the features that ended up deciding for one story to rank higher than the other. This has proved to be very useful in debugging issues and in making the product team and other stakeholders understand much better how the models behave and what matters or doesn’t.

![](/blog/images/18-04.png)
![](/blog/images/18-05.png)

#### 9. you don’t need to distribute your ML algorithm

There is an industry trend recently where it seems that you should distribute your machine learning algorithm by default. If you are not doing so, it might mean that your data is not “big enough”, right? Well, here I am to tell you that no, that’s not the case. As a matter of fact, most of what people need to do with practical machine learning applications should fit into a single (multi-core) machine. Of course, there are notable exceptions to this such as if you are building large-scale Deep Artificial Neural Networks to identify cats. But, most of us are not in that business.

![](/blog/images/18-06.png)

Of course, in order to fit things into a single machine, you need to understand other approaches. For example, you need to understand the benefits of (smart) data sampling, how to use offline processing schemes, or how to parallelize in a single machine. I actually talked about all these points in my [original 10 lessons](http://technocalifornia.blogspot.com/2014/12/ten-lessons-learned-from-building-real.html).

![](/blog/images/18-07.png)

Furthermore, I think that approaches such as Hadoop and Spark, that offer “easy” access to distributed platforms by hiding most of the complexity are, in a way, dangerous. In particular, if you care about costs or latencies, not making them transparent or easy to understand is usually not a good idea.

Here is an interesting example from Quora that illustrates some of this. At some point we realized that we had a Spark implementation that was especially inefficient. It took 6 hours and 15 machines to run something that a back-of-the-envelope computation told us it should take much less. One of our engineers spent 4 days and looked into it by, for example, analyzing how the Spark scheduler unfolded the queries. The final C++ implementation runs now on a single machine and takes only 10 minutes to compute!

#### 10. the untold story of data science and vs. ML engineering

We have all heard and read [answers](http://qr.ae/Rbl3R7) to what a Data Scientist is. Most of them talk about how they need to combine math, software, and domain expertise.

![](/blog/images/18-08.png)
![](/blog/images/18-09.png)

A different question, is though, how do Data Science teams fit into an organization. Many companies have struggled, and are still struggling, with this. Most will agree that it is important to have strong data scientists who can generate value and knowledge from the data. However, regardless of what some may say, strong data scientists with solid engineering skills are unicorns, and finding them is not scalable. This often leads to situations where data scientists need to rely on engineers to bring things to production and, on the other hand, engineers don’t want to do this because they already have enough things on their plate to worry about productionizing somebody else’s ideas.

So, what is the solution? My recommendation is to think about the innovation funnel of a typical ML-related project. There are 3 distinct phases.

The first part of the funnel is where data research happens. This is where the team is looking at data and trying to understand what are the issues in order to formulate hypothesis. Are my users clicking more on the red button on Friday evening than in other days? Do users prefer newer content even if it is potentially lower quality? How can we tackle the exploration/exploitation tradeoff?

The second part of the funnel happens once the hypothesis has been formulated and we need to implement a machine learning solution. It involves choosing the model, doing feature engineering, and implementing the solution in production. It also includes not only the initial version of the solution, but future iterations to optimize and improve the current system.

The third and final part of the funnel focuses on running online experiments (AB tests) and analyzing the results. This way we can understand whether the solution that was implemented actually worked and confirmed our initial hypothesis.

It is important to note that these three phases do not need to take a long time and they should happen as quickly as possible in the context of a very iterative and agile process.

Based on many examples that I have seen, heard of, and experienced, my proposal is for Data Science to own and lead parts 1, and 3, and for engineering to own part 2. Having Data Science own part 1 and 3 in an iterative process also allows for faster iterations since results from experiments quickly flow back into improving our hypothesis. At Quora we do this by including both Data Scientists and Engineers in all projects and having them work close together while still allowing them to have clear understanding of their focus and area of ownership in the project. See [William Chen’s answer](http://qr.ae/RbUNyl) to “What is the difference between a machine learning engineer and a data scientist at Quora?” for more details.

![](/blog/images/18-10.png)

A final note is that in order for the machine learning engineering team to be effective, it may need to broaden its definition of what ML engineering means. Again, it is hard to find a large group of engineers that are both excellent at machine learning and software engineering (just as it is hard to build a soccer team by putting together 11 players that are all goal scorers). A good ML engineering team will include all the way from coding experts with high-level ML knowledge to ML gurus with acceptable software skills.

### Conclusions

There are many non-obvious issues related to implementing practical machine learning solutions. Some of them are pretty different from what you will read in publications or even different from what is accepted to be the “common approach”. If I had to summarize these 10 new lessons in a couple of dimensions I would highlight the following:

- Make sure you teach your model what you want it to learn
- Ensembles and the combination of supervised/unsupervised techniques are key in many ML applications
- It is important to focus on feature engineering
- Be thoughtful about
- your ML infrastructure/tools
- organizing your teams

I hope these suggestions are helpful, but I will admit to at least some of them being controversial. I would love to hear more about different opinions and approaches in the comments.

```
---
id: 37
title: "What’s the relationship between machine learning and data\_mining?"
date: '2016-01-14T00:00:00+00:00'
author: xamat
##layout: post
permalink: /what-s-the-relationship-between-machine-learning-and-data-mining-8c8675966615/
header:
  teaser: https://amatriain.net/blog/images/14-01.png
reading_time:
    - ''
    - ''
categories:
    - Uncategorized
```

This is not an easy question because there is no common agreement on what “Data Mining” means. But, I am going to say that I disagree with the answer from Wikipedia that [Yuvraj Singla](https://www.quora.com/profile/Yuvraj-Singla) points to. I don’t think saying that machine learning focuses on prediction is accurate at all although I mostly agree with the definition of Data Mining focusing on the discovery of properties on the data.

So, let’s start with that: **Data Mining** is a cross-disciplinary field that focuses on discovering properties of data sets. (Forget about it being the analysis step of “knowledge discovery in databases” KDD, this was maybe true years ago, it is not anymore).

There are different approaches to discovering properties of data sets. Machine Learning is one of them. Another one is simply looking at the data sets using visualization techniques or [Topological Data Analysis](https://www.quora.com/topic/Topological-Data-Analysis)

On the other hand **Machine Learning** is a sub-field of data science that focuses on designing algorithms that can learn from and make predictions on the data. Machine learning includes [Supervised Learning](https://www.quora.com/topic/Supervised-Learning) and [Unsupervised Learning](https://www.quora.com/topic/Unsupervised-Learning) methods. Unsupervised methods actually start off from unlabeled data sets, so, in a way, they are directly related to finding out unknown properties in them (e.g. clusters or rules).

It is clear then that machine learning **can be used** for data mining. However, data mining can use other techniques besides or on top of machine learning.

Btw, to make things even more complicated, now we have a new term, [Data Science,](https://www.quora.com/topic/Data-Science) that is competing for attention, especially with Data Mining and KDD. Even the SIGKDD group at ACM is slowly moving towards using Data Science. In their website, they now describe themselves as “*The community for data mining, data science and analytics*[\[1\]](https://www.quora.com/Whats-the-relationship-between-machine-learning-and-data-mining/answer/Xavier-Amatriain?srid=cgo&share=4db6d33c#flotS)*”*. My bet is that KDD will disappear as a term pretty soon and data mining will simply merge into data science.

[\[1\]](https://www.quora.com/Whats-the-relationship-between-machine-learning-and-data-mining/answer/Xavier-Amatriain?srid=cgo&share=4db6d33c#cite-flotS)[About SIGKDD](http://www.kdd.org/about)

Originally published at [*www.quora.com*](https://www.quora.com/Whats-the-relationship-between-machine-learning-and-data-mining/answer/Xavier-Amatriain?srid=cgo&share=4db6d33c)*.*

```
---
id: 64
title: 'Where am I?'
date: '2016-02-02T18:35:00+00:00'
author: 'Xavier Amatriain'
##layout: post
permalink: /where-am-i/
header:
  teaser: https://amatriain.net/blog/images/14-01.png
categories:
    - Uncategorized
---
```

There have recently been some articles (e.g. [This](https://www.poweradmin.com/blog/51-devops-influencers-to-start-following-today/) list of influencers) that have pointed to this blog and lamented that I don’t update it regularly anymore. It is true. I now realize I should have at least posted something here to direct readers to the places where I keep posting in case they find I might have something interesting to say.

First and foremost, given that I joined [Quora](http://www.quora.com/) about a year ago, I have been using the Quora product itself to post most of my writing. You can find my profile [here](https://www.quora.com/profile/Xavier-Amatriain). I have found that I can reformulate almost anything I want to say in the form of an answer to a Quora question. Besides, my posts there get a ton of views (I am almost about to reach 2 million views in about a year) and good interactions. Also, I have written some posts in the [Quora Engineering Blog](https://engineering.quora.com/) describing some of our work.

I also keep very active on [Twitter](https://twitter.com/xamat), and every now and then I will update my [LinkedIn](https://www.linkedin.com/in/xamatriain) with some professional posts.

Recently, I gave [Medium](https://medium.com/@xamat) a try. I am not really sure how often I will update my blog there, but I am almost certain that my Medium blog will take precedence over this one. Medium is indeed a much better blogging platform than Blogger.

So, yes, I guess this is a farewell to Technocalifornia unless every now and then I decide to simply post a collection of posts elsewhere just to make sure that people visiting this blog don’t get the sense that I am not active anymore. Let me know if you feel that would be interesting for you.

```
id: 17
title: "How do you deal with strong difference of opinion among expert engineers in the\_team?"
date: '2016-03-03T00:00:00+00:00'
author: xamat
##layout: post
permalink: /how-do-you-deal-with-strong-difference-of-opinion-among-expert-engineers-in-the-team-
9bdc5b102e5f/
header:
  teaser: https://amatriain.net/blog/images/14-01.png
reading_time:
    - ''
    - ''
categories:
    - Uncategorized
---
```

This is a great question. As a TLDR; the way to deal with strong differences between engineers is not necessarily to build consensus but to have a clear owner in charge of making the final decision.

First, let me say that it is great for an organization when people not only have opinions but feel empowered to express them, even strongly. Conflict avoidance is in fact one of the leading causes of poor decisions and ultimately organizations and teams failing. It is good to encourage discussion especially when a technical decision needs to be made.

However, I have also seen situations where two or more engineers came in with radically different ideas and there was no way to come to an agreement. I have even witnessed cases where this kind of situation ended up leading to pretty bad outcomes. In all of those cases, looking back, the problem was that the decision process, including the owner, was not clear.

Here is the way to address technical decisions/discussions (or most decisions, for that matter): it is called the **consultative model**. Whenever a (complex) decision needs to be made, start by defining who will be the **owner**. The owner will ultimately be in charge of making the decision. That person will also be made **accountable** for the results of the decision. In order to make the best decision possible, this owner will be encouraged to **consult** with as many engineers as they see fit. Engineers involved in the discussion will be encouraged to express their opinion as strongly as they want. However, they will all accept that it will be the owner the one who will make the final decision. It is up to this owner to make sure the different parts feel like their opinions are being taken into account. However, the goal is not build consensus. The owner can, for example, say things like “I totally get what you are saying, and this is a valuable opinion. However, we are going to go for the other option because of X and Y”.

A couple more things are important when you implement a model like this one:

First, you need to make sure people understand the meaning of “decide and commit”. This means that once a decision has been made, regardless of what your opinion was during the discussion, you should commit to what was decided and work with the group to implement it. This is a really key trait of great team players, and one that can be learned, by the way.

Finally, in this model you should also account for bad decisions being made. As I mentioned before, the main correction mechanism is holding owners accountable for their decisions. However, sometimes it is best not to wait for the outcome of the decision because it might be very costly. It is important that just as engineers feel empowered to voice their opinions during the discussion phase, they should also feel empowered to **escalate** concerns whenever they worry that a bad decision was made. Escalation is not a failure of this kind of system, it is a feature that should be used as much as needed.

Originally published at [*www.quora.com*](https://www.quora.com/How-do-you-deal-with-strong-difference-of-opinion-among-expert-engineers-in-the-team/answer/Xavier-Amatriain)*.*

```
---
id: 20
title: 'Evil ads, ad blockers, and the end of Web diversity'
date: '2016-04-01T00:00:00+00:00'
author: xamat
##layout: post
permalink: /evil-ads-ad-blockers-and-the-end-of-web-diversity-1e996a396f8c/
header:
  teaser: https://amatriain.net/blog/images/17-01.png
reading_time:
    - ''
    - ''
categories:
    - Uncategorized
---
```

Ad blockers might be good for the individual consumers that are using it, short term. However, longer term, if successful, ad blockers will shape the Internet in a way that will end up hurting consumers. Let me illustrate this by talking about an obscure Spanish site called [La Frikipedia.](https://en.wikipedia.org/wiki/La_Frikipedia)

```
Frikipedia was a controversial parody of Wikipedia. It was pretty popular among geeks in Spain because of
its smart humor, especially in tech-related topics. The site was involved in many controversies (see
Wikipedia article linked above) until it finally decided to shut down in 2015. In the final goodbye (see
[Despedida y cierre](http://frikipedia.es/) only in Spanish), the owners of the site cite ad blockers as one
of the main reasons they can't make enough money to pay for servers and legal defense. They mention that
their audience is typically tech-savvy and that results in most of them blocking ads, leaving them with no
resource to make money.

**Ad blockers &amp; Web diversity**

The reality right now is that "only" between 5 and 10% of users block ads. I honestly don't expect (and
hope) this to grow much more. As people start using more ad blocking technology, "good" ads will also learn
to circumvent. Otherwise, the end result would resemble the Frikipedia story I mentioned above. You should
expect small, targeted sites to close down one after the other.

It turns out that ads is one of the most democratic and user-friendly business model for media content
producers. This is somewhat counterintuitive since we would all agree that as users we don't necessarily
benefit from ads. But, think for a minute, if tomorrow all ads stopped working in the internet, what would
be the sites that could continue operating? The largest ones that have built a sufficiently large scale to
make any other business model feasible. In a world without ads I could see people paying a monthly fee for
Facebook, Google, or the New York Times. Would anyone pay for a small site offering cooking recipes?
Probably not.

Even companies like Netflix with no ads are basing their success on scale. And [some analysis]
(https://www.quora.com/What-are-the-biggest-risks-to-Netflixs-business-model/answer/Sizheng-Chen-
%E9%99%88%E6%80%9D%E6%94%BF) point out that not having ads is actually one of their main risks.

**Ads are not evil**

Even some people have a completely [negative position](https://medium.com/@dtunkelang/the-end-game-for-ad-
supported-content-3a477645dd2e#.msu7dgint) on ads and ad-supported businesses, I think it is important to
understand that ads in themselves are not evil. Yes, we all have experiences with annoyingly bad ads that
interfere in your user experience. However, it is also true that many of us like seeing some ads. For
example, I really enjoy when I get targeted ads on Facebook or Twitter and I get "informed" of an online
sale for running gear or a nearby race I might want to run. On the other hand, there is also a ton of
"content" that is even worse than ads. I am sure we have all fallen victim of clicking on one of those
"recommended" links that drive to really bad content farms (see example below).

![](/blog/images/17-01.png)

    Some "content" can be much worse than ads</figcaption></figure>So, ads can be good if targeted and
driving good commerce. And they can be awful, just as content itself can. Hating ads just because they "make
money" makes absolutely no sense.

So, ads are not evil. And, even if ad blocking seems like a benefit to consumers in the short term, it will
end up creating a less diverse and more concentrated media landscape. This will end up hurting consumers
much more than ads themselves.

*Originally published at* [*www.quora.com*](https://www.quora.com/Is-Adblock-good-for-
consumers/answer/Xavier-Amatriain)*.*
---
id: 6
title: 'Football or Futbol? Or why Deep Learning will not make other Machine Learning approaches obsolete'
date: '2016-04-20T00:00:00+00:00'
author: xamat
##layout: post
permalink: /football-or-futbol-or-why-deep-learning-will-not-make-other-machine-learning-approaches-
obsolete-666658ed4167/
header:
  teaser: https://amatriain.net/blog/images/16-01.png
reading_time:
    - ''
    - ''
categories:
    - Uncategorized
---

You have probably heard a lot about Deep Learning and how it is taking over the world in general, and the
area of Machine Learning in particular. But, does that mean that Deep Learning will be the solution to all
our (machine learning) problems? No. There are several reasons why there will always be a place for other
algorithms to be better suited than deep learning in some applications.

**Need for feature engineering**

There are many cases where you need to have an understanding of the domain in order to have optimal results.
```

While some proponents of Deep Learning describe their approach as being general-purpose, I don’t think that will ever be true.

Occam’s razor

Given two models that perform more or less equally, you should always prefer the one that is less complex. Even the authors of the [Deep Learning](http://www.deeplearningbook.org/) book mention this. For this reason, there will always be cases where Deep Learning will not be preferred, even if it has managed to squeeze an extra 1% in accuracy on the testing set.

Ensembles

Even in the very unlikely case where most ML problems end up being suited for Deep Learning Approaches, there will always be a place for ensembles. Given the output of a Deep Learning prediction, I am sure you will be able to combine it with some other model or feature to improve the results.

A simple example: Football or Futbol?

Let’s imagine that your very demanding boss asks you to implement an image classifier to detect whether sport images contain scenes coming from a [Football (US)](https://www.quora.com/topic/Football-US) game or a [Football (Soccer)](https://www.quora.com/topic/Football-Soccer-2) game. You are well read in Deep Learning literature, so you train a two-class classifier using [Convolutional Neural Networks (CNNs)](https://www.quora.com/topic/Convolutional-Neural-Networks-CNNs) feeding it thousands of labeled images containing both sports. The classifier works out pretty good and you get an accuracy of, say, 95%.

![](/blog/images/16-01.png)

Football or Futbol?</figcaption></figure>You present the work to your boss and she is super impressed. Next thing you know, your deep classifier is being used on a dataset of images from Spain. Now of course, every now and then, the image is really poor quality and blurry, so the classifier has a hard time. Let’s say it finds an image where the probability of it being Football is 60%, should it decide to classify it as such? Of course not! An image with 60% probability of being Football coming from a Spanish database should be classified as Futbol for sure… almost nobody watches Football in Spain!

As a matter of fact, in this example, a dumb classifier that just labels everything as Futbol for this dataset would manage to have much more than a 95% accuracy and would work much better than your fancy Deep Neural Network!

So, what’s going on? Well, you should have thought better about the problem you had at hand and realized that it it not only an image recognition problem. Things like “popularity” or “location” could make a huge difference. Of course these are simple features, you could think about many more, but you get the point.

You could feed those features into your problem by using an ensemble as mentioned above. Or you could use a simple Bayesian prior. You could even force it as a feature into the Neural Network.

In any case, this simple example proves that you will always have the need to understand your domain and do some feature engineering, favor simple models whenever possible, and most likely use ensembles.

Originally published at [*www.quora.com*](https://www.quora.com/Will-deep-learning-make-other-Machine-Learning-algorithms-obsolete/answer/Xavier-Amatriain)*.*

```
---
id: 34
title: Curai
date: '2018-06-28T00:00:00+00:00'
author: xamat
##layout: post
permalink: /curai-6408bbc78b87/
header:
  teaser: https://amatriain.net/blog/images/15-01.png
reading_time:
    - ''
    - ''
categories:
    - Uncategorized
---
```

I haven’t updated this channel in a while. I hope that by now you have already found out elsewhere that about a year ago I co-founded a company in the AI/Healthcare space: [Curai](https://www.curai.com). A few weeks back, we came out of stealth so I thought it would be interesting to do a quick post gathering some of the resources that can help you understand what we are up to.

Before I do that, let me quickly summarize who we are and what we are up to. The three co-founders are Neil Hunt, former Chief Product Officer at Netflix of 18 years, Neal Khosla, Stanford CS/Math graduate with strong ties to the startup/entrepreneurship scene, and myself. We have gathered a terrific team of engineers, AI and product experts from companies like Facebook, Google, or Netflix. Our mission is “Scaling the world’s best healthcare to every human being”. We are using artificial intelligence / machine learning with a user-centric focus to provide instant medical expertise that is accurate, trustworthy, relevant, and actionable. We will be unveiling more details about our product and technology in the future, but please do reach out to me if you want to learn more than what you will find in the resources below.

![](/blog/images/15-01.png)

A good follow up to the paragraph above with more details about who we are is our foundational [blog post](https://medium.com/curai/using-ai-ml-to-scale-the-worlds-best-healthcare-to-every-human-being-8cbc56df21d6).

You might also be interested to read about our Series-A founding in this [Techcrunch article](https://techcrunch.com/2018/05/30/curai-picks-up-10-7m-to-create-a-smarter-system-to-help-patients-supply-the-best-info-for-their-doctors/).

If you want to start getting a bit more technical, [this video] (https://youtu.be/N9D60zDUr48) introduces the space and some of the challenges we are tackling in the intersection of ML/AI and medicine:

One particular aspect that we are focusing on is the design of a conversational engine for medicine. [These slides](https://www.slideshare.net/xamat/learning-to-speak-medicine) from my most recent talk introduce some of the latest research in this space.

Finally, I am happy to say that a year after starting the company, we have already had our first paper accepted to a conference, the Machine Learning for Healthcare conference (MLHC). You can read [the preprint](https://arxiv.org/abs/1804.08033) on ArxiV.

We are growing strong and adding engineers, ML/AI researchers, physicians, and medical informaticists. Please reach out to me if you'd like to hear more about it.

```
![](/blog/images/15-02.jpeg)
---
id: 14
title: "The Memo\_Culture"
date: '2018-07-01T00:00:00+00:00'
author: xamat
##layout: post
permalink: /the-memo-culture-41a2debefe2d/
header:
  teaser: https://amatriain.net/blog/images/14-01.jpeg
reading_time:
    - ''
    - ''
categories:
    - Uncategorized
---
```

Some days back I was having a conversation about the importance of memos and documents in an agile/fast-moving startup company. The person I was talking with was surprised with my thoughts and asked me to send some links so they could follow up and share with others in the company. Unfortunately, after some searching around, I did not find anything online that was comprehensive enough to include all my thoughts. So, here I am trying to be good to the internet by filling in the void on a topic that is important but maybe not "cool" enough for tech leaders to spend much time sharing. To be clear, I don't claim to be an expert on this, and, I would love to hear your feedback.

I have been leading technical teams for many years now, and something that I have learned to appreciate is the importance of… writing documents. Yes, you heard me right. I know what some of you might be thinking now: this guy has not read the Agile Manifesto and doesn't understand that in-person meetings are much better than documentation or that code and tests are the best documentation for any piece of software. As a matter of fact, I have reached this conclusion after many years of trying many flavors of agile methodologies, both formal and informal. I now believe that having the **right** documentation is precisely one of the best tools to be agile and move fast. As Eric Ries describes in his [Lean Startup](http://theleanstartup.com/) book, the main goal of a past-paced organization is to maximize learning rate. And, in order to maximize learning rate, you need to use data and document your decisions.

But, you don't need to trust my word on this though. The most agile and innovative companies that you can probably think of, from Netflix to Amazon, and also including Facebook, Google or the likes, they all use a flavor of what I will call "The Memo Culture".

![](/blog/images/14-01.jpeg)

So, what is the memo culture? I use this term to loosely refer to company cultures in which generating some form of document or memo is an important part of that culture.

The most extreme and clear example is Bezos' Amazon, where every meeting needs to be preceded by a short document written by the organizer of the meeting (see [here](https://www.cnbc.com/2018/04/23/what-jeff-bezos-learned-from-requiring-6-page-memos-at-amazon.html), for example). The document is shared right at the meeting and the first few minutes of every meeting are used for attendees to read the document. There is no expectation of anyone reading the document before the meeting. There are different goals of such process at Amazon:

- Make sure that any meeting that is called is actually needed. If you, as the organizer, are not willing to spend a few hours preparing a document it is very likely the meeting is a waste of time.
- Make sure the meeting has clear goals and focus
- Make sure everyone in the meeting has the right context before the discussion starts
- Use the memo as documentation of the meeting and its final resolutions

I have seen several variations of this particular culture. Most companies won't really **require** a document for every single meeting, but rather establish a subset of meetings that do benefit from them. I find this to be a less extreme and more productive process. That said, it does have the downside of not strictly leading to a reduction in meetings.

Another variation that I like is sharing the document **before** the actual meeting. If you use an online collaborative tool like Googledocs or Quip, starting the discussion on the document before the meeting can be extremely useful. As a matter of fact, I have seen many times situations where the discussions on the comment threads in the doc where so useful that a meeting was no longer needed. The other benefit is that discussions on the doc give a clear preview to the organizer of what elements need more clarification and discussion during the meeting and avoid the need to start with an open-ended discussion to gauge general opinions. The obvious downside is that following this process you are not guaranteed to reduce the overall time spent on making a decision.

It is also important to address the two elephants in the room: **Powerpoint** and **meetings**.

I have seen organizations using Powerpoint (or equivalent) as their main documentation format. I hope I can easily convince you that this is plain wrong. Powerpoint is designed as a one-directional tool for a presenter to backup their presentation with visuals. Period. Any extension of this is likely to fail miserably. A set of slides, for example, tends to maximize visually appealing presentations and leave out many details that will be filled in by the presenter. Trying to reconstruct those details from only the slides is most of the time next to impossible. Also, slides are not a great vehicle for collaboration and online discussion. So, please, do your organization a favor and do limit the use of Powerpoint to external presentations where all the issues above do not matter that much. If you don't trust me on the potential evils of Powerpoint, you can read [Wired](https://www.wired.com/2003/09/ppt2/), [New York Times](https://www.nytimes.com/2010/04/27/world/27powerpoint.html?src=me&ref=homepage), and [Harvard Business Review](https://hbr.org/2010/04/powerpoint-is-evil-redux) writing about it.

Regarding meetings, I know that some agile methodologies promote the value of "in-person communication and I agree there is some truth to this. However, it is also true that many organizations that attempt to become "agile" end up throwing their employees into a non-sustainable cadence of meetings. If your technical people are spending over 20% of their time on meetings, you probably have a problem (see [this article](https://dzone.com/articles/meetingsa-culture-of-excess-that-hurts-our-product) for some good thoughts on the problem with meetings heavy cultures). Collaborative documentation is a way to avoid this. Remember the Bezos rules for memos is precisely designed to avoid useless or costly meetings and to make them more useful. Yes, excessive useless documentation should be avoided, but the same is true for excessive useless meetings. And, the good news is that both these things tend to balance each other if managed right.

Documents used in the memo culture should be treated as "living documents". This means that they should go through as many short iterations as possible during the initial phases. Those iterations should include comments and feedback from all stakeholders (see Jeff Bezos' [recommendation](https://www.cnbc.com/2018/04/23/what-jeff-bezos-learned-from-requiring-6-page-memos-at-amazon.html) on how to write good memos). It is likely that the discussions on the document might lead to some meetings where those can be addressed and discussed in person with the relevant participants. Even once the initial iterations on the document are finalized, it should not be considered as "closed" since new revisions could come at any point in time.

This gets me to the important question of how to keep documents updated. I unfortunately don't have a silver bullet for you here. I totally understand why developers prefer to keep documentation close and coupled to the code to avoid them drifting apart and updated. That is hard to accomplish with other kinds of documents. My suggestion here is to treat documents as an evolving entity and use versioning and change logs as much as possible. Furthermore, as you will see in the next section, some documents (e.g. postmortem or AB Test design) never become outdated. You should push your organization to write mostly evergreen documents and avoid as much as possible those that are likely to become easily outdated.

### Some example documents

As mentioned above, I am not a fan of enforcing a policy like Amazon's where every single meeting should include a document. However, this does bring the follow up question of what meetings and what documents usually make sense. Here are some that I found useful. Do not treat this as an exhaustive list, but rather a collection of examples, which might be more or less relevant depending on your organizational details.

#### Software/system design doc

This document is used to describe a software component or system before it is implemented. It is important to understand that a design document should not attempt to capture all details in the final design but instead be used as a way to clarify that all people involved are on the same page before starting the project. This document should include the initial design agreement or “contract” and will then be updated as the project evolves to reflect the reality of the final design as closely as possible. Therefore, the document should be used as high-level technical reference (the low level reference being the code itself).

A software/system design doc will usually include the following sections:

1. Use cases
2. Requirements
3. Architecture
4. Tools/libraries/components
5. Risks/unknowns
6. Expected timeline

The software/system design document is used by many organizations. You will easily find many example of templates online designed by anyone from [NASA]
(https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20160011412.pdf) to the EU Commission. Keep in mind that you want to keep your document light, and alive. You do not need a 100 page document unless you are literally designing a rocket from scratch.

#### Build vs. buy doc

The goal of this doc is to document the pros/cons of building a piece of technology vs. acquiring it from a 3rd party. Note that because pros/cons of building inhouse are very related to the cost and complexity of the design, this document will be very connected to a Software/System Design doc. As a matter of fact, it could easily become a section of the document above. On the other hand, a build vs. buy document can also be written when design details of the inhouse system are not clear enough to write an independent document.

In any case, the build vs. buy documentation should include the following:

1. Supported use cases and high level requirements
2. High level description of in-house alternative (can be skipped if this is part of the system design doc)
3. Third party alternatives
4. Pros/cons of (2) and (3)
5. Decision and rationale

#### Post-mortem

As Malcom Forbes said “Failure is success if we learn from it.” and in a fast moving organization, as I described above, you want to keep your learning rate as high as possible. However, learning from failure is not so straightforward. You need to think about why things failed, analyze the details, and propose changes so the failure does not happen again. All those things are documented in the Post-mortem document.

A post-mortem document document usually includes the following information:

1. Date of the incident and timeline
2. Summary of the incident and impact
3. Root cause summary
4. Detailed description
5. What worked well/could have worked better
6. Corrective actions

See [this great post](https://medium.com/production-ready/writing-your-first-postmortem-8053c678b90f) for more details on the postmortem process as a whole and pointers to post-mortem templates.

#### AB Test design doc

Just as any software or system design process benefits from discussing and documenting the decisions, the same is true for AB tests. If you are in a data-driven organization I don’t need to convince you of the importance of AB tests. However, even organizations that are thoroughly convinced and executing on this mantra can be really bad about documenting decisions. I have seen people at companies debating endlessly about an AB test that happened a couple of years back and everyone remembers somewhat differently. I have also experienced miscommunication between PMs, engineers, and Data Scientists on how an ongoing AB test had been designed. Finally, and very importantly, I have also witnessed AB tests that are running without a clearly specified hypothesis and metrics! The AB test design doc addresses all of this and more.

The purpose of the AB test design doc is to make sure all the stakeholders are on the same page about what is being tested and how it is being tested. It is not very different from a traditional science experiment design document really. As always, the document should be initially considered as a vehicle for feedback and discussion and eventually evolve into being documentation that should be useful years from now.

Some of the sections an AB Test Design doc should include:

1. High level description of the goals
2. Hypothesis that are being tested
3. Primary and secondary metrics
4. Cohort selection and duration of the AB test

5. Engineering design or pointer to software/system design where this is described

See [this example](https://help.optimizely.com/Ideate_and_Hypothesize/Create_a_basic_experiment_plan) of AB Test design document from Optimizely.

#### AB Test results doc

Documenting the results of an AB Test is as important or more than documenting the AB Test design itself. That said, the AB Test results can be documented as an addendum to the AB Test Design doc. I generally would recommend that particular approach. Otherwise, both documents should definitely link to each other since they are tightly coupled and you want to avoid the overhead of copying and pasting the test design details into a new doc.

That being said, the goal of this doc/section is to explain the details of the results of test, document the final decision and its rationale. As such, the document should include the following:

1. Test result on primary/secondary metrics
2. Interpretation/discussion of the results
3. Final decision &amp; next steps

#### Data Analysis doc

The AB Test results doc is not the only data analysis document that you should be thinking about. As a matter of fact, documenting data analysis is, in my experience, of vital importance in a data-driven organization. The last thing you want is for people to be guessing at why a decision was taken a few months back now that nobody remembers the details of the data. Documenting data analysis also enables organizations to re-visit decisions in a much more rational way.

The Data Analysis doc is, in reality, a more general purpose document than the AB Test Results one. You can use it to document anything from exploratory data analysis using product data to market research. It is hard to standardize the format because its contents can be very diverse. That said, any data analysis doc should probably include:

1. Goal of the analysis
2. Description of the data analyzed including possible shortcomings, etc…
3. Data analysis results (of course with lots of pretty graphs and plots)
4. Recommendations

### Conclusions

I hope I have convinced you by now that writing the right kind of documents can increase your organization’s velocity and long-term success. To be clear, this is not an area I claim to be an expert on, so I would love to get feedback and other ideas from you.

#### More where this came from

This story is published in [Noteworthy](http://blog.usejournal.com), where thousands come every day to learn about the people &amp; ideas shaping the products we love.

Follow our publication to see more product &amp; design stories featured by the [Journal](https://usejournal.com/?utm_source=usejournal.com&utm_medium=blog&utm_campaign=guest_post) team.

```
---
id: 11
title: "The year in AI/ML advances: 2018\_roundup"
date: '2018-12-23T00:00:00+00:00'
author: xamat
##layout: post
permalink: /the-year-in-ai-ml-advances-2018-roundup-db52f7f96358/
header:
  teaser: https://amatriain.net/blog/images/13-01.png
reading_time:
    - ''
    - ''
categories:
    - Uncategorized
---
```

It has become a sort of tradition for me to try to summarize ML advances at this time of the year (see [here](https://www.quora.com/What-are-the-most-significant-machine-learning-advances-in-2017/answer/Xavier-Amatriain) for my Quora answer last year, for example). As always, this summary will necessarily be biased by my own interests and focus, but I have tried to keep it as broad as possible. Note that what follows is a blog post version of my Quora answer [here](https://www.quora.com/What-were-the-most-significant-machine-learning-AI-advances-in-2018/answer/Xavier-Amatriain?ch=2&srid=cgo).

If I had to summarize the main highlights of machine learning advances in 2018 in a few headlines, these are the ones that I would probably come up:

- AI hype and fear mongering cools down
- More focus on concrete issues like fairness, interpretability, or causality
- Deep learning is here to stay and is useful in practice for more than image classification (particularly for NLP)
- The battle on the AI frameworks front is heating up, and if you want to be someone you better publish a few frameworks of your own

Let’s look at all of this in some more detail.

If 2017 was probably the cusp of fear mongering and AI hype, 2018 seems to have been the year where we have started to all cool down a bit. While it is true that some figures have continued to push their message of AI fear, they have probably been too busy with other issues to make of this an important point of their agenda . At the same time, it seems like the press and others have come at peace with the idea that while self-driving cars and similar technologies are coming our way, they won’t happen tomorrow. That being said, there are still voices defending the bad idea that [we should regulate AI](https://www.theverge.com/2017/7/17/15980954/elon-musk-ai-regulation-existential-threat) instead of focusing on regulating its outcomes.

It is good to see that this year though, the focus seems to have shifted to more concrete issues that can be addressed. For example, there has been a lot of talk around **fairness** and there are not only several conferences on the topic (see [FATML](http://www.fatml.org/) or [ACM FAT](https://fatconference.org/)) even some online courses like [this one](https://www.blog.google/technology/ai/new-course-teach-people-about-fairness-machine-learning/amp/) by Google.

![](/blog/images/13-01.png)

Fairness online course by Google</figcaption></figure>Along these lines, other issues that have been greatly discussed this year include **interpretability**, **explanations**, and **causality**. Starting with the latter, **causality** seems to have made it back to the spotlight mostly because of the publication of Judea Pearl’s “The Book of Why”. Not only the author decided to write his first “generally accessible” book, but he also took to Twitter to popularize discussions around causality. In fact, even the popular press has written about this as being a “challenge” to existing AI approaches (see this [article in The Atlantic](https://www.theatlantic.com/technology/archive/2018/05/machine-learning-is-stuck-on-asking-why/560675/?single_page=true), for example). Actually, even the best paper award at the ACM Recsys conference went to a paper that addressed the issue of how to include causality in embeddings (see “[Causal Embeddings for Recommendations](https://recsys.acm.org/recsys18/session-2/)”). That being said, many other authors have argued that causality is somewhat of a theoretical distraction, and we should focus again on more concrete issues like interpretability or explanations. Speaking of explanations, one of the highlights in this area might be the publication of the paper and code for [Anchor](https://github.com/marcotcr/anchor), a follow up to the well-known [LIME](https://homes.cs.washington.edu/~marcotcr/blog/lime/) model by the same authors.

![](/blog/images/13-02.jpeg)

Fairness online course by Google</figcaption></figure>While there are still questions about the Deep Learning as the most general AI paradigm (count me in with those raising questions), while we continue to skim over the nth iteration of the discussion about this between Yann LeCun and Gary Marcus, it is clear that Deep Learning is not only here to stay, but it is still far from having reached a plateau in terms of what it can deliver. More concretely, during this year Deep Learning approaches have shown unprecedented success in fields different from Vision, ranging from Language to Healthcare.

In fact, it is probably in the area of **NLP**, where we have seen the most interesting advances this year. If I had to choose the most impressive AI applications of the year, both of them would be NLP (and both come from Google). The first one is [Google’s super useful smart compose](https://ai.googleblog.com/2018/05/smart-compose-using-neural-networks-to.html?m=1), and the second one is their [Duplex](https://ai.googleblog.com/2018/05/duplex-ai-system-for-natural-conversation.html) dialog system.

![](/blog/images/13-02.png)

Fairness online course by Google</figcaption></figure>A lot of those advances have been accelerated by the idea of using **language models**, popularized this year by Fast.ai’s [UMLFit](http://nlp.fast.ai/classification/2018/05/15/introducting-ulmfit.html) (see also “[Understanding UMLFit](https://yashuseth.blog/2018/06/17/understanding-universal-language-model-fine-tuning-ulmfit/)”). We have then seen other (and improved) approaches like Allen’s [ELMO](https://arxiv.org/abs/1802.05365), [Open AI’s transformers](https://blog.openai.com/language-unsupervised/), or, more recently Google’s [BERT](https://github.com/google-research/bert), which beat many SOTA results out of the gate. These models have been described as the “Imagenet moment for NLP” since they show the practicality of transfer learning in the language domain by providing ready-to-use pre-trained and general models that can be also fine-tuned for specific tasks. Besides language models, there have been plenty of other interesting advances like [Facebooks multilingual embeddings](https://code.fb.com/ml-applications/under-the-hood-multilingual-embeddings/), just to name another one. It is interesting to note that we have also seen how quickly these and other approaches have been integrated into more general NLP frameworks such as [AllenNLP’s](https://github.com/allenai/allennlp/releases/tag/v0.7.2) or Zalando’s [FLAIR](https://github.com/zalandoresearch/flair).

![](/blog/images/13-03.png)

    Fairness online course by Google</figcaption></figure>Speaking of **frameworks**, this year the “war of the AI frameworks” has heated up. Surprisingly, **Pytorch** seems to be catching up to **TensorFlow** just as [Pytorch 1.0 was announced](https://developers.facebook.com/blog/post/2018/05/02/announcing-pytorch-1.0-for-research-production/). While the situation around using Pytorch in production is still sub-optimal, it seems like Pytorch is catching up on that front faster than Tensor Flow is catching up on usability, documentation, and education. Interestingly it is likely that the choice of Pytorch as the framework on which to implement the Fast.ai library has played a big role. That being said, Google is aware of all of this and is pushing in the right direction with the [inclusion of Keras](https://www.tensorflow.org/guide/keras) as a first-class citizen in the framework or the addition of key developer-focused leaders like [Paige Bailey](https://dynamicwebpaige.github.io/info/). At the end, we all benefit from having access to all these great resources, so keep them coming!

![](/blog/images/13-04.png)

    Fairness online course by Google</figcaption></figure>Interestingly, another area that has seen a lot of interesting developments in the framework space is **Reinforcement Learning**. While I don’t think RL research advances have been as impressive as in previous years (only the recent [Impala](https://arxiv.org/pdf/1802.01561.pdf) work by DeepMind comes to mind), it is surprising to see that in a single year we have seen all major AI players publish an RL Framework. Google published the [Dopamine framework](https://github.com/google/dopamine) for research while Deepmind (also inside of Google) published the somewhat competing [TRFL](https://deepmind.com/blog/trfl/) framework. Facebook could not stay behind and published [Horizon](https://research.fb.com/publications/horizon-facebooks-open-source-applied-reinforcement-learning-platform/) while Microsoft published [TextWorld](https://www.microsoft.com/en-us/research/blog/textworld-a-learning-environment-for-training-reinforcement-learning-agents-inspired-by-text-based-games/?OCID=msr_blog_textworld_icml_tw), which is more specialized for training text-based agents. Hopefully, all of this open source goodness will help us see a lot of RL advances in 2019.

Just to finish up on the frameworks front, I was happy to see that Google recently published [TFRank](https://ai.googleblog.com/2018/12/tf-ranking-scalable-tensorflow-library.html) on top of Tensor Flow. Ranking is an extremely important ML application that is probably getting less love than it deserves lately.

It might seem like Deep learning has ultimately removed the need to be smart about your data, but that is far from true. There are still very interesting advances in the field that revolve about the idea of improving data. For example, while **data augmentation** has been around for some time and is key for many DL applications, this year Google published [auto-augment](https://ai.googleblog.com/2018/06/improving-deep-learning-performance.html?m=1), a deep reinforcement learning approach to automatically augment training data. An even more extreme idea is to train DL models with **synthetic data**. This has been tried in practice for some time and is seen as key to the future of AI by many. NVidia presented interesting novel ideas in their [Training Deep Learning with Synthetic Data](https://arxiv.org/abs/1804.06516) paper. In our “[Learning from the Experts](https://arxiv.org/abs/1804.08033)”, we also showed how to use expert systems to generate synthetic data that can then be used to train DL systems even after combining with real-world data. Finally, also interesting is the approach of reducing the need to have large quantities of hand-labelled data by using “**weak supervision**”. [Snorkel](https://blog.acolyer.org/2018/08/22/snorkel-rapid-training-data-creation-with-weak-supervision/amp/?__twitter_impression=true) is a very interesting project that aims at facilitating this approach by providing a generic framework.

![](/blog/images/13-05.png)

    Fairness online course by Google</figcaption></figure>As far as more foundational **breakthroughs** in AI, it might be me and my focus, but I haven’t seen many. I don’t entirely agree with Hinton when he says that this lack of innovation is due to the field having “[a few senior guys and a gazillion young guys](https://www.wired.com/story/googles-ai-guru-computers-think-more-like-brains/)” although it is true that there is a trend in science where [breakthrough research is done at a later age](https://phys.org/news/2011-11-breakthrough-scientific-discoveries-longer-dominated.html). In my opinion, the main reason for the current lack of breakthroughs is that there are still many interesting practical applications of existing approaches and variations so it is hard to risk in approaches that might not be practical right away. That is even more relevant when most of the research in the field is sponsored by large companies. In any case, an interesting paper that does challenge some assumptions is “[An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling](https://arxiv.org/pdf/1803.01271.pdf)”. While being highly empirical and using known approaches, it opens the door to uncovering new ones since it proves that the one that is usually regarded as optimal is in fact not. To be clear, I do not agree with [Bored Yann LeCun](https://twitter.com/boredyannlecun) that Convolutional Networks are the ultimate “master algorithm”, but rather that RNNs are not either. Even for sequence modeling there is a lot of room for research!. Another highly exploratory paper is the recent NeurIPS best paper award winner “[Neural Ordinary Differential Equations](https://arxiv.org/abs/1806.07366)”, which challenges a few fundamental things in DL including the notion of layers itself.

![](/blog/images/13-06.png)

    Fairness online course by Google</figcaption></figure>Interestingly, this last paper was motivated by a project where the authors were looking into **healthcare** data (more concretely Electronic Health Records). I cannot finish this summary without referring to the area of research in the intersection of AI and Healthcare since that is where my focus at Curai is at. Unfortunately, so much is going on in this space that I would need another post only for this. So, I will only point you to the papers that were published at the [MLHC conference](https://www.mlforhc.org/) and the [ML4H NeurIPS workshop](https://ml4health.github.io/2018/). Our team at [Curai](https://www.curai.com) managed to get papers accepted at both, so you will find our papers there among many other interesting ones that should give you an idea of what is going on in our world.

```
---
id: 23
title: "Cultural over/under-fitting and transfer learning. Or why the “Netflix Culture” won’t work in
your\_company."
date: '2019-02-12T00:00:00+00:00'
author: xamat
##layout: post
permalink: /cultural-overfitting-and-underfitting-or-why-the-netflix-culture-won-t-work-in-your-company-
af2a62e41288/
header:
  teaser: https://amatriain.net/blog/images/12-01.jpeg
reading_time:
    - ''
    - ''
categories:
    - Uncategorized
---
```

![](/blog/images/12-01.jpeg)

A couple of weeks back, I gave a talk in the [SFELC](https://sfelc.com/summit2019) (San Francisco Engineering Leadership Conference). As I was preparing the slides for the talk I reflected on something pretty interesting: I have been managing technical teams for over 25 years. I have also been giving public talks for about the same time. However, this was the first time I was giving a talk about managing teams. I was also surprised that, despite a packed house and many people congratulating me after the talk, my tweets about the talk and the conference got very little engagement. I guess technical management doesn’t get as much traction as AI/Machine Learning. I get it. In any case, one of the most important points I made during the talk directly related to a some machine learning concepts like transfer learning, or under/over-fitting. So, maybe by sprinkling some of the “AI hype” onto management cultureI will be able to get people’s attention on some very important issues for not only any company, but also every founder, manager, and potentially any employee .

### **Overfitting and underfitting**

In very simple terms, underfitting happens when we try to explain a complex real-world phenomenon with a model that is too simple. As an example, this often happens when we “rush” to simplistic conclusions to explain something after just observing one of the causes without realizing that there are many more. On the other hand, overfitting happens when we do the opposite: we use too complex a model to explain a phenomenon. This complex model might explain the current observation, but does not generalize well to similar situations. While over/under-fitting is a very important issue in machine learning, it is also present in every day life in many ways and forms. We humans underfit and overfit all the time.

![](/blog/images/12-02.png)

### Examples of over/under-fitting in company culture

So now that we have the “theory” under our belt, let’s see some examples of underfitting and overfitting in company culture. I will refer to some popular management books for that purpose.

First, let’s start with **underfitting**. The way to uncover underfitting is to look for the following pattern: someone claims (or implies) that the success of a given organization is explained by one (or a very limited set of) aspect. A good example of this is the book “Measure What Matters” by John Doerr. As much as I like OKRs, if you read that book you might come to the conclusion that all the companies listed in the case studies (e.g. Google) have been successful because of OKRs. While OKRs might indeed have played a minor role, it should be clear that there are many other variables that have had an impact on why Google and others in the book have been successful. You could even argue that OKRs may have not **caused** them to be successful, but rather the other way around. Other successful cultural and organizational traits caused the companies to adopt OKRs, so these are reflecting success rather than causing it. In other words, and for the geeks reading this post, while there might be some correlation, there is no causation.

There are as many or probably even more examples of **overfitting**. The pattern we are looking for here is someone claiming that a complex model that was successful in one (or a very limited set of) company, will generalize to others. Of course, you can already see how this pattern repeats in almost every other management book: famous business person A explains what they did at successful company X and will claim you should do the same. A prime example of this is [Ray Dalio trying to convince you](https://www.amazon.com/Principles-Life-Work-Ray-Dalio/dp/1501124021/ref=sr_1_2?ie=UTF8&qid=1549859434&sr=8-2&keywords=principles+ray+dalio) that the “extreme” culture he implemented at Bridgewater is going to work for you. There are actually many such books or posts all around. Laszlo Bock [will try to convince you](https://www.amazon.com/Work-Rules-Insights-Inside-Transform/dp/1455554790) that the Google culture works by just giving you one example of success: Google itself. Patty McCord does the same thing in her book [Powerful](https://www.amazon.com/Powerful-Building-Culture-Freedom-Responsibility/dp/1939714095/ref=sr_1_1?ie=UTF8&qid=1549930869&sr=8-1&keywords=powerful+patty+mccord) when trying to get her point across that the

Netflix Culture should generalize well and apply to almost any tech company.

Let me dive into on of these examples a bit more, because I think they are very dangerous, both for companies trying to define who they are or for older companies trying to re-invent themselves. I will use the example of the Netflix Culture because it’s, out of all these, the one I know best.

I should start by saying that I **loved** the Netflix Culture. I really thrived in that environment, and when I was there I thought it was almost perfect. Even now I think it is maybe 90% good. Still, I can also say that this culture will not work anywhere else, even if you apply everything in Patty’s book step by step. For example, are you willing to give up on hiring junior candidates and/or interns just because they are likely to do poorly in such culture? Are you willing to bias your company towards some personality types that will thrive in that culture while penalizing others? Are you ok with giving up on having remote offices? And, so on. The Netflix Culture only worked at Netflix because of hundreds of small details that end up connecting directly to the business model, mission, and leadership team. I will say it once again: it will not work anywhere else.

A particularly annoying manifestation of overfitting is when people in a company use the following rationale: “Everyone who works at my company is happy and successful, that proves that my culture is right”. In *Principles*, Ray Dalio says that new employees usually have a hard time at Bridgewater during the first few months, but then end up loving it there. Patty McCord says in [*Powerful*](https://www.amazon.com/Powerful-Building-Culture-Freedom-Responsibility/dp/1939714095/ref=sr_1_1?ie=UTF8&qid=1549859487&sr=8-1&keywords=powerful+patty+mccord) that “all engineers” despise process, and that is why all are happy at a culture like Netflix’. That is, of course, BS, and it is just an example of survival bias. Everyone who “survived” at your company did so **because** they adapted to the culture and were in some way predetermined to do so. This proves nothing except that you managed to rule out everyone else.

![](/blog/images/12-03.png)

### **Transferring learnings**

In machine learning, we talk about transfer learning when a model that has been originally trained for one problem can also be applied to a different problem, maybe with some “minor tweaks”. Similarly, while in the previous section I made it sound like it is hard to apply learnings from one company to another, there are certainly many lessons that have repeated enough to be transferable. As a matter of fact, even most of what is written in the books I criticized above can be transferred if you are careful enough to adapt them to your situation and context.

After having been at very different companies, all the way from large 100 year old multinationals to my own startup with only 3 people on day one, I feel like I am in a relatively good position to highlight things that are easily transferrable and you might want to apply anywhere, and things that are much harder to adapt. Of course, I might be over/under-fitting myself, so take everything with a grain of salt.

### **Transferable lessons**

So, what are the traits that over and over I have seen contribute to a culture’s success no matter the context, and the company?

1. **People above all (aka You are exactly as good as your team)**

This might be the most important invariant: It doesn’t matter how good you are as a manager, you will fail with a bad team, and even the worst manager will seem good when leading an awesome team. Of course, it is unlikely that a great manager ends up leading a bad team or for a bad manager to assemble a great team.

So, how do you build a great team that will make you (and everyone else) successful? There is one, and only one, answer: **hiring**. As a manager/leader, recruiting should be your number one priority. This might seem an overstatement, but this is indeed an important thing that separates successful teams from unsuccessful ones, and successful companies from unsuccessful ones. And, if you still think this is extreme, you might want to know that at Netflix managers who were struggling to hire someone important to their team were encouraged to stop doing everything else and focus on that hire 100% until they closed someone. And, when I say everything, I mean everything. They would stop even attending “important” meetings.

Maybe one of the saddest moments of my hiring manager career was when a manager that I was interviewing told me that if they could, they would fire 40% of their team. That didn’t say much of the manager, and even less of the company as a whole. Interestingly, the company does not exist anymore.

There are of course other important things to keep your team happy and performing for a long time after they have been hired. It is beyond the scope of this post to go into all the different management techniques that can help. However, if I had to highlight just one that seems to valid no matter the context and the situation, that would be the importance of regular[1:1s with the team](https://css-tricks.com/the-importance-of-one-on-ones/).

2. Culture eats strategy for breakfast

It is extremely important for a company’s success to have a strong/clear culture. This might be even more important than having a strong/clear strategy, because in fact a strong culture will likely lead to a strong strategy, but not the other way around. Think about any of the successful companies right now. Google, Apple, Netflix, Salesforce, Amazon… they all have a strong clear culture that you could describe. Note that they don’t all have the **same** culture, and indeed some of them are very different, but they all have strong and clear traits that define who they are.

My take on this is that, [as Google discovered](https://www.nytimes.com/2016/02/28/magazine/what-google-learned-from-its-quest-to-build-the-perfect-team.html), one of the most important keys to success in any organization is [psychological safety](https://en.wikipedia.org/wiki/Psychological_safety). While there is much more to it, an important aspect of such safety is having clarity and understanding of how people will respond and act in the organization in response to anything you do or say. This is only possible in an environment where you have a *strong* culture and norms that dictate what everyone should expect of each other.

A word of caution here. *Strong* culture as I have used above might be misunderstood as one that strongly limits freedom or strongly selects individuals who are “fit” for it. Besides the many well-known benefits of diversity in any organization, you should build a culture to scale and therefore to include as many (different) individuals as possible. Of course, there will always be some “kinds” of people that might be more or less attracted to who you are. That will depend not only on your culture, but also your mission, product, and stage. That being said, you should do your best to minimize “culture cliques” unless you want to face the issues that many early successful companies face as soon as they start scaling up.

As an example of this, when we started our company [Curai](https://www.curai.com), I talked about setting up our cultural values asap. However, my cofounder pointed me to a post where they advised not to write down your company culture until you had at least 20 employees. After thinking about we agreed they were right. If you set up the culture with only 2 people in the room, you run the risk of that culture mirroring your own values. Having input from at least 20 peoples will make your culture much **stronger**.

3. Clarity &amp; ownership are important

As mentioned in the previous paragraph, one of the most important keys to success is clarity. No matter how big or small a company is, it is important to have clarity in mission, strategy, and ownership. Of course, such clarity is not easy, but it is on leaders and managers to push for as much clarity as possible. It is really hard for anyone to work in a confusing environment where goals are not clear and the direction is changing constantly. You might think that this is precisely the definition of early startup, and you are probably right. However, even in such environment you can seek and provide clarity.

A couple of techniques that I like are the concept of DRIship, and OKRs.

DRI, or Directly Responsible Individual, is a concept used to explicitly describe that any activity in an organization has to have a clear owner. Confusion over ownership is one of the worst things for a team. Have you ever been involved in a situation where someone thought someone else was in charge of making the decision they should have made? Of course. We all have. I won’t have time in this post to convince you why ownership matters so much, but I would recommend the book [Extreme Ownership](https://www.amazon.com/Extreme-Ownership-U-S-Navy-SEALs-ebook/dp/B00VE4Y0Z2), which makes a really compelling case of why that matters in **all** kinds of situations.

[OKRs](https://en.wikipedia.org/wiki/OKR) (Objectives and Key Results) are also a good way to provide clarity on goals and ways to measure progress in an organization. While I did refer to this book as an example of underfitting in the first section, I do think that John Doerr’s [Measure What Matters](https://www.amazon.com/Measure-What-Matters-Google-Foundation/dp/0525536221/ref=sr_1_3?ie=UTF8&qid=1549845797&sr=8-3&keywords=measure+what+matters) is well worth a read.

4. Process matters

Make no mistake: every company has some flavor of process. Of course some companies might have way too much process while others like Netflix strive to only define the MVP (minimum viable process). In any case, process matters. Process brings clarity on what to expect and how things are run. As you will read in the [Netflix slide deck](https://www.slideshare.net/reed2001/culture-1798664), there is such thing as “good process”.

Even when deciding **not to** define a process you are implicitly defining one and you are inviting others (teams, individuals) to implement their own. So, you should be intentional about what process needs to be defined to bring clarity and what process can be left out for people to figure out on their own. That decision should, of course, be directly connected to the company culture.

### What needs adaptation

There are many other things that cannot be easily transferred from one company to another. Here is my top three selection.

1. **Culture**

This might be a bit counter-intuitive. I mentioned above that having a strong culture is a “must-have” for any successful company. What I am saying here is that the details of that particular culture will not travel well from one company to another. In other words, as I mentioned in the title of this post, the Netflix Culture will not work in your company. Another way to see this is by looking at the many successful companies I listed before (Google, Apple, Netflix, Salesforce…) and realizing that their cultures are not at all similar. Some times they are very, very different.

So, why is that? I have already mentioned a few things in this post that partly explain this. A company culture needs to be aligned to the company mission, stage, business model, focus area, strategy, and people, among many others. For example, as much as I loved Netflix Culture, this culture is based on the premise that anyone you hire has to be “experienced”. I knew this was not going to be very successful when going into Quora and realizing that there were many bright recent grads, and my boss and CEO was twenty-something.

This anecdote is true for any other culture. You want to adapt Bridgewater’s culture outside of finance? Good luck with that. Please let me know if anyone is even mildly successful at that attempt. Do you want to copy Facebook’s culture in your company? Good luck if your company is anything but a social media startup. As a matter of fact, Facebook is even struggling with scaling their culture now that they are not a startup anymore!

2. Process

Similarly to culture, while process always matters, the details of the process are hard to directly transfer from one company to another. Yes, you will be able to use some concrete techniques like OKRs, but beyond that, most things will need to be adapted to your particular context. As a matter of fact, process is a representation of your culture plus your strategy, so they need to go hand in hand.

3. People

Finally, and perhaps more importantly, people are very different from one company to another. While, as I mentioned above, you should strive to make your culture as inclusive as possible, your company will attract different people, and for different reasons, than any other company.

This is key, since everything else, including culture and process, will need to align to the people you have in the company.

Is that all?

Of course not, there are many other things that matter in a company culture. For example, I am a fan of the three C’s: Challenge, Care, and Communicate. This is an extension from the two-dimensional challenge and care framework presented in [Radical Candor](https://www.amazon.com/Radical-Candor-Kick-Ass-Without-Humanity/dp/1250103509/ref=sr_1_1?ie=UTF8&qid=1549859348&sr=8-1&keywords=radical+candor). I do think those three things in combination are really key. However, I am also aware that there are many examples of successful companies that do not observe one or more of them. For example, Apple does not care too much about the “communicate”, while Netflix does not care too much about the “caring” aspect either.

### Some reading recommendations

I know in this post I have been generally critical of management book. However, there are many great books on these topics out there. Even the ones I criticized above I would recommend if you take with a grain of salt. However, if I had to pick a few to recommend, these are probably my favorite five. Note that any of them in isolation is also likely to over/under-fit. However, in combination, I think they do give a great way to start addressing culture:

- [The Advantage](https://www.amazon.com/Advantage-Organizational-Health-Everything-Business/dp/0470941529/ref=sr_1_1?ie=UTF8&qid=1549859249&sr=8-1&keywords=the+advantage)
- [The Alliance](https://www.amazon.com/Alliance-Managing-Talent-Networked-Age/dp/1625275773/ref=sr_1_1?ie=UTF8&qid=1549859280&sr=8-1&keywords=the+alliance+managing+talent+in+the+networked+age)
- [Multipliers](https://www.amazon.com/Multipliers-Best-Leaders-Everyone-Smarter-ebook/dp/B003M69A4Q/ref=sr_1_2?ie=UTF8&qid=1549859322&sr=8-2&keywords=multipliers+book)
- [Radical Candor](https://www.amazon.com/Radical-Candor-Kick-Ass-Without-Humanity/dp/1250103509/ref=sr_1_1?ie=UTF8&qid=1549859348&sr=8-1&keywords=radical+candor)
- [Extreme Ownership](https://www.amazon.com/Extreme-Ownership-U-S-Navy-SEALs/dp/1250183863/ref=sr_1_1?ie=UTF8&qid=1549859373&sr=8-1&keywords=extreme+ownership)

![](/blog/images/12-04.png)

```
---
id: 31
title: "Data as Prior/Innate knowledge for Deep Learning\_models"
date: '2019-04-18T00:00:00+00:00'
author: xamat
##layout: post
permalink: /data-as-prior-innate-knowledge-for-deep-learning-models-23898363a71a/
header:
  teaser: https://amatriain.net/blog/images/11-01.png
reading_time:
    - ''
    - ''
categories:
    - Uncategorized
---
```

There is a long history of debate about how much of human knowledge is innate and how much is learned from experience/data. This is also known as the nature vs. nurture debate (See for example some of [Gary Markus' recent papers](https://arxiv.org/pdf/1801.05667.pdf) for more details on this). Traditional Bayesian methods explicitly model these two components with a prior component that can be derived from existing knowledge and a likelihood that is learned from the data.

The recent rapid advances in deep learning continue to demonstrate the significance of end-to-end training with no apriori knowledge (such as domain-aware feature engineering) in many computer vision and NLP tasks. However, when models require reasoning or need to do forward prediction, or operate in low data regime, most AI researchers and practitioners will agree that incorporating prior knowledge, along with end-to-end training can introduce better inductive bias, beyond what is provided in the training data. This is especially true in complex domains such as healthcare where precision of the systems are central, and the combinatorics required for generalization is large.

Despite the general agreement that innate structure and learned knowledge need to be combined, there is no simple approach to incorporate this innate structure into learning systems. As a scientific community, we are just starting to research approaches to incorporate prior into deep learning models. (See some of these [recent approaches](https://openreview.net/forum?id=r1E0OsA9tX) or [discussions](https://www.quora.com/Is-there-a-way-to-encode-prior-knowledge-in-deep-neural-networks) on how to include prior knowledge in DL networks). In fact, not surprisingly, there is little consensus on whether priors needs to be hard-coded into the model or can be learned through, for example, using a reward function (see [this great summary](https://www.abigailsee.com/2018/02/21/deep-learning-structure-and-innate-priors.html) of the debate between LeCun and Manning)

The prior is, [generally speaking](https://en.wikipedia.org/wiki/Prior_probability), a probability distribution that expresses one's beliefs about a quantity before some evidence is taken into account. If we restrict ourselves to an ML model, the prior can be thought as of the distribution that is imputed before the model starts to see any data. However, a more holistic view into the process should consider not only the model training, but the design of the system as a whole, including the choice of model and the data. Therefore, the prior includes anything from the choice of the algorithm itself to the way that the data is labeled (this is in fact related to what [Markus calls](https://arxiv.org/pdf/1801.05667.pdf) innate algorithms, representational formats, and knowledge).

Therefore, a perhaps not so obvious way to inject prior knowledge and beliefs into the process is through the training data selection. Indeed, by selecting and labeling training samples for any supervised learning model, we encode knowledge into the learning model. The data labeling process might be trivial in some domains, but will require a good deal of domain knowledge in others. For example, to label cats and dogs we need to understand the difference between these two animals, which might seem very "common sense", but to label medical images as "cancer" or "not cancer", we need deep medical expertise.

As yet another example, an interesting but negative case of injecting "prior belief" in the data selection/labeling process is related to some well-known examples of algorithmic bias. If the belief of whoever is creating the predictive model is that people of a certain background are more likely to commit a crime, this belief is likely to be fed into the training and testing process by selecting and labeling a dataset that confirms that prior. In fact, most existing datasets, like collections of written texts, have priors/biases that will affect the learned model (see [Semantics derived automatically from language corpora contain human-like biases](http://science.sciencemag.org/content/356/6334/183))

In [a recent paper](https://medium.com/curai-tech/the-science-of-assisting-medical-diagnosis-from-expert-systems-to-machine-learned-models-cc2ef0b03098), we presented an approach to incorporate prior knowledge into DL systems by using synthetic data. While we presented this approach for a particular application (medical diagnosis), I believe this has broader implications that can be used in many other domains. To be clear, synthetic data has been used to some extent in other domains. For example, self-driving researchers [have used synthetic data from the Grand Theft Auto](https://www.technologyreview.com/s/602317/self-driving-cars-can-learn-a-lot-by-playing-grand-theft-auto/) videogame to train machine learning systems. However, our approach follows a flexible and reproducible process that involves the following steps:

1. Use an existing rule-based expert system to generate data
2. Inject noise to the generated data to make learned system more resilient
3. Combine synthetic data from real-world data
4. Train a deep learning system

The diagram below illustrates the process.

![](/blog/images/11-01.png)

As we showed in the paper, the process of combining synthetic data with noise plus real world data to train a deep learning model has several advantages:

1. It allows to introduce innate domain-specific knowledge
2. It allows to control the balance of innate vs. real-world learned knowledge by simply controlling the balance between synthetic and real-world data.
3. Adding noise to the synthetic data allows us to learn a system that is more robust to noisy data in inference time
4. Introducing real-world data means that we can have an extensible continuous learning system that is

better over time than the expert system specified by the domain experts.

In our paper, we show a concrete example of how this approach can be successfully applied to modeling a medical diagnosis system. However, again, I think that this could be useful for others and look forward to hearing about it being applied elsewhere or for feedback and comments.

```
---
id: 30
title: 'Making decisions in an organization'
date: '2019-08-20T00:00:00+00:00'
author: xamat
##layout: post
permalink: /making-decisions-8e8a902c0d3b/
header:
    teaser: https://amatriain.net/blog/images/10-01.png
reading_time:
    - ''
    - ''
categories:
    - Uncategorized
---
```

![](/blog/images/10-01.png)

Nobody would probably argue that it is important for any organization to make the “right” decisions. However, if you ask anyone you know how are decisions made at their company, you will most likely get vague answers at best. Maybe something like “we empower everyone to make good decisions” or “we have a distributed decision-making process”. Ask them, “What does that actually mean? Who makes a decision? When and how?” and you are likely to get a silent stare for an answer.

Of course, strong well-oiled cultures like Netflix go a step beyond and describe a few more details of the process. Netflix [culture](https://jobs.netflix.com/culture) talks about “informed captains”, “disagree openly”, and “context not control”, all of which are great aspects of a good decision-making process. However, I still find the description a bit vague for anyone outside of Netflix to really understand.

What follows is my attempt to clarify what is the optimal decision-making process for most organizations. This is based on my experience not only at Netflix, but at several other companies, including, at this point, [Curai](https://www.curai.com). [As always](https://towardsdatascience.com/cultural-overfitting-and-underfitting-or-why-the-netflix-culture-wont-work-in-your-company-af2a62e41288), I am not claiming this is a silver bullet that works always. But, I do believe this model is general enough that it should work if you have a reasonable culture to support it.

### Decision-making models

There are many models of making decisions in a group setting, but most of them can be summarized in the following three categories:

1. **Consensus-based**: decision won’t be reached until everyone in the group agrees.
2. **Majority-based**: the group votes after a discussion and the proposal with the most votes is chosen. Variations of this include things like weighted-majority.
3. **Authoritarian**: the person with the most “authority” in the group decides disregarding the opinion of the rest.

Deciding what is the right model depends on what is most important in the process. In fact, when evaluating any decision-making process it is important to look at several dimensions. The most important ones are:

- **Optimality**: How likely the model is to produce an optimal (or good) decision
- **Cost**: How long it will take to reach a decision
- **Alignment**: How hard/easy it will be to align the group with the final decision
- **Accountability**: How much clarity there will be after the fact on who is accountable for the decision

Any model will have to trade-off among those dimensions. For example, consensus-based models are generally high on alignment, but have a high cost, low accountability, and, arguably, low optimality. On the other hand, authoritarian models have low cost and high accountability, but very low alignment.

Another interesting, and somewhat controversial example, is Democracy, which can be mostly understood as a majority-based model. Democracy has relatively low cost since all it takes is to count votes. It has low optimality, as we have all learned recently, and low accountability since nobody feels responsible for the result. However, it does index high on alignment, or so it hopes. In a majority-based model everyone has had a chance to get their voice heard and it is easy to convince people to go with what “most people” want.

Unfortunately, all the above models tend to work very poorly in any company. Consensus or majority-based models are extremely inefficient and lead to sub-optimal decisions and lack of accountability. I have heard companies say they make decisions by consensus, and I hardly believe that is possible. Would love to see counter examples. On the other hand, it should be clear that authoritarian models can only get you so far and, while low-cost, are very low on optimality and alignment.

Is there a model that can be seen as the ideal tradeoff between all those competing dimensions? Yes! Meet the **Consultative decision-making model**.

### The Consultative model

This model of decision-making is based on a series of simple premises:

- The final decision will be made by an individual who is appointed ahead of time
- This individual is sometimes called [**DRI**](https://about.gitlab.com/handbook/people-operations/directly-responsible-individuals/#targetText=Apple%20coined%20the%20term%20%22directly,or%20failure%29%20of%20that%20project.), a term that was coined by Apple, stands for Directly Responsible Individual, and is used at many tech companies.
- The DRI will be held accountable for the consequences of the decision
- It is the DRI’s responsibility to include whoever is needed in the discussion phase so as to reach the best possible decision
- Once the decision is made, it is the DRI’s responsibility to make sure people are aligned
- Anyone consulted in the discussion commits to offering the best information so as to help the DRI make the right decision. They should also follow the understand/discuss/commit model described below. Consultants will be held accountable for their performance during the process, particularly when that has an impact (positive or negative) on the quality of the decision.

### The Understand-&gt;Discuss-&gt;Decide-&gt;Commit cycle

Besides the concrete model used to make the decision itself, it is important to understand that the decision process includes other activities that start before and end after the decision has been taken. In order to have an efficient and collaborative environment, it is important for everyone involved in the process to understand what is expected of them during the different phases of the process:

UNDERSTAND-&gt;DISCUSS-&gt;MAKE DECISION-&gt;COMMIT

The first two phases are pre-decision, while the last one is post-decision.

(Note that this is an extension of the popular [Disagree and Commit](https://en.wikipedia.org/wiki/Disagree_and_commit) model popularized by Sun, Intel, and more recently Amazon).

#### Understand

Anyone involved in the discussion leading to the decision should do their best to understand the constraints and the goals of the decision. It is on the DRI to clearly communicate those, but other participants should make sure they understand them and ask questions or clarifications if needed. This first phase should seek **clarity**.

#### Discuss

Once constraints and goals have been understood, people should be invited to discuss options and possibilities as openly and broadly as possible. This phase is likely to include an initial **brainstorming** activity to generate ideas. Once those ideas are generated, they should be narrowed down to the most likely and preferably ranked by all participants and ultimately the DRI. It is important to also add a **challenge** activity in which people can openly challenge the initial assumptions or the existing ranking to make sure that nothing is being missed in the discussion. This phase can take 15 minutes, or a 3 weeks since the timeline depends on the complexity of the decision and the amount of information needed to make a good decision.

Note that the goal of this phase **is not** to have everyone agree. The goal is to surface all possible relevant aspects so the decision is made with as much information as possible.

#### Commit

After the decision is made, the DRI will communicate to whoever is affected. It is very important for people to commit to the final decision, regardless of whether they agree with it or not. Getting alignment by everyone involved is extremely important to make sure the decision is then executed properly with everyone giving their best to make it succeed. It is extremely important to avoid people disengaging or even boycotting a process/project because they disagree with the decision.

As mentioned above, it is mostly on the DRI to get people to commit. An important part of doing this is communicating the decision in the right way, giving enough reasons about why the decision was taken and also acknowledging there were other options that were considered, but did not seem as good. For important decisions, it is also good to mention that the decision can be iterated on or revised at later times and give a sense of the timeline.

### Escalation

An important aspect of any decision-making process is the need to have an ability to **escalate**. If you feel strongly that the DRI is making the wrong decision or did not run the process correctly, you should feel totally ok with escalating up the DRI chain and raise your concerns to their manager or their manager's manager.

Escalation is a good thing in any system, and nobody should feel bad about doing it. Actually, many disasters in companies and history in general could have been avoided with proper escalation.

### A simple example

So, how does this work in practice? Let's run through a simple example:

We have to decide where to organize our next team lunch. Since it is unclear who should be driving this, the executive team gets together. They decide that Johnny is probably the best person so they assign the task to him. Johnny is overwhelmed at first since this is an important decision, but he then remembers how he should be using the consultative model. He identifies two people in the office who have strong opinions about food (Rob, and Lisa). He also remembers that Amber has a lot of knowledge of local restaurants and is a Yelp power user. He calls for a meeting with the 3 people. He states the goal of the meeting as follows: "Let's come up with a ranked list of options for our team lunch with pros/cons for each one". After the meeting, they end up with a ranked list of 5 options. The ranking is done roughly based on the pros/cons identified by Rob, Lisa, and Scott. At the end of the meeting, Johnny thanks them for the feedback, and tells them he will come back with a decision. He spends a couple of hours researching in detail the options and realizes that the highest option is over budget, and the second is temporarily closed. He gets back to Rob, Lisa, and Scott informing them of the final decision (originally ranked 3rd) and explains the reasons. He invites them again to voice any concern. They all respond saying that this seems great. Johny then confirms with the executive team. Finally, he sends an email to everyone in the company announcing the decision, thanking the people who helped them, and providing some brief background on how the decision was made.

```
---
id: 22
title: "The year in AI: 2019 ML/AI advances\_recap"
date: '2019-12-16T00:00:00+00:00'
author: xamat
##layout: post
header:
     teaser: https://amatriain.net/blog/images/09-01.png
reading_time:
    - ''
    - ''
categories:
    - Uncategorized
---
```

It has become somewhat of a tradition for me to do an end-of-year retrospective of advances in AI/ML (see [last year's round up](https://medium.com/@xamat/the-year-in-ai-ml-advances-2018-roundup-db52f7f96358) for example), so here we go again! This year started with a big recognition to the impact of Deep Learning when Hinton, Bengio, and Lecun were awarded [the Turing award](https://www.nytimes.com/2019/03/27/technology/turing-award-ai.html). You might think that after a few years of neck-breaking speed in innovation, this kind of recognition might be signaling that we are getting near some sort of plateau. Well, think again. That is nowhere in sight yet. It is true that hot areas have clearly shifted and while a few years ago image recognition was all the rave this year we have seen more impressive advances in language. I will go into all of this below, but let me start by summarizing the biggest headlines of AI in 2019, in my own very biased opinion:

- Computers learn to talk (i.e. language models like Bert and specially GPT-2 get **scaringly** good)
- AI becoming good at creating synthetic content has some serious consequences
- The biggest theoretical controversy continues to be how to incorporate innate knowledge or structure into machine learned models. There has been little practical progress towards this end, and little progress towards any other theoretical breakthrough.
- The revolution may get unsupervised at some point, but for now we can make it self-supervised
- Computers continue to get better at playing games and can now collaborate in multi-agent escenarios
- Other areas like Healthcare and Recommender Systems continue to see advances by using Deep Learning, but some of these advances are questioned
- The war between frameworks continues, with a major TensorFlow release and also big movements on the Pytorch arena.

![](/blog/images/09-01.png)

But, let's get right into it and dive into each of these fascinating 2019 headlines.

The year of the Language Models

I think it is hard to argue against the fact that this has been the year of Deep Learning and NLP. Or more concretely, the year of language models. Or even more concretely the year of Transformers and GPT-2. Yes, it might be hard to believe, but it has been less than a year since OpenAI first released talked about their [GPT-2 language model](https://openai.com/blog/better-language-models/). That blog post sparked a lot of discussions about AI safety since OpenAI did not feel comfortable releasing the model. Since then, the model was publicly replicated (see [here](https://medium.com/@NPCollapse/replicating-gpt2-1-5b-86454a7f26af) and

[here](https://blog.usejournal.com/opengpt-2-we-replicated-gpt-2-because-you-can-too-45e34e6d36dc)), and [finally released](https://openai.com/blog/gpt-2-1-5b-release/). However, this has not been the only advance in this space. We have seen Google publish [AlBERT](https://arxiv.org/pdf/1909.11942.pdf), [XLNET](https://arxiv.org/abs/1906.08237), and [Universal Transformers](https://ai.google/research/pubs/pub48062), and also talk about how BERT has been the [largest improvement to Google search in years](https://www.blog.google/products/search/search-language-understanding-bert). Besides Google, most of the other big players in the AI space have also published their own language models: [Salesforce](https://blog.einstein.ai/introducing-a-conditional-transformer-language-model-for-controllable-generation/), [Amazon](https://arxiv.org/abs/1904.09408), [Microsoft](https://github.com/microsoft/dialogpt), or [Facebook](https://ai.facebook.com/blog/a-new-generative-qa-model-that-learns-to-answer-the-whole-question/) seem to all have really bought into the Language Model revolution.

![](/blog/images/09-02.png)

What can these models do in practice? Besides the obvious and scary “generate credible fake tweets”, there are much more constructive ones that we have seen during this past year. For example, Google told us how they were using them not only for search as mentioned above, but also for their [Smart compose](https://arxiv.org/abs/1906.00080) feature. Facebook learned models [to answer questions wholistically](https://ai.facebook.com/blog/a-new-generative-qa-model-that-learns-to-answer-the-whole-question/) and [Allen Institute’s Aristo AI passed an eighth-grade science test](https://www.geekwire.com/2019/allen-institutes-aristo-ai-program-finally-passes-8th-grade-science-test/). In fact, if we look at the SQUAD leaderboard, it seems nowadays anyone can surpass human-level reading comprehension by combining some of these known approaches (see image below).

![](/blog/images/09-03.png)

I do expect to see many more impressive advances in this space in 2020 as it seems we are getting closer and closer to passing the Turing Test and having computers that “can speak human”. That being said, we should also temper our expectations since there have been many papers that have also identified the limitations of the current approaches. To start with, [Google’s impressive study](https://arxiv.org/abs/1910.10683) sets a good backdrop on the limitations of transfer learning in language. In [Limitations of Language Models for generating text or storytellers](https://arxiv.org/abs/1909.10705), the Stanford NLP folks walk us through situations when these language models work, and many others where they don’t. Of course, a key aspect of these limitations is the fact that these models are expected to generalize across a wide range of tasks and even domains. However, we know that, as shown in “[To Tune or Not to Tune? Adapting Pretrained Representations to Diverse Tasks](https://arxiv.org/abs/1903.05987)”, it usually pays off to fine tune models to specific tasks. Our team at Curai came to that exact conclusion when comparing general language models to those trained on the medical domain in “[Domain-Relevant Embeddings for Medical Question Similarity](https://arxiv.org/pdf/1910.04192.pdf)”. So, we are still far from having general-purpose language models that can tell good stories and adapt to different tasks and domains. Finally, I could not finish this paragraph on limitations of language models without mentioning Merity’s great “[Stop Thinking With Your Head](https://arxiv.org/abs/1911.11423)” where he shows how for many tasks a simple LSTM model can perform almost as well as the most complicated Transformer.

Combining knowledge/structure with deep learning

In 2019 we continued to hear loud voices advocating for AI not to get stuck in a Deep local maxima. According to many, me included, we should be able to combine data-intensive deep learning approaches with more knowledge-intensive methods to add some form of innate structure. While it is true that there is a lot of work to be done in that space, we did see many examples of research combining deep learning and more “traditional” AI.

In “[Transferable Multi-Domain State Generator for Task-Oriented Dialogue Systems](https://arxiv.org/abs/1905.08743)”, Salesforce presents a state-of-the-art approach for slot-filling task-oriented dialogue systems that combines deep learning with more traditional conversational methods. “[Neural Assistant: Joint Action Prediction, Response Generation, and Latent Knowledge Reasoning](https://arxiv.org/abs/1910.14613)” is a recent paper from Google that also combines several deep and knowledge-intensive approaches for the same purpose. [Wizard of Wikipedia: Knowledge-Powered Conversational agents](https://arxiv.org/abs/1811.01241?fbclid=IwAR3ca-6Dy2nP1cxl8D0iG0QxAyqLIAvJixRqApWZn5IVXSsAQXqf7LDDsEE) is Facebook’s response in that same space.

“[ERNIE: A knowledge graph-enhanced language model](https://arxiv.org/abs/1905.07129)” is a novel approach where a language model is trained not only on natural language data, but also a knowledge graph. Interestingly though, it turns out that while language models might benefit from being trained on knowledge graphs, they themselves also encode knowledge and can be used as knowledge bases (see “[Language Models as Knowledge Bases?](https://arxiv.org/abs/1909.01066)”). Similarly, deep learning models like BERT or ELMO do not only encode knowledge, but also syntax. The Stanford NLP team showed how [syntax trees can be directly inferred from such models](https://nlp.stanford.edu/~johnhew/structural-probe.html).

![](/blog/images/09-04.png)

The self-supervised revolution

If I had to name two important fundamental trends behind many of the advances highlighted above those would be: transfer learning, and self-supervision. Transfer learning (the idea that you can train a model on an original dataset and apply the resulting model elsewhere) is a pretty obvious idea behind language models, but also earlier image models trained on Imagenet and the like. The idea of self-supervision might be a bit less obvious. Maybe that is why some are calling it [The Quiet Revolution](https://towardsdatascience.com/the-quiet-semi-supervised-revolution-edec1e9ad8c) despite Yan LeCun having screamed it out loud for years to anyone who was listening. In any case, self-supervision, the idea that you can train a model on unlabelled data by exploiting the context in the data itself, is catching on. Not only language models like BERT or [ALBERT](https://arxiv.org/abs/1909.11942) use the idea extensively, but this same notion is being applied to other domains, making it easier to train on large corpuses without needing to spend huge efforts in annotation. For example, self-supervision is being used to improve image classification models. See for example "[Self-Supervised Learning of Pretext-Invariant Representations](https://arxiv.org/abs/1912.01991)", "[Data-Efficient Image Recognition with Contrastive Predictive Coding](https://arxiv.org/abs/1905.09272v2)", or the recent "[Self-training with Noisy Student improves ImageNet classification](https://arxiv.org/abs/1911.04252)". All of these approaches improve on SOTA supervised methods while using much less labeled data.

![](/blog/images/09-05.png)

A fascinating application of self-supervision that takes the idea a step further is Facebook's "[Unsupervised Question Answering by Cloze Translation](https://ai.facebook.com/blog/research-in-brief-unsupervised-question-answering-by-cloze-translation/)" where they split the question answering problem into two steps. The first steps generates synthetic training data with a model that synthesizes fill-in-the-gap questions from documents. The second step uses a traditional Q&amp;A model. This is similar to our "[Learning from the experts](https://arxiv.org/abs/1804.08033)" where we sidestep the need for costly and noisy labeling of medical data by generating synthetic training data.

Other miscellaneous research advances

The year also came with other advances that don't neatly fit into the main trends of combining knowledge with deep learning, or self-supervision. What follows are some of my favorite highlights in this miscellaneous category.

In "[The Lottery ticket hypothesis](https://arxiv.org/pdf/1803.03635.pdf)" the authors show a fascinating result: due to sheer chance, some subnetworks with many less parameters than the original network have comparable accuracy. For some reason their connections have initial weights that result in a much more effective training. The authors also present an algorithm to identify those "winning tickets". In the same vein of finding more efficient yet performing models, "[EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks](https://arxiv.org/abs/1905.11946)" introduces an approach to uniformly scale all dimensions in a CNN.

[Rectified Adam](https://medium.com/@lessw/new-state-of-the-art-ai-optimizer-rectified-adam-radam-5d854730807b) is a variation over the well-known Adam optimizer that results in better training and higher accuracy (if you don't know Adam, you probably should since, [according to Chip Huyen](https://twitter.com/chipro/status/1152079136908648448), it's the most commonly asked question during interviews).

In "[Classification Accuracy Score for Conditional Generative Models](https://arxiv.org/abs/1905.10887)" the authors present a new way to evaluate generative models by training a classifier on synthetic data but predicting labels on real data. While this is not strictly speaking a novel idea (see e.g. "[LR-GAN: Layered Recursive Generative Adversarial Networks for Image Generation](https://arxiv.org/abs/1703.01560)"), and the paper applies it only on GANs for image, it does show an interesting path for evaluating other generative models in different domains. In "[Training on Synthetic Noise Improves Robustness to Natural Noise in Machine Translation](https://arxiv.org/abs/1902.01509)" we see how injecting synthetic noise during training can improve the quality of the trained models. I personally believe that the use of synthetic data plus noise is going to bring a lot of advances in AI in the near future. But, maybe I am just biased because our own publication "[Learning from the experts: From expert systems to machine-learned diagnosis models](https://arxiv.org/abs/1804.08033)" already proposed a combination of these two techniques.

Another important line of research is on how to apply learned models in "the wild" by modeling uncertainty and out-of-distribution modeling. In real-life it is important to understand the uncertainty of model predictions and whether the data point is outside of the distribution on which the model was trained on. A few papers on this space have been presented at NeurIPS 2019 (see e.g. "[Modeling Uncertainty by Learning a Hierarchy of Deep Neural Connections](https://arxiv.org/abs/1905.13195)" and "[Likelihood Ratios for Out-of-Distribution Detection](https://arxiv.org/abs/1906.02845)". This is also a very important aspect to tackle in healthcare, and it is indeed the focus of our recent paper "[Open Set Medical Diagnosis](https://arxiv.org/abs/1910.02830)".

Finally, I should note that there have been a lot of publications in the broad space of what I would call human-AI-Interaction that includes research areas like fairness, bias, or interpretability. Hard to pick the most impactful works in this space, but I will highlight two with almost opposite takeaways. AI2 presented [AllenNLP-Interpret](https://allennlp.org/interpret), a toolkit for interactive model interpretations and explanations. This work won the EMNLP best demo award. On the other extreme, in "[Manipulating and Measuring Model Interpretability](https://arxiv.org/pdf/1802.07810.pdf)" Microsoft researchers surprisingly concluded that model transparency and interpretability not only did not help, but could hamper user ability to detect model mistakes. And one last, and very recent, piece of news on this space of human-AI is Facebook's announcement of a [$1M deep-fake detection challenge](https://ai.facebook.com/blog/deepfake-detection-challenge-launches-with-new-data-set-and-kaggle-site/). Clearly detecting fake content is going to be a huge deal in the future, and it is good to see that we are already putting efforts into this.

Let’s keep playing

It has been more than 3 years since Alpha Go beat Lee Sedol, but we are still receiving the aftershocks of such a feat with Sedol recently [announcing his retirement](https://www.businessinsider.com/deep-mind-alphago-ai-lee-sedol-south-korea-go-2019-11) because of that defeat. And, while it might seem like there is not much more progress to be made in AI for games, computers insist on getting better at more, and more complex games. This year we saw two major feats, with DeepMind reaching [human-level performance in Quake III Arena Capture the Flag](https://deepmind.com/blog/article/capture-the-flag-science) and wining the [Starcraft competition with AlphaStar](https://deepmind.com/blog/article/AlphaStar-Grandmaster-level-in-StarCraft-II-using-multi-agent-reinforcement-learning). Both these advances show us the ability of algorithms not only to master complicated but highly structured games like Go, but also to adapt to more fuzzy strategic goals in which even collaboration is needed.

![](/blog/images/09-06.png)

A final, and pretty recent, advance in this space is [Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model](https://arxiv.org/abs/1911.08265) where Deepmind again shows like a combination of search and a learned model can be applied to gain superhuman performance not only in a single game, but in a range of games.

Is there room for deep outside of text and image?

Of course the Deep revolution is impacting way beyond text and image. I will focus on the two areas I follow most closely: recommender systems and healthcare. Interestingly, I have seen a similar pattern in both areas this year (warning: you should know that us “scientists” see patterns all around us).

In recommender systems, Deep Learning has been impacting the research community for some time now, probably ever since Youtube published their[first paper on using Deep Learning for recommendations](https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/45530.pdf) in 2016. Maybe unsurprisingly, most of the results related to deep learning continue to come from industry continues. My former team at Netflix has definitely jumped on the DL train, and they have been speaking publicly about different deep learning enhancements to the Netflix recommender system. See this [recent tutorial](https://www.slideshare.net/AnoopDeoras/tutorial-on-deep-learning-in-recommender-system-lars-summer-school-2019) by Anoop Deoras on using Deep Learning for recommendations. Even Facebook, who are honestly not very active in the recommendation arena, made quite a splash this year by [open-sourcing a Deep Learning recsys model/framework](https://ai.facebook.com/blog/dlrm-an-advanced-open-source-deep-learning-recommendation-model/). But, not everything is shiny and bright on the deep side of the recsys street. As a matter of fact, the best paper award at the Recsys conference went to [a paper](https://arxiv.org/pdf/1907.06902.pdf) that questions most of the recent advances in using deep learning approaches and shows how simpler methods obtain similar to better results.

![](/blog/images/09-07.png)

In healthcare, the deep learning revolution has already been here for some time too. There have been many research papers in this intersection. So much, that Stanford/Google felt like we are at a point when we can even publish “[A guide to deep learning in healthcare](https://www.nature.com/articles/s41591-018-0316-z.epdf?author_access_token=6pwmnaFm6innDwDCVpb3ydRgN0jAjWel9jnR3ZoTv0NKvdocQpp0HGqaPYdoBZOn-AH9UBsE2ULhICB_2tJMiCRHhApSSPRBzd-SgEHKQH4qtEbGcs9Cf-AHQfYwaHZKKnBsJOeRQBEGpNaa_mMCzg%3D%3D)”. Truth be told, the most interesting/credible applications of deep learning to healthcare are actually still when applied to either images or text (see some of our own examples in “[Domain-Relevant Embeddings for Medical Question Similarity](https://arxiv.org/abs/1910.04192)” or “[Prototypical Clustering Networks for Dermatological Disease Diagnosis](https://arxiv.org/abs/1811.03066)”). However, when applied to more complex data like Electronic Health Records (EHR) we show that much simpler models perform just as well as deep neural networks (see our upcoming “The accuracy vs. coverage trade-off in patient-facing diagnosis models”).

The Framework/Platform war

Unsurprisingly, the “AI framework war” that I already mentioned in [last year’s round up](https://medium.com/@xamat/the-year-in-ai-ml-advances-2018-roundup-db52f7f96358) has not cooled down. The two main contenders continue to be Google’s TensorFlow and Facebook’s Pytorch. Who will win the war remains to be seen, but [according to some data](https://thegradient.pub/state-of-ml-frameworks-2019-pytorch-dominates-research-tensorflow-dominates-industry/), Pytorch continues to win the research battle, while TensorFlow dominates in production-ready systems.

This last year, TensorFlow released the highly-anticipated [TF 2.0](https://github.com/tensorflow/tensorflow/releases/tag/v2.0.0), its main highlights being tight integration with Keras, default eager execution mode, and more Pythonic function execution. In other words, TF is trying to become more usable and friendly… more Pytorch-like. TensorFlow also introduced its [Swift library](https://www.tensorflow.org/swift), which immediately attracted many programmers. Even fast.ai [announced that it would embrace Swift](https://www.fast.ai/2019/03/06/fastai-swift/), and maybe question the use of Pytorch over time, which would be a huge deal. On the other hand, Pytorch has continued to evolve. Probably the most notable news late this year was another popular library [Chainer merging into Pytorch](https://preferred.jp/en/news/pr20191205/).

![](/blog/images/09-08.png)

There was a lot of movement outside of the two big players too. Microsoft recently announced a really interesting [serving infrastructure](https://github.com/microsoft/onnxruntime), which provides highly efficient production-ready serving of models trained with any framework supporting the ONNX standard. This can be a huge win for Pytorch since its serving infrastructure is lagging behind TensorFlow's for sure. On the NLP-frameworks arena, I have to obviously mention the great work by the folks at [Huggingface](https://arxiv.org/abs/1910.03771) who seem to release the code to any NLP advance before the paper even hits arxiv. A huge accomplishment especially given that all big players are interested in this space. Even Uber published their own [framework for dialogue system research](https://eng.uber.com/plato-research-dialogue-system/).

![](/blog/images/09-09.png)

What to expect in 2020

Thanks for making it this far! I know this is a long post. Plus, I am always much better at explaining the past than predicting the future. So, I won't keep you here much longer. I don't have risky predictions for what 2020 will bring, but I am sure of a few things:

1. There will be more advances in NLP, some of which will be categorized as breakthroughs
2. AI will get better at faking all kinds of content, and we will see efforts on how to avoid the possible negative side effects
3. Aspects such as uncertainty modeling, out-of-distribution modeling, metalearning, or interpretability will continue to be top of mind for many.
4. AI will continue to hugely impact broad application areas such as Healthcare
5. We won't get see self-driving Teslas on the roads
6. We won't solve AGI

Hope you enjoyed the post, and looking forward to your feedback and comments!

Thanks to Anitha Kannan for feedback on an early version of this post.

```
---
id: 24
title: 'The 7+1 Habits of Highly Successful Leaders'
date: '2020-08-03T00:00:00+00:00'
author: xamat
##layout: post
header:
     teaser: https://amatriain.net/blog/images/08-01.png
permalink: /the-7-1-habits-of-highly-successful-leaders-91cdcd37765/
reading_time:
    - ''
    - ''
categories:
    - Uncategorized
---
```

I recently had a candidate I was interviewing to ask about my leadership style. I gave a short answer, but I was left thinking that I would have liked to continue talking about this for a much longer time. My "style", if there is such a thing, is composed of several approaches and techniques that I have added through the years. Some of them come from direct experience and trial and error, but others come from reading books. I missed not having had the time to go into more details about some of my favorite leadership "tricks". I also missed not having mentioned some of the books where I found out about them and recommending them to that candidate.

This post is what I missed saying in that conversation. I have structured some of what I consider to be the most important habits of a leader around recommended readings. Regardless of whether you trust or agree with my advice, I hope this will be at the very list an interesting reading list recommendation. I should also stress that while the focus of this post is on leadership strategies, I hope anyone can benefit from these, regardless of your current position. In fact, hopefully by reading this post I can encourage you to tap to that "leader in you".

### The 8 habits

So, let's dive right into it: What are the 8 habits of a successful leader?

> 1\\. Be a Multiplier

> 2\\. Promote Extreme Ownership

> 3\\. Practice Radical Candor

> 4\\. Make an Alliance

> 5\\. Overcommunicate Clarity

> 6\. Measure what matters

> 7\. Create purpose and Meaning

> 8\. Treat people like adults

In the rest of this post, we will get into each of these habits and connect them to a book where they are explained in much more details.

![](/blog/images/08-01.png)

### 1. Be a Multiplier

According to Liz Wiseman, the author of [Multipliers](https://thewisemangroup.com/books/multipliers/), *“Multipliers, are the leaders who use their intelligence to amplify the smarts and capabilities of the people around them. When these leaders walk into a room, light bulbs go off over people’s heads; ideas flow and problems get solved.”*

![](/blog/images/08-02.jpeg)

As a leader, you don’t want to be the smartest person in the room, nor to have the answer to every problem. What you do want is to grow everyone in your team so they maximize the impact and performance of the team as a whole. Being a multiplier involves attracting and cultivating talent. It also means creating a culture of debate, challenge, and ownership at all levels. But, most of all, it involves investing in people and believing in their potential to grow.

Being a multiplier is in my point of view perhaps the number one goal of any leader and manager.

### 2. Promote Extreme Ownership

In Extreme Ownership, Jocko Willing, and Leif Babin, two former officers of the Navi SEALs, explain why as a leader it is very important that you feel [extreme ownership](https://echelonfront.com/extreme-ownership/) over your team’s results. If the team wins, you owe it to the team. If the team fails, it is your fault. This is a radical way of thinking about leadership, as the title of the book suggests, but it is the only way to gain ultimate respect as a leader and to get people to also feel similarly about their own work.

![](/blog/images/08-03.jpeg)

I have to say that despite coming from a military family I am not a fan of “military culture”, so I always feel a bit awkward about recommending a book about the Navy Seals. However, this book has a lot of universal lessons that anyone in any position of leadership should know about and possibly implement in their context. Being in an extreme position where the lives of people (including your own team) depend on you makes it more obvious how you should behave, and that adds clarity to the message. But, I truly believe any leader should learn how to check their ego out and give everything for the team.

This book also has very valuable lessons on prioritization, execution, clarity, and decision-making under uncertainty. Again, all of these are very valuable assets for a manager of any high performing team.

### 3. Practice Radical Candor

![](/blog/images/08-04.jpeg)

[Radical Candor](https://www.radicalcandor.com/) is perhaps the perfect complement to Multipliers. If you want to make people better and grow them, challenge them, and care about them, says Kim Scott. I love the simplicity of Kim’s two dimensional space of caring and challenging. In order to be a “kick-ass” boss, you need to combine those two dimensions. If you just challenge without caring, you enter the territory of Obnoxious Aggression (aka jackass), but if you care without challenge you end up on ruinous empathy. This is all illustrated in the simple two-dimensional graph below.

![](/blog/images/08-05.png)

Radical candor is one of my all-time favorite management books, and I was fortunate to meet Kim and learn from her directly. An instant classic with lots of interesting examples from Kim’s and other managers personal experiences that also includes practical advice on aspects such as how to use one-on-ones’s or how to understand what motivates every individual.

### 4. Make an Alliance

I picked up a copy of [The Alliance](https://www.amazon.com/Alliance-Managing-Talent-Networked-Age/dp/1625275773/) from the Netflix cafeteria years ago. When I was reading it at home, I realized it had a personal dedication from Reid Hoffman (founder of LinkedIn) to Reed Hastings (founder of Netflix). I sent Reed an email, but he said I should keep the book: He had left it on purpose for someone to read it after he had finished it. I do feel like I own a piece of Silicon Valley history with that personalized copy.

![](/blog/images/08-06.jpeg)

The Alliance focuses on the interesting question of how to address the tension between what the employee wants, and what the company needs. The answer is, of course, to form an explicit alliance. The individual will give their best, working hard and optimizing for what is best for the company. The company promises to give back not only compensation, but also growth and career opportunities.

To be honest, the book goes perhaps into too much detail of LinkedIn’s approach to implementing the Alliance through “Tours of Duty” (4 years in which the employee commits to a particular “project”) and “employee networks”. Both these ideas are somewhat interesting, but not worth a book. However, the overall idea of an alliance, and many of the practical tips on how to manage it are really useful and important for any manager.

### 5. Overcommunicate Clarity

[The Advantage](https://www.tablegroup.com/books/the-advantage/) is another book that came to me through Netflix’ Reed Hastings since he recommended it to all managers at Netflix back then. The book is a great read on organizational culture. You can think of it as an updated book-sized version of Peter Drucker’s “Culture eats strategy for breakfast”. Patrick Lencioni’s [The Advantage](https://www.tablegroup.com/books/the-advantage/) does a great job in explaining why despite the lack of a unique formula to find the “right culture” for your company, finding that culture and making sure it fits your mission and your business context is maybe the most important thing you should do as a leader. And, while it is true that the specifics of the culture will depend on things like your mission, your product, and your business context, there is an important aspect of any successful business culture: clarity.

![](/blog/images/08-07.jpeg)

Lencioni describes his four disciplines model:

1. Build a cohesive leadership team
2. Create clarity
3. Overcommunicate clarity
4. Reinforce clarity

As you can see, most of the disciplines about creating and reinforcing clarity, which is a key requirement for any successful leader. Clarity leads not only to better alignment and execution, but also to [pshycological safety](https://hbr.org/2017/08/high-performing-teams-need-psychological-safety-heres-how-to-create-it) since people in the organization understand the expectations and the norms, and can therefore feel safe to operate within them.

### 6. Measure what matters

As an engineer/scientist I am a fan of data-informed decisions. However, people and organizations are complex, so making decisions that are backed by things that you can measure is hard. In [Measure What Matters](https://www.amazon.com/Measure-What-Matters-Google-Foundation/dp/0525536221), John Doerr describes the well-known approach to doing this based on [OKRs](https://en.wikipedia.org/wiki/OKR) (Objectives and Key Results).

![](/blog/images/08-08.jpeg)

So, what is the OKR approach in a nutshell? Easy: you describe the goals for your company, teams, and individuals as the combination of **objectives** (what you are trying to accomplish), and associated **key results** (how you will measure whether you have accomplished those objectives). OKRs are ideally propagated up and down the company’s hierarchy all the way to the individual in a team. The key results at each level should become objectives a level down. The book also has many other useful recommendations on how to write key results in a way that they are really measurable and also in a way that they promote stretch goals.

You might decide that the OKR approach is not the right one for your team or company (in fact another of my recommendations below argues against them), however, understanding this method and going through the many interesting examples in this book is a must for any leader. Doerr describes how OKRs have been used at large companies such as Google or Intel, but also how they have been useful in small startups.

Besides describing the OKR approach, Measure What Matters also goes into aspects of leadership in the form of “superpowers”, all of which I think are very important:

1. Focus
2. Align and Connect
3. Track for Accountability
4. Stretch for Amazing

### 7. Create Purpose and Meaning

I first heard about Kofman’s approach to leadership directly from him in a talk. I will be honest to say that I wasn’t very impressed probably because it was a relatively short presentation, and he just brushed over a lot of things at a very high level. His constant references to LinkedIn did not help much in getting my attention either. However, I did decide to read [The Meaning Revolution](https://www.amazon.com/Meaning-Revolution-Power-Transcendent-Leadership/dp/1524760730) not long ago after it was recommended to me at work, and I did end up thinking it is a very good leadership book, totally worth a read, particularly because it offers a complementary view to most of the books I listed above.

![](/blog/images/08-09.jpeg)

The main thesis of the book is that in order to become a great leader, you need to transcend and motivate employees by offering them meaning and the opportunity to be part of something “bigger”. This really aligns with my personal approach, and my career choices have always led me to find companies with more and more “meaning”. I will say that the one thing I disagree with Kofman is his idea that you can offer transcendence and meaning regardless of what your company does (he convincingly uses the example of a large insurance company).

That being said, I really liked most of the other “side” lessons and pieces of advice in the book. Kofman has very useful advice on how to think about company culture, and how to push for “response-ability” while also encouraging collaboration. Despite of what you think about the main thesis of the book or how many holes you can poke at it, I strongly believe that this central part of the book has very good and useful advice for any leader or leader-to-be.

### 8. Treat People Like Adults

In [Powerful](https://www.amazon.com/Powerful-Building-Culture-Freedom-Responsibility/dp/1939714095), Patty McCord describes the most important aspects of the Netflix culture. As unique as it is, the Netflix culture has had a huge impact specially in the tech and startup world. And, of course, it has had a tremendous impact on me. I will always consider myself a Netflix alumni. Yes, I [have written](https://towardsdatascience.com/cultural-overfitting-and-underfitting-or-why-the-netflix-culture-wont-work-in-your-company-af2a62e41288) about how this is not a culture that you can copy (as none other is), but there are many aspects of the Netflix culture I love. Patty, who was Chief Talent Officer during my time at Netflix, was the main architect of that culture, and she has a lot of valuable lessons.

![](/blog/images/08-10.jpeg)

The book touches upon many different aspects of the Netflix culture, all of which can be summarized by the subtitle of “Freedom and Responsibility”, but also by the First Chapter’s subtitle: “Treat People Like Adults”. This might seem like an obvious thing to do, but when you are reminded of how most tech companies treat their employees you realize it is not. McCord argues against perks, structured career ladders, and compensation reviews. She does propose a culture in which employees are valued for their impact, and their contribution to the company’s success. Interestingly, there are many aspects of this that relate to both “The Alliance” and Kofman’s “Meaning and Purpose”.

The book also talks a lot about communication aspects such as radical honesty and a culture of debate. Finally, it goes into practical aspects such as how to set up a compensation “policy”, and how to hire and fire. All of these are important aspects of leadership that I would recommend anyone in such a position to consider and at the very least have an informed opinion on.

### A final recommendation

Many of you might have recognized that the title of this post honors one of the most well-known and respected business books ever written: [The 7 Habits of Highly Successful People](https://www.amazon.com/Habits-Highly-Effective-People-Powerful/dp/0743269519) by Stephen Covey. I had heard about this book many times, but I was honestly put off by a title that seemed like a marketing gimmick, and the fact that it was usually described as a “self-help book”. Was I wrong. This book has extremely important and deep lessons useful not only for business, but for life in general (I guess that is why many people will call it “self-help”). The reason I finally decided to pick up the book is because I saw it in the recommendation list of someone I trust. Maybe you are reading this and wondering whether it’s worth it. I hope to convince you it is.

![](/blog/images/08-11.jpeg)

I won’t go into any details, but I think it is worth to list Covey’s 7 habits:

1. Be Proactive
2. Begin with End in Mind
3. Put First Things First
4. Think Win/Win
5. Seek First to Understand, then to Be Understood
6. Synergize
7. Sharpen the Saw

Note how many of these habits touch upon several of the themes of my 8 habits above such as ownership, communication, and goal setting or prioritization. Covey does add an important and missing dimension in the 7th habit “Sharpen the Saw”, where he describes how any personal (or organizational) growth needs to be interpreted as a continuous and incremental process. I am a fan of incremental improvement, and I am really happy we added the great value of “Incremental Excellence” to [our values at Curai](https://curai.com/work).

Whatever your current position and style or approach is, one thing I can tell you for sure: you are not perfect, and you can get better. This idea of continuous growth and improvement is one of the most awesome things of being a human being. Embrace and enjoy it!

```
---
id: 28
title: "Scaling healthcare through AI"
date: '2020-12-09T00:00:00+00:00'
author: xamat
##layout: post
permalink: /scaling-healthcare-through-ai-3dc526c24ba9/
header:
     teaser: https://amatriain.net/blog/images/07-01.png
reading_time:
    - ''
    - ''
categories:
    - Uncategorized
---
```

[Curai](https://www.curaihealth.com/)’s mission is to “provide the world’s best healthcare to everyone”. In order to do so, we strive to scale the reach of an individual physician by creating tools that automate large portions of a medical interaction under their supervision. While our end-goal is to provide full end-to-end medical care to everyone, we initially focus on **primary care**.

Primary care is an essential component of improving healthcare access and outcomes, and its benefits have been shown in many studies. For example, in his [2012 meta study](https://pubmed.ncbi.nlm.nih.gov/24278694/), Leiyu Shi reports that “*In both developed and developing countries, primary care has been demonstrated to be associated with enhanced access to healthcare services, better health outcomes, and a decrease in hospitalization and use of emergency department visits. Primary care can also help counteract the negative impact of poor economic conditions on health*.” Primary care should be the first and most common touchpoint between patients and the medical system. Primary care is also broad in coverage. This provides a great starting point for scaling through automation.

Recent advances in AI also provide an optimistic outlook for making this mission a reality. We at Curai are convinced that AI and ML will enable significantly better access to and higher quality care. In order to get there, we believe that physicians need technology that helps them operate more efficiently and at a larger scale while also helping them improve quality of care. How do we get started on that?

Our approach to scaling and augmenting physicians is based on the following premises:

1. If you want to automate anything, first do it manually.
2. Automation should not decrease quality. In fact, it should aspire to improve it.
3. The first approach to automating any process should be as simple as possible, yet designed in a flexible way.
4. Automation should improve over time.
5. Humans in the loop provide quality assurance that helps with (2), and also valuable data that helps with (4).

To accomplish our goals while following the previous premises, we are using a combination of:

1. State-of-the-art AI in the wild
2. AI with experts-in-the-loop

In the rest of this post, we will describe these two dimensions and the techniques that support them.

### State-of-the-art AI in the wild

In order to make a real impact in the world in general and healthcare in particular, AI needs to accomplish two important goals. First, it needs to deliver performance comparable if not better than experts on well understood datasets. But, just as importantly, it needs to be able to operate “in the wild’’ in conditions different from the lab where there might be noise and incomplete information. Our goal is to tackle both these requirements in parallel.

Our models are best-in-class when compared to known competitors. As an example, see our diagnosis accuracy when compared to many online symptom checkers on the benchmark Semigran dataset in the graph below (see “[Evaluation of symptom checkers for self diagnosis and triage: audit study](https://www.bmj.com/content/351/bmj.h3480)”. Semigran et al. 2015). Note how our in-house algorithm beats all of the others in the benchmark significantly both in top-3 and top-1 accuracy. Comparing to other currently existing diagnosis algorithms is hard since most companies are not transparent about the accuracy of their models, and present them only on partially available or fully proprietary datasets. As an example, [Raazaki et al](https://arxiv.org/pdf/1806.10698.pdf). self report on a manually selected subset of Semigran. When extrapolated to the full dataset, this would yield a top-1 accuracy of about 0.46 and a top-3 of 0.65.

![](/blog/images/07-01.png)

The accuracy of our algorithms improves over time, and some of them are close to the performance of average physicians. See improvement over time and comparison to doctors in the graph below. (Note that the average physician accuracy is derived from “[Comparison of Physician and Computer Diagnostic Accuracy](https://jamanetwork.com/journals/jamainternalmedicine/fullarticle/2565684)” \[Semigran et al. 2017\] on a small dataset of 234 physicians that decided to diagnose at least one case, as opposed to our AI algorithm that must respond to all cases. Even with that caveat, our Dx algorithm is already surpassing expert physicians on top-3 accuracy). Very importantly, we also include the accuracy of our COVID-aware diagnosis algorithm on pre-existing cases. Even in this situation, our algorithm is delivering close to human doctor accuracies while having quickly adapted to a previously unknown condition (see comparison of with/without COVID from Nov 2020 accuracies). This notion of ever-learning algorithms in the wild is extremely important and contrasts with the approach taken by most existing diagnosis apps for which accuracy not only doesn’t improve, but is likely to get worse over time (see [very recent publication at the Journal of Medical Internet Research](https://www.jmir.org/2020/12/e18097)). Speaking of this very recent publication hot off the press, we just ran the published vignettes through our system and scored 2.5, which is in line with the best system in the study, and well over our competitors.

![](/blog/images/07-02.png)

While Diagnosis (Dx) is definitely a core component of medical practice, there are many other aspects that we are trying to automate with our AI approaches. We focus on the following three medical processes:

1. Information gathering
2. Assessment/Diagnosis
3. Planning

At Curai, we are approaching these problems with a combination of state-of-the art Deep Learning and traditional AI approaches such as expert systems and knowledge bases. Here are some concrete examples of how we are approaching the problems:

- For the Information gathering step we are using a combination of expert systems and NLP in our History Taking algorithm, which is able to surface the optimal medical question to ask given the current differential diagnosis and the findings gathered from the patient so far. These questions are sometimes directly sent to the patient or suggested to the providers as a way to improve their efficacy while also providing feedback to the model.
- For Triaging, Diagnosis, and treatment recommendation, we are using a combination of different approaches. These approaches vary from Deep Learning models learned from EHR to expert systems built in collaboration with top medical institutions in the country.

We have published several [peer-reviewed publications](https://medium.com/curai-tech/research-publications-at-curai-ee22a820b807) on our approaches, and have an assigned patent.

While having public benchmarks and publishable results is important so as to provide some form of comparison with existing approaches, the reality is that diagnosis is not performed in a noise-free environment with perfect information. That is why we also focus on understanding the performance of our algorithms “in the wild”. Once AI is deployed, it needs to observe two important properties:

1. Algorithms need to be able to model “the unknown”
2. Algorithms need to not only improve over time, but be able to increase their scope and to adapt to new data

We have tackled these two issues since the get go and have designed approaches to modeling “out of band” knowledge and algorithms that can quickly adapt to new/unseen data. For example, in “[Open Set Diagnosis](https://arxiv.org/abs/1910.02830)”, we show how different out-of-band modeling approaches perform in situations where we have a set of known and unknown conditions in our population. In “[Learning from the Experts](https://arxiv.org/abs/1804.08033)”, we show how static expert systems can adapt to new conditions by training a deep neural network on a combination of synthetic and natural data. In a [recent article](https://arxiv.org/abs/2008.03323), we also show how algorithms can quickly adapt to a rapidly changing situation such as the COVID-19 pandemic.

### AI with experts in the loop

Healthcare is made up of many different processes, most of which are medical in nature and therefore require a high level of domain expertise. While those processes have varying degrees of complexity, they can be effectively simplified to an information exchange between a patient and a healthcare professional (i.e. expert) plus access to internal data such as Electronic Health Records (EHR) and external data such as publications, standards, or even lab results.

When trying to automate any of such processes, we always think about keeping physicians in the loop. Just as physicians benefit from the augmentation that AI provides, AI also benefits from keeping experts in the loop, and learning from them. Experts not only provide a safety net for when the AI fails, but also are an essential part of the feedback loop. How is this “expert in the loop” model implemented in practice? In our context, each of the medical processes can be understood as the interaction between patients and our own healthcare professionals through our platform and internal/external data.

![](/blog/images/07-03.png)

Both patients and physicians interact with the Curai platform to communicate. Physicians also interact with the internal data (mostly our internal Electronic Health Record or EHR, but also our KB, a curated knowledge base that represents medical concepts and the relations between them) and all kinds of external data such as publications, web pages, medical standards, or protocols. For example, when gathering information during the encounter the physician asks the patient questions while interacting with the Curai system. The physician also enters information in the EHR depending on the patient's answers. And, of course, they solicit information based on their knowledge of medical literature and best practices. Any of those processes can be automated.

In a fully automated system, we would only have interaction between the patient and the Curai system. In a fully automated process, the information gathering step consists of an AI that asks relevant questions to the user based on existing information in the EHR and external knowledge (e.g. protocols). The automated process also fills in relevant information into the EHR. While we do aspire to drive full automation in many parts of the medical encounter, many of our current efforts are currently focusing on driving "partial automation" by keeping experts in the loop.

![](/blog/images/07-04.png)

Many of our research efforts focus on automating the interaction between the physician and the system. By automating that interface, we make physicians much more efficient, but also improve quality of care. Examples of projects like these include suggesting questions or responses to physicians, automated charting for improved EHRs, or content/literature recommendation for easier access to external data. Other automation efforts however focus on the interaction between the patient and the system. These include efforts like extracting information from the user reported "reason for encounter", content recommendation for the user, or automatic routing of the user to the right level of care depending on their input.

All the ML models we use at Curai are getting validated and retrained thanks to the feedback and the usage we get from our healthcare professionals and our patients. As an example, when we suggest a response to clinical associates and physicians, we record their acceptance, rejection, or modifications. We use that information to improve our models. Our own proprietary data, generated by thousands of conversations between patients and physicians, provides a unique asset from which we can learn medical knowledge. See for example below embeddings learned exclusively from conversations in our product. We have also been able to learn accurate diagnosis models using exclusively conversational data from our users and doctors.

![](/blog/images/07-05.png)

### Where are we now, and the road ahead

As of today, we are able to automate about 30% of the medical conversation (defined as the percentage of medical findings that make it to the EHR and are automatically generated by the system). However, the improvement we are seeing while integrating automation features both on the Patient and Provider Clients (see below) gives us confidence that we will be able to get that number close to 80–90% in the near future. In order to do so, we need to expand our current product features to include other modalities and inputs as well as tackle other processes in the medical workflow. Our [research efforts](https://medium.com/curai-tech/research-at-curai-fa09cae6727a) are already underway and we hope to soon include many of those results in the product itself.

![](/blog/images/07-06.png)

The road towards full automation includes two phases: (1) Automate only high incidence conditions, (2) Broaden the scope of the automation to cover 80–90% of primary care. Focusing on high-incidence conditions (e.g. common cold or UTI) makes sense for several reasons. For one, we want to focus first on automating those conditions that will have more impact on our ability to scale. Also, those conditions are the ones we get more data on, and therefore where the learning loop is able to quickly improve over the baseline. Note though that this automation is designed in a generalizable way so that we can expand the automation coverage easily over time.

Automation, understood as a way to augment and scale physicians, is a fundamental hypothesis of our approach to radically improving healthcare access and quality. We believe in a world where people will have access to high quality medical advice and care 24/7 on their phones, and we at Curai are determined to make this happen soon.

```
---
id: 10
title: 'On Decoupling'
date: '2021-01-31T00:00:00+00:00'
author: xamat
##layout: post
permalink: /on-decoupling-af1ecc539a98/
header:
    teaser: https://amatriain.net/blog/images/06-01.png
reading_time:
    - ''
    - ''
categories:
    - Uncategorized
---
```

You have probably heard of the [butterfly effect](https://en.wikipedia.org/wiki/Butterfly_effect): a distant butterfly flapping its wings can have a hard-to-predict effect on a future event. Everything seems to be connected in the world, and tech companies, let alone startups, are microcosms where that seems to be even more the case.

Everything is connected

![](/blog/images/06-01.png)

Startups make a big effort to have people focused and aligned (i.e. on the same boat). A small team means that people need to understand “everything” and be flexible to do whatever is needed. As a result of this, everything not only seems connected, but it actually is, by design. This leads to a situation where everything, from systems to processes, are completely coupled. Furthermore, coupling is valued as a positive, not as a flaw.

The benefits of coupling

Before we talk about the downsides and alternatives, it is important to acknowledge that coupling has its merits. Having everyone and everything on the same boat, if possible, means that you will be able to move faster, particularly when changing direction.

![](/blog/images/06-02.png)

The extreme of coupling is designing everything as if it belonged to the same system. In software engineering we usually refer to this pattern as the [monolith] (https://en.wikipedia.org/wiki/Monolithic_application). There are different variations of the monolith, but in general you can think about designing your system as a single tier application where all code and services are interconnected and live in the same place (repository, server, etc…).

While monoliths have generally a bad press, there are again benefits with an approach where everything is in the same place and tightly coupled. It is true that nowadays people preaching the benefits of monoliths are seen as contrarians, but I would recommend you read some of what they have to say (see e.g. [In Defence of the Monolith](https://www.infoq.com/articles/monolith-defense-part-1/) or [Monoliths are the future] (https://changelog.com/posts/monoliths-are-the-future?_lrsc=8f40a77e-9ec3-46cd-97fe-8a8d965c6685)). You will generally read about benefits like maintainability, simplicity, or efficiency. Probably the main benefit is that keeping everything connected and centralized makes it easier to reason about it (e.g. test it). It is also interesting to keep in mind that very successful and large-scale applications such as [Facebook] (https://softwareengineeringdaily.com/2019/07/15/facebook-php-with-keith-adams/), or Quora where I led engineering, were a monolith until very late in their growth curve.

Regardless of what your philosophical stance is on the issue, it is hard to argue that no matter what your end game will be, it is usually more practical to start with a monolithic design to then evolve it into something maybe different. See what software architecture guru Martin Fowler has to say about this in his [MonolithFirst](https://martinfowler.com/bliki/MonolithFirst.html) pattern discussion.

But there comes a time

Whatever you think about the pros/cons tradeoffs in favor of coupling, there is an undeniable downside of a single monolithic and coupled system: there comes a time when it does not scale. The system has become too large and complex for any one person to understand it fully. Touching one piece of a tightly coupled system you don’t fully understand is a recipe for disaster since you cannot predict how it will affect the system as a whole.

Note that what I am saying here does not apply only to a software system, but to any organizational design. For example, in a tightly coupled team everyone is expected to know the details of what everyone else is working on since there will be dependencies between everyone’s work, priorities and timelines. It should be obvious to see that this can work only up to a certain size of a team.

![](/blog/images/06-03.png)

Everyone on the same fast and nimble boat</figcaption></figure>So, when that time comes, and it will, it should be clear that the benefits of a loosely coupled system with fewer dependencies will outweigh the ones of a tightly aligned system. When that happens, it is important to not over-react. I hear too many people that react to the monolith by going to the other extreme of [microservices] (https://en.wikipedia.org/wiki/Microservices). Don’t get me wrong, microservice architectures are great, and it might be just what you need. But, there are also many things in between a monolith and microservices. I, for one, am a fan of starting separating services (or teams) one at a time until the need to go fully decoupled and into microservices becomes clear.

At this point it is interesting to see what others have said (and more importantly done) in the past. When doing so, it is also very important to look at those who have been successful.

On the shoulder of decoupling giants

Many highly successful companies have made of decoupling an important aspect of their culture. I will start with one close to home: Netflix.

At Netflix, one of our important principles was: **Highly Aligned, Loosely Coupled**. You can see below the slide from the famous [Netflix Culture Deck](https://www.slideshare.net/reed2001/culture-1798664/94-Highly_Aligned_Loosely_Coupled_Highly). This is a great principle since it not only touches on the importance of low coupling, but it also introduces the importance of doing so while remaining aligned. Loosely coupled teams have high level of autonomy and can move much faster independently. They can eliminate roadblocks themselves and their success does not depend on anyone else. Of course, the risk of this low coupling is that every teams moves in a completely different direction. That is why high alignment on the overall direction is so important.

![](/blog/images/06-04.png)

Everyone on the same fast and nimble boat</figcaption></figure>Maybe the most well known example of a company that is strong on the idea of decoupling is Amazon. The company culture in that respect is well represented in Bezos' API mandate/manifesto from circa 2002 (see below). In this mandate, Bezos required that all teams removed all interdependencies to the point that they could only communicate through service interfaces. According to many, this is one of the secrets to Amazon's success in scaling so much and so quickly. At the very least it is a key aspect of AWS', and overall cloud computing, success for sure.

![](/blog/images/06-05.png)

Somewhat related to the "Monolith first" pattern mentioned above, Reid Hoffman's [Blitzscaling](https://www.amazon.com/Blitzscaling-Lightning-Fast-Building-Massively-Companies/dp/1524761419) describes how to start a company as a small "pirate boat" initially, but be ready to (blitz)scale at some point. When that happens, there are eight key transitions the company needs to go through:

1. **Small to large teams**
2. **Generalists to specialists**
3. Contributors to managers
4. **Dialogue to broadcasting**
5. Inspiration to data
6. **Single focus to multithreading**
7. **Pirate to navy**
8. Founder to leader

![](/blog/images/06-06.png)

I have bolded the ones that are directly related to decoupling. Some of them are very obvious (e.g. single focus to multithreading), but I think there is one that is worth highlighting: from generalists to specialists. Indeed, a single generalist doing everything is the ultimate monolithic organization. Hiring only generalists assumes that "understanding everything" and being flexible is the most important aspect. However, when an organization starts embracing decoupling, it will understand that someone can be very impactful and successful while specializing on a particular aspect of what is needed.

The one notable exception

I will bring it up before you do: sure there are exceptions to decoupled organizations, perhaps the most notable being Apple. It is important to understand though that while those companies might not make decoupling a top-level value, and in fact might praise the virtues of centralization, in order to function at their scale they need to have some form of decoupling. Nobody at Apple, not even Tim Cook, understands everything that is going on at the company. Besides, more centralized organizations like Apple might be reasonable for developing artifacts like devices, but not for products or services (read more about this from Ben Thompson [here](https://stratechery.com/2016/apples-organizational-crossroads/)).

![](/blog/images/06-07.png)

So what can you do?

What should be your takeaway after reading all this? Well, that will depend on your concrete context and situation. As mentioned throughout this post, the exact tradeoff between loose and tight coupling will depend on your particular stage and situation. However, a few things you should always keep in mind:

- If you are starting anything out, start with a monolith with the idea that you will decouple aspects over time.
- Look out for excessive complexity or things blocking each other. This might be a sign of too much tight coupling
- Explicitly call out dependencies between systems, project, or teams at any stage.
- Do not over-react to excessive decoupling by trying to decouple everything. Find the right timing for decoupling different aspects of your "monolith".
- Try to measure the benefits of coupling/decoupling each aspect or element in your system
- As you decouple anything, make sure that you continue to prioritize high-level alignment.

Final thoughts

As I explained in “[Cultural over/under-fitting and transfer learning. Or why the “Netflix Culture” won’t work in your company.](https://towardsdatascience.com/cultural-overfitting-and-underfitting-or-why-the-netflix-culture-wont-work-in-your-company-af2a62e41288)”, the ideal company culture is very context-dependent. In fact, the culture will need to evolve and adapt to each of the stages of the company growth Hoffman explains in Blitzscaling. Coupling/decoupling may be just one more of those cultural aspects, but it is an important one that can determine the degree of success. Paying attention to it is likely to pay dividends in the future.

```
---
id: 18
title: "The AI/ML FDA Plan"
date: '2021-02-11T00:00:00+00:00'
author: xamat
##layout: post
permalink: /the-ai-ml-fda-plan-efb384c9bf31/
header:
    teaser: https://amatriain.net/blog/images/05-01.png
reading_time:
    - ''
    - ''
categories:
    - Uncategorized
---
```

In case you are not familiar with it, the [Food and Drugs Administration](https://en.wikipedia.org/wiki/Food_and_Drug_Administration) (FDA) is a federal agency of the Department of Health and Human Services. that is responsible for the control and supervision of food safety, drugs, vaccines, medical devices, and the likes. In recent times the FDA has been thinking about how to regulate medical devices that are driven or based on AI as a result of several recent efforts, they published their *“*[*Artificial Intelligence/Machine Learning (AI/ML)-Based Software as a Medical Device (SaMD) Action Plan*](https://www.fda.gov/media/145022/download)*”* in January 2021. To understand this action plan, you need to go and read a number of related documents. I did that for you, and my hope is that this post summarizes the most important aspects of the FDA’s plan around Artificial Intelligence and Machine Learning. Make sure to read the last section for a surprise “ending” to it all.

I will start by saying that if you, like myself, come from a technical background, you might be coming into this thinking that the FDA is likely an old bureaucratic institution filled with old-school lawyers that know nothing about novel AI/ML approaches. I can neither confirm or deny that, but what I will say is that this particular action plan has very interesting aspects that I am sure you will agree with if you are working in applied AI in general, and AI for healthcare in particular. As a teaser, here is what the FDA’s vision is, in their own words:

> *“The ability for AI/ML software to learn from real-world feedback (training) and improve its performance (adaptation) makes these technologies uniquely situated among SaMD and a rapidly expanding area of research and development. Our vision is that with appropriately tailored regulatory oversight, AI/ML-based SaMD will deliver safe and effective software functionality that improves the quality of care that patients receive. “*

You probably have noted by now that this action plan is focusing in the so-called SaMD (Software as a medical device). SaMD is a piece of software that on its own performs a medical function. This is in contrast to SiMD (Software **in** a medical device) where the software needs to be embedded in a piece of hardware to perform its medical function.

The Action Plan that was recently presented is a continuation of the so-called *“*[*Proposed Regulatory Framework for Modifications to Artificial Intelligence/Machine Learning (AI/ML)-Based Software as a Medical Device (SaMD)*](https://www.fda.gov/media/122535/download)*”* published in April 2019. The proposed framework included a number of questions for feedback from stakeholders that the current action plan takes into account and responds to by proposing a follow up plan in 5 different directions. So, in order to understand the current Action Plan, we need to first understand the proposed Framework.

### **The “Proposed Framework”?**

Traditionally, the FDA reviews medical devices through an appropriate premarket pathway, such as [premarket clearance (510(k)),](https://www.fda.gov/medical-devices/premarket-submissions/premarket-notification-510k "Premarket Notification 510(k)") [De Novo classification](https://www.fda.gov/medical-devices/premarket-submissions/de-novo-classification-request "De Novo Classification Request"), or [premarket approval](https://www.fda.gov/medical-devices/premarket-submissions/premarket-approval-pma "Premarket Approval (PMA)"). Generally speaking, these premarket approvals need to happen in 3 situations:

- A change that introduces a new risk or modifies an existing risk that could result in significant harm
- A change to risk controls to prevent significant harm
- A change that significantly affects clinical functionality or performance specifications of the device

However, this cannot be used in an AI/ML case where the “device” is expected to continuously improve as it gets more data and models are improved. The proposed framework addresses this particular concern as well as other requirements of AI-based medical devices.

In order to get there, it first defines a risk categorization framework based on: (1) the significance of information provided by the SaMD to the healthcare decision, and (2) the state of healthcare situation or condition, which identifies the intended user, disease or condition, and the population for the SaMD. See table below:

![](/blog/images/05-01.png)

The framework also defines 3 types of not mutually exclusive AI/ML-based SaMD modifications:

1. **Performance** (e.g. retraining, change in AI architecture…)
2. **Inputs used by the algorithm** (e.g. adding new input data types and sources)
3. **Intended use** (e.g. from a confidence score that is ‘an aid in diagnosis’ (drive clinical management) to a ‘definitive diagnosis’ (diagnose) or inclusion of pediatric population where the SaMD was initially intended for adults ages 18 years or older)

Finally, and very importantly, it describes a Total Product Lifecycle (**TPLC**) Regulatory Approach for AI/ML-Based SaMD illustrated below. The TPLC has 4 different regulatory components (i.e. touchpoints with the FDA): (1) A set of Good Machine Learning Practices, (2) a Premarket Assurance, (3) a Change Protocol Review Process, and (4) ways to define Real-World Performance. Let’s look briefly into these 4 components.

![](/blog/images/05-02.png)

1. **Quality Systems and Good Machine Learning Practices (GMLP)**

All SaMD, devices that rely on AI/ML are expected to demonstrate analytical and clinical validation, as described in the SaMD: Clinical Evaluation guidance (see figure below). Example considerations include: Relevance of available data to the clinical problem and current clinical practice; data acquired in a consistent, clinically relevant and generalizable manner; appropriate separation between training, tuning, and test datasets; and appropriate level of transparency (clarity) of the output and the algorithm aimed at users.

![](/blog/images/05-03.png)

2. Initial Premarket Assurance of Safety and Effectiveness

SaMD will require a “**Predetermined change plan**” that should include:

- **SaMD Pre-Specifications (SPS)**: manufacturer’s anticipated modifications to “performance”, “inputs,” or “intended use”
- **Algorithm Change Protocol (ACP)**: Specific methods in place to achieve and control the risks of the anticipated modifications in the SPS. See table below for more details on what is expected in the ACP.

![](/blog/images/05-04.png)

3. Approach for modifications after initial review with established SPS &amp; ACP

Besides the predetermined change plan, the SaMD also needs to specify an approach for modifications. The requirements depend on the concrete situation and whether the SPS and ACP above had been approved. See workflow below.

![](/blog/images/05-05.png)

4. Transparency &amp; real-world performance monitoring of AI/ML-based SaMD

Finally, the regulatory framework requires AI-based medical devices to address transparency and real-world performance. There are not many details of how that should be implemented in the framework, but mostly some examples on how to ensure transparency and monitoring. E.g.1. Transparency may include updates to FDA, device companies and collaborators of the manufacturer, and the public, such as clinicians, patients, and general users. E.g. 2. Real-world performance monitoring may also be achieved in a variety of suggested mechanisms that are currently employed or under pilot at FDA, such as adding to file or an annual report

Each of the sections above comes with a number of questions to the “stakeholders”. Besides, the document includes a number of real-world examples and scenarios including:

- Intensive Care Unit (ICU) SaMD
- Skin Lesion Mobile Medical App (MMA)
- X-Ray Feeding Tube Misplacement SaMD

### **Feedback**

After its publication, and for the following few months, the framework above received [133 public comments](https://www.regulations.gov/docketBrowser?rpp=25&so=DESC&sb=commentDueDate&po=0&dct=PS&D=FDA-2019-N-1185). Those came from large corporations such as GE Healthcare or Anthem, but also individuals. There are even anonymous comments. All of them are available in the public site.

![](/blog/images/05-06.png)

### **Pilot: AI-guided Cardiac Ultrasound**

As a way to test the framework in the real-world, the FDA applied it to [a pilot](https://www.fda.gov/news-events/press-announcements/fda-authorizes-marketing-first-cardiac-ultrasound-software-uses-artificial-intelligence-guide-user) for regulating a device developed by a startup called [Caption Health](https://captionhealth.com/). This device uses AI to guide ultrasounds, and to enable untrained registered nurses to perform ultrasounds at the same level of quality as trained specialists

![](/blog/images/05-07.png)

### **Digital Health Center of Excellence**

Also in the context of these AI/ML initiatives, the FDA started the [Digital Health Center of Excellence](https://www.fda.gov/medical-devices/digital-health-center-excellence) late 2020. The center includes many external collaborators from universities and other non-profits (not for profit companies as far as I can see). Its 3 listed objectives are:

- Connect and build partnerships
- Share knowledge
- Innovate regulatory approaches

### The Action Plan

Ok, so we are now that we understand the original framework, and the process, we are ready to go back to the January '21 action plan. The plan includes a summary of the feedback received ("What they heard"), and a plan on what the next actions will be ("What they'll do"), all of it grouped in 5 different areas:

1. **Tailored Regulatory Framework for AI/ML-based SaMD**

- **What they heard:** Many suggestions including re:Predetermined Change Control Plan
- **What they'll do:** Update the proposed framework, including issuance of Draft Guidance on the Predetermined Change Control Plan for public comment

2. Good Machine Learning Practices (GMLP)

- **What they heard:** Strong general support for the GMLP + a call for FDA to encourage harmonization through consensus standards efforts.
- **What they'll do:** Encourage harmonization of Good Machine Learning Practice development through institutions such as IEEE and AAMI

3. Patient-Centered Approach Incorporating Transparency to Users

- **What they heard:** Call for further discussion on how AI/ML-based technologies interact with people, including their transparency to users and to patients
- **What they'll do:** Following up on recent Patient Engagement Advisory Committee meeting, next step: hold public workshop on how to support transparency to users and enhance trust in AI/ML-based devices

4. Regulatory Science Methods Related to Algorithm Bias &amp; Robustness

- **What they heard:** : Stakeholders described the need for improved methods to evaluate and address algorithmic bias and to promote algorithm robustness.
- **What they'll do:** Support regulatory science efforts to develop methodology for the evaluation and improvement of algorithms, including for the identification and elimination of bias, and promotion of algorithm robustness (e.g. Centers for Excellence in Regulatory Science and Innovation (CERSIs) at UCSF, Stanford, and Johns Hopkins University)

5. Real-World Performance (RWP)

- **What they heard:** Stakeholders described the need for clarity on Real-World Performance (RWP) monitoring for AI/ML software
- **What they'll do:** Work with stakeholders who are piloting the RWP process for AI/ML-based SaMD

To summarize, the action plan proposes to:

- Update the proposed regulatory framework in the AI/ML-based SaMD discussion paper, including issuance of a Draft Guidance on the Predetermined Change Control Plan.
- Harmonize development of GMLP through additional FDA participation in collaborative communities &amp; standards development efforts.
- Continue to host discussions on the role of transparency to users, including holding a public workshop on medical device labeling to support transparency
- Support regulatory efforts on methodology for evaluation &amp; improvement of algorithms, including for the identification &amp; elimination of bias, &amp; robustness and resilience of algorithms.
- Advance real-world performance pilots in coordination with stakeholders and other FDA programs, to provide additional clarity.

### A last-minute hurdle?

If you think everything you read until now makes sense, you will be surprised to read what happened next. In the middle of the chaos of Trump’s administration last days, the HHS published a request to exempt a large number AI devices from FDA regulation. This piece of news was pretty shocking for the community following these efforts and it is still unclear what it means, particularly since the new Biden administration will need to decide whether to approve or not the requested exemption (read more [here](https://www.statnews.com/2021/01/16/slippery-slope-territory-health-officials-propose-waiving-regulatory-review-of-medical-ai-tools/#:~:text=Days%20before%20leaving%20office,respiratory%20disease%20on%20medical%20images.), paywalled, or [here](https://medcitynews.com/2021/01/in-the-final-days-of-trump-administration-agencies-clashed-over-how-to-regulate-medical-ai/), free)

![](/blog/images/05-08.png)

In any case, these next few months promise to be really important and decisive for the whole space of AI in healthcare regulation. I look forward for the outcome of it all to provide a good avenue for aggressive innovation plus patient safety and medical quality.

```
---
id: 27
title: 'Valuing Equity in Startups: A Decision-Making Framework (and a spreadsheet)'
date: '2021-03-21T00:00:00+00:00'
author: xamat
##layout: post
permalink: /valuing-equity-in-startups-a-decision-making-framework-7cfc39d99597/
header:
    teaser: https://amatriain.net/blog/images/04-01.jpeg
reading_time:
    - ''
    - ''
categories:
    - Uncategorized
---
```

![](/blog/images/04-01.jpeg)

(In case you came here looking for the spreadsheet, [here](https://docs.google.com/spreadsheets/d/1uqETG2kl2mnbqh7aQvJJIu7DeAT0uLOwuwxrki0vgs4/edit?usp=sharing) it is. However, I recommend you continue reading to help you understand the overall framework.)

I am often surprised that many folks I talk to, even those working at startups, have very little understanding of how to think about the value of their equity. I should credit [Chip Huyen](https://huyenchip.com/) for [her post](https://huyenchip.com/2021/02/27/why-not-join-a-startup.html) about working on a startup. While I do feel in order to make justice to that post I need to write a much broader write-up about working in a startup, the current one is addressing the discussion of how to value equity in the “[You won’t get rich](https://huyenchip.com/2021/02/27/why-not-join-a-startup.html#4_equity)” section. If you are joining a startup you should probably be doing it for many other reasons than to “make money”. However, I do think it is important to have a framework to evaluate the economic value (or risk) of joining one.

What you will read here is not the result of [survivorship bias](https://en.wikipedia.org/wiki/Survivorship_bias) of someone who’s made it big in a startup: I have yet to have a big exit from a startup (both the two startups I have been at Quora and [Curai](https://www.curai.com) have not had an exit yet). I wish I had known more of this stuff during my time at Netflix though. I would have made much more money from my options. I will admit though that Iam a founder, and I might be biased. So, just do your research and make your personal choices. I just hope that the information in this post helps you build a better model of your prospects in a startup.

### Risk, probabilities, and expected value

In order to value equity in a startup it is useful for you to become acquainted with the notion of [expected utility](https://www.investopedia.com/terms/e/expectedutility.asp) and [expected value](https://en.wikipedia.org/wiki/Expected_value). Expected utility is a economical theory and framework to make decisions under uncertainty. While there is some limitations to equating expected utility and value, this can be used as a good approximation (in reality [most limitations](https://www.investopedia.com/terms/e/expectedutility.asp) stem from the fact that risk and probabilities are subjective, which we will talk about in the post). The expected value of an “event” is simply the value you should expect giving the possible outcomes and the probabilities for each of the outcomes.

![](/blog/images/04-02.png)

Strictly speaking, a startup has an infinite number of possible outcomes, but you can generally simplify to a few. Just keep in mind that the probabilities need to add up to 1 and that you need to add the probabilities in bins (i.e. the probability of the outcome being 10 is going to be in reality the probability of the outcome being 0&gt;x≤10).

Let’s compute your yearly compensation expected value using [Chip’s example](https://huyenchip.com/2021/02/27/why-not-join-a-startup.html) (Note that in order to be as fair as possible I didn’t take the max or the min, but rather the average for each of the possible outcomes in the column on the right):

![](/blog/images/04-03.png)

E = 0.8 x 120,000+ 0.05 x (270,000-170,000)/2 + 0.005 x (2,620,000-620,000)/2 + 0.0005 x (25,000,000-5,0000,000)/2 + 0.1445 x 2.12 = **$284k**

So… if we accepted those probabilities in the post to be correct, you should expect your average outcome to be around $300k a year. So, were you to assume these probabilities, and **if you only cared about the actual money** made, you should compare the offer you get from this hypothetical startup to an established company that offered a stable salary at around or probably above **$300k**. The way to think about this is the following: I am getting a compensation that is minimum $120k a year, if I am really lucky I can get up to $25M a year, but on average I should expect to get close to $300k (again, if we accept those probabilities to be correct).

It should be clear by now that a really important thing here is how you determine those probabilities. I, for example, think the probabilities in the example Chip gives above are way off in her case. Is a probability of a startup failing really 80%? Well… it depends. While it is true that the probability of a generic startup failing [can be as high as 90%](https://www.failory.com/blog/startup-failure-rate), this greatly depends on a multitude of factors: Is the startup in Silicon Valley or Barcelona? Are the founders technical or business experts? Do they have prior leadership experience? Is the startup backed by top VCs or came out of YC? Is the startup working on a “hot space” like AI? And so on. As an example, the article linked above already reduces the risk of failure to 75% if the startup is in the Silicon Valley and to less than 40% in some sectors. So, what happens if we move the 80% failure rate to 10% and we assign that 70% to the next likely scenario of being acquired? Well, we end up with an expected value of **$985k a year**… not bad. Again, I am not saying the second choice of probabilities is the right one for most situations, but it clearly will be for some. You need to work on figuring out what are the right ones in your case.

Something else for you to consider when joining a startup is that you can continuously re-evaluate those probabilities. Of course during the interview process you should try to understand as much as you can to make an informed decision, and that includes the current valuation of the company. However, no matter how much you learn before you join a startup, you are much more likely to get detailed data once you are working there. This is particularly important at early stage startups where the potential of the company can change very quickly and depends greatly on the quality of the team, which is much easier to evaluate from the inside. Keep your expected value computation at hand, and re-evaluate as you learn more about the company and the space. Speaking of which, I always recommend learning about most direct competitors and use their valuations or market caps as a good way to estimate potential outcomes. You might want to use a combination of companies in your space that have already had an exit (IPO or acquisition), and some that are closer to whatever the stage of your startup is. Another important aspect of joining a startup if you have a strong resume is that if you join and you don’t like what you learn once you are in, you can always leave. While I always hope that is not the conclusion of anyone joining my startup, the reality is that the optionality to leave adds to the upside of a startup opportunity (see [this blog post](https://www.benkuhn.net/optopt/) for more details on this).

### Strike price, vesting, fair market value, and preferred price

The discussion above is valid regardless of what kind of equity you are getting at your company/startup. However, it is key that you understand the exact details of what you are getting and how to think about it.

Generally speaking, in a startup you will get “stock **options**”. This is different from e.g. RSU’s that you will get at a larger company. Options should be understood as your “right to buy shares”. When you get an option, you really have nothing except the ability to buy **shares,** but you don’t own the shares themselves. Weird, you might say. Well, let’s understand how that works at least at a high level.

When you get an option grant, say for 1,000 options, you usually don’t get anything on day one. With your grant, you will get what is called a **vesting schedule**. This is basically telling you how those 1k options will be distributed over time. A typical schedule is 4 years with a one year cliff. This means that out of those 1k options, you will get 250 after your first year anniversary (cliff). You will get the other 750 monthly, over the next 3 years.

Again, when you get your options, you don’t have shares of the company. All you have is the option to buy those shares. And, what is the price you can buy your shares at? That is the **strike price**. The strike price is determined at the time you get your original grant by using the concept of fair market value (FMV). This is determined by a process run by a financial institution (accounting firm, bank, or even Carta) and it is called the [409A valuation](https://carta.com/blog/what-is-a-409a-valuation/). Note that the strike price for this particular grant will remain constant over time. What makes the grant valuable for successful start-ups is that over time the difference between the selling price and the purchase price will become very large (the **spread**).

![](/blog/images/04-04.png)

So, how can you determine the “stock price” over time in the graph above? The reality is you can’t determine

100% until there is some form of liquidity event (see section below). However, a good approximation is to look at the (preferred) share price of the last valuation since, if things go well, that is all you need to go. If things don't go well, investors get… well, preference. They might get their money back while employees get nothing. So, yes, this probability, however small it might be, should be factored in your 0 equity payback scenario when computing your expected value.

How should you compute the other possible outcomes for the expected value computation? Simple: take the number of options/shares times the spread (stock price minus strike price). In order to come up with possible stock prices look at comparable companies as mentioned above and find a multiplier for the current startup valuation. Don't worry too much by how realistic they are. You can always adjust that in the probabilities! (Also, if you are wondering about dilution, we'll discuss that later in the post)

Also, you should understand that the initial compensation package that you get when joining a startup is not the end-of-it-all. Part of your research about the company should include asking whether the company updates salaries on a regular basis, promotes, and offer grant refreshers, which are new equity grants that are issued during your tenure so that you are always vesting shares. Your first grant might just be the onramp to getting value out of company equity, and similarly your first cash salary is likely to be upgraded as you and the company grows.

### Liquidity events

While you can exercise your vested options (i.e. buy shares for options that have vested) at any time, you can really only sell them for money if there is someone willing to buy. This generally only happens if there is a "**liquidity event**". The most likely liquidity events for a startup are an IPO, or an acquisition. So, generally speaking you should not expect to sell (and therefore make money) before then. This is important, since, regardless of your expected value, the timeline of that expected value might affect your computations.

There are other less common liquidity events that have to do with the so-called "**secondary market**".In some circumstances, startup shares might be sold before the company goes IPO or is acquired, either when the company itself chooses to offer to buy some of the shares back, or via companies specializing in selling your private shares. More recently, there are companies that specialize in selling your private shares to private buyers that are interested in acquiring them, often at a discount though (see e.g. [here](https://equityzen.com/knowledge-center/newsletter/what-is-a-secondary-market/) or [here](https://sharespost.com/marketplace/selling-shareholders/) for more details).

### When to exercise

So, given that you won't be able to sell your shares until a liquidity event, you might be asking yourself why you would want to exercise your options before then. Again, exercising your options means buying the shares for your vested options at the strike price. So, you will have to pay real $$'s out of your pocket. Why would you do that? In a word: taxes.

Again, it is beyond the scope of this post to go into the details of how options and shares are taxed, but it is in your best interest that you understand these details on your own (see [this guide](https://www.holloway.com/g/equity-compensation/sections/taxes-on-equity-compensation) for more details). Roughly speaking though, depending on what you do you can end up paying as low as 20% in long-term capital gains, or almost double in ordinary income. That depends on a number of things, including the kind of options you have (ISO or non qualified).

![](/blog/images/04-05.png)

If we look at ISOs (the most common case), you generally don't have to pay taxes when you exercise ([unless you trigger the alternative minimum tax](https://turbotax.intuit.com/tax-tips/investments-and-taxes/incentive-stock-options/L4azWgfwy)) and you will pay Capital Gains when you sell. If you sell at least one year after you exercise, and two from your grant, you will qualify for long-term gains. In some cases, particularly in very early stage startups, you might even benefit from exercising options before they have even vested. In other words, you are buying something you still don't have the rights to with the condition that you will be giving it back if you leave before vesting. This is called [early exercising](https://www.holloway.com/g/equity-compensation/sections/early-exercise) and is not always possible or advisable. But, it is something you might want to know about.

![](/blog/images/04-06.png)

You should become familiar with tax implications, since they'll be critical if the company has a successful exit.

### Dilution

A common mistake made by many startup employees is to over-obsess and misunderstand dilution. The reality is that unless you are a founder or a very early employee, dilution doesn't matter much. And, even in those cases, it only matters if handled incorrectly.

So, what is dilution anyway? Dilution is the process in which when a company issues additional shares, your overall ownership of the company gets reduced. This usually happens in the context of a new financing round. You might own 1% of the ACME startup after series A. In the next financing round, the company issues more shares to accommodate the new investors. Your 1% is no longer 1%.

That is many times interpreted as a negative. However, that is usually not the case. If the startup is growing, and the new shares are issued in the context of an up-round, the value of your options or shares is probably going to be higher than before, regardless of what percentage of the company you own! The value of your equity is always the number of shares times the price per share. And the price per shares is determined by the valuation of the company divided by the total number of shares issued. As long as that multiplier keeps going up, you are in good shape. Your percentage of ownership of the company is simply a side effect of the above and will be reduced over time. See [this example](https://carta.com/blog/equity-101-stock-economics/) in the Carta blog for more details.

So, how should you account for dilution in your expected value computation? The easiest way to think about it is the following: in order for the shares value to multiply times 10, the company valuation needs to multiply by more than 10. How much more? Hard to say. That depends on the number of rounds from now until the liquidity event and how much the company is diluted in each round. As you can see [here](https://www.quora.com/What-is-the-usual-percentage-of-shares-that-go-to-seed-Series-A-and-Series-B-rounds) and other answers on Quora, a typical startup will take 20–30% dilution in the initial rounds and less than 10% at later rounds. Note though that there is a lot of variability on this, so you need to look at many things to get it right. My advice is simply tweak the probabilities in the expected value to account for this. Also remember that if your compensation goes up and you get equity refreshers those are more than likely going to make up for any dilution.

It is important to note that just as it doesn’t make much sense to compare percentages of ownership across different startups, it does not make sense either to compare number of options or shares. That is just like comparing 10 American dollars to 10 Japanese yens. You should be comparing the value, not the number of bills you own!

### Bottom line: don’t join a startup only for the money

I hope the post has clarified how to think about the monetary opportunity of a startup. Once we got that out of the way, here’s the deal: if you are only thinking about becoming rich, don’t join a startup. While making money is a real possibility, there is also quite some risk associated with it. Plus, there is the timeliness element of having to wait for liquidity. Even if you’re just focused on money, you should consider factors with massive long-term financial impact: career growth, networking, technologies exposure, leadership opportunities. Going back to what I introduced at the beginning of this post, the **utility value** of you going to a startup might be in fact higher than the **expected value**! I hope to get into those other factors in a future post.

### A spreadsheet helper

If you liked what you read, and you want to this framework, [here](https://docs.google.com/spreadsheets/d/1uqETG2kl2mnbqh7aQvJJIu7DeAT0uLOwuwxrki0vgs4/edit#gid=0) is a simple spreadsheet for you to play around with.

### Thanks

To Steve Herrod, Erik Bernhardsson, Neal Khosla, and Francois Huet for feedback on early drafts of this post.

```
---
id: 26
title: "On the “Usefulness” of the Netflix\_Prize"
date: '2021-06-23T00:00:00+00:00'
author: xamat
##layout: post
permalink: /on-the-usefulness-of-the-netflix-prize-403d360aaf2/
header:
    teaser: https://amatriain.net/blog/images/03-02.png
reading_time:
    - ''
    - ''
categories:
    - Uncategorized
---
```

It has been over 10 years since the Netflix Prize finished, and I was not expecting to write a blog post about it at this point. However, just in the past couple of weeks I have found myself talking about it extensively both in the context of a [Twitter thread and discussion](https://twitter.com/skyetetra/status/1405243698888118274?s=20), as well as the shooting of an upcoming documentary series. Given that there seems to be continued interest as well as misunderstanding around the prize and its outcome, I thought it might be worth to “set the record straight” in a dedicated post.

TLDR*; While I am often misquoted as having said that the Netflix Prize was not useful for Netflix, that is only true about the grand prize winning entry. Along the way, Netflix got far more than our money’s worth for the famous prize.*

### **What was the Netflix Prize anyways?**

Ten years ago, when I was leading Algorithms at Netflix, I would start my public presentations by asking the audience a couple of questions: “Who has heard about the Netflix Prize?” and “Who has participated in the Prize by submitting at least an entry?”. Virtually all hands would go up to the first question. Depending on the audience, more than half of the people would also raise their hands to the second. I would obviously not expect that to be the case anymore. However, I have recently been pretty surprised by how much folks in the ML/AI community remember or simply know about the contest. It is clear that it was somewhat of a historical event.

![](/blog/images/03-01.png)

In 2006, Netflix launched a $1M competition to improve their recommendation algorithm. At that point I was a researcher in the field in a Barcelona lab, so I don’t have first-hand details of how and why the prize was decided internally and how it all got started (to be fair, I did get some of that insight once I joined the company 5 years later, but that part of the story is not very relevant anyways). For the thousands of participants around the world, the rules were quite simple: if you were able to beat the Netflix baseline (called Cinematch), you would win $1M. Not surprisingly, many researchers, students, and amateur mathematicians and computer scientists around the world jumped at the opportunity.

The mechanics of the contest were also quite simple: there was a training set consisting of around 100M data points including a user, a movie, a date, and a rating from 1 to 5 stars. There was a smaller public validation dataset called “probe” consisting of around 1.5M data points with no ratings. Finally, and very importantly, there were two test sets that hid ratings from participants. In order to test your algorithm you would submit your predictions to the so-called quiz test and you would get back the accuracy measured in RMSE (root mean squared error). Yearly progress prizes, and especially the grand prize, would be measured against a different test set.

![](/blog/images/03-02.png)

There were hundreds of forum and blog posts and research publications detailing the different approaches to the Netflix Prize. The first year progress prize was won by the Korbell team (Yehuda Koren, Robert Bell, and Chris Volinsky from AT&amp;T labs) using an ensemble of a variation of SVD (singular value decomposition, although the variation is really more a matrix factorization than a traditional SVD) and RBM (restricted Boltzmann machines, a form of ANN). The SVD had an RMSE of 0.8914, the RBM 0.8990, and a linear blend of both got to an RMSE of 0.88.

It took 3 more years and thousands of teams trying to get from there to the 0.8572 RMSE on the test set that was required to win the grand prize. The [winning entry](https://www.wired.com/2009/09/how-the-netflix-prize-was-won/) was an ensemble of 104 individual predictors developed by multiple teams, and ensembled by a single layer neural network.

### **Making history**

Back in 2006 there was no Kaggle, open source was something that fringe Linux hackers did on their “free time”, and AI was a dirty word. Taking that context into account, it is easy to see how groundbreaking the prize was in many ways. Releasing a large dataset and offering $1M to anyone who was able to beat a metric was definitely a bold move, and one that not many understood back in the day. Hell, many even don’t understand it nowadays.

I am pretty sure that without the Netflix Prize we would not have data competitions as we have now. Or, at the very least, it would have taken much more for Kaggle to take off. I have to wonder if much of the rest of the open ecosystem around AI, which includes from datasets à la Imagenet, to open source frameworks like TensorFlow or Pytorch, would be here if it hadn’t been for the Netflix Prize. Yes, I know that there was open source and even open data back in the day, but none of that was supported by a large company, let alone one that was actively supported with actual cash.

### **The grand Netflix Prize solution**

The 2007 Progress Prize solution that combined SVD+RBM was already significantly better than the existing Cinematch algorithm. So, Netflix put a few engineers to work on productionizing the algorithm. That included rewriting the code and making it scalable, plus able to retrain incrementally as new ratings came in. When I joined Netflix I took over the small team that was working and maintaining the rating prediction algorithm that included the first year Progress Prize solution.

But, what about the grand Prize solution with the 104 algorithms? As I mentioned in [a blog post](https://netflixtechblog.com/netflix-recommendations-beyond-the-5-stars-part-1-55838468f429) back then, we decided it was not even worth to productionize. It would have taken a large engineering effort for a small gain in accuracy that was most likely not worth it for several reasons. The main one is that at that point in time, once the transition to streaming (vs DVD-by-mail) was obvious, it was clear that predicting consumption was much more important and impactful than predicting ratings.

![](/blog/images/03-03.png)

### **Let’s talk ROI**

I am often surprised that after reading the story above [many will conclude] (https://www.wired.com/2012/04/netflix-prize-costs/) that Netflix' $1M investment was not worth it. That is definitely a very short-sighted read. Do you know how many Silicon Valley engineers Netflix can hire for 3 years for $1M? Probably less than one.

What did Netflix get from the $1M invested in the prize?

- Thousands of researchers and engineers around the world thinking about a problem that was important to Netflix
- A solution (the first year Progress Prize) that could be launched into production with measurable gains
- Becoming a recognizable brand name as a company that innovates in the space

One thing that I am absolutely certain of is that if it hadn't been for the Netflix Prize me and many others like me would not have ended up working at Netflix. I doubt Netflix would have been able to innovate at the pace they did during those years if it wasn't because of the talent that it attracted as a result of the prize. That in itself is worth much more than $1M.

### **Conclusion: are open algorithmic contests useful and valuable?**

Of course, you can argue that I am quite biased on this topic. However, even as of today, I would say that there is huge value in them. Getting a large community to think about a data set and a particular problem often produces lots of insights and valuable results, some of which are applicable in practice. They also provide a unique opportunity for people around the world to learn, get some exposure, and in many cases even find a dream job they would not have found otherwise.

That being said, I do agree that algorithmic contests are not the silver bullet that will solve your company's algorithmic needs. They should not be considered a way to outsource problems to cheaper countries. And, of course, it is important for the organizations in charge of proposing the competitions to make sure there are good incentives and fair rules. But I feel companies like Kaggle (now Google) have done a really good job at setting up the right framework for this.

Finally, I will say that I am very intrigued and interested by Andrew Ng's proposal to turn algorithmic contests around by keeping the algorithm fixed while having participants work on the data in his [Data-centric AI Competition](https://https-deeplearning-ai.github.io/data-centric-comp/). I am sure we can learn quite a lot from having a large community getting an incentive to work on data rather than on the algorithm.

```
---
id: 9
title: "The death of the stars: A brief primer on online user\_ratings"
date: '2022-05-30T00:00:00+00:00'
author: xamat
##layout: post
permalink: /the-death-of-the-stars-a-brief-primer-on-online-user-ratings-6740453f27ed/
header:
    teaser: https://amatriain.net/blog/images/01-01.png
reading_time:
    - ''
    - ''
categories:
    - Uncategorized
---
```

A few weeks back, Netflix [announced](https://about.netflix.com/en/news/two-thumbs-up-even-better-recommendations) that they were adding a two thumbs up option to give feedback on their content. At the same time I was being asked on Twitter why we had given up on 5 star ratings at Netflix back in the day. Given that I have personally spent a lot of time in my career looking into ratings and user feedback, I thought I'd share some of what I have learned not only during my time at Netflix, but also before and after.

I became fascinated with user ratings like many other folks in the context of the famous $1M [Netflix Prize](https://en.wikipedia.org/wiki/Netflix_Prize). That dataset had a lot of interesting data, and also very unexpected findings. Lots was written about it at that time. For example, I wrote [a pretty popular post](http://technocalifornia.blogspot.com/2011/04/recommender-systems-were-doing-it-all.html) around that time explaining why the assumption that ratings are in an ordinal scale might be wrong.. Many of us continued looking into the topic for years to come and even made a career of it. If you are interested in some of the many lessons I learned from that contest I wrote about it in an earlier [post](https://medium.com/p/403d360aaf2).

![](/blog/images/01-01.png)

One of the things that becomes obvious as soon as you start looking at user feedback is that there is a tremendous amount of noise. The rating a user will give to an item will depend on how the item is presented, how much is remembered, or whether it is a sunny or rainy day. We wrote about it in our "I like it, I like it not" [paper](https://scholar.google.com/citations?view_op=view_citation&hl=en&user=YYUK51oAAAAJ&citation_for_view=YYUK51oAAAAJ:Y0pCki6q_DkC). In that work, we devised an experiment where we ask the same users to give feedback to the same items at three different times while controlling for some of the variables such as the order they were rating the items. The takeaway of the experiment was that the error of a user rating a movie in a 1 to 5 stars scale was "at least" of +/-

0.5 stars. This error depends on many aspects, including the rating itself since ratings in the middle of the scale (2 or 3) are consistently more noisy. It is important to keep in mind therefore that a user preference can never be considered as a “ground truth” for any predictive task. As a side note, this is one of the main thesis in Kahneman’s [Noise book](https://readnoise.com/), which I totally recommend.

![](/blog/images/01-02.png)

Another important aspect that affects the reliability of a rating is the expertise of the rater. Expert ratings are in general much more reliable and predictable, and, as we showed in our “[Wisdom of the Few](https://scholar.google.com/citations?view_op=view_citation&hl=en&user=YYUK51oAAAAJ&citation_for_view=YYUK51oAAAAJ:IjCSPb-OGe4C)” work, they can be much more reliably predicted and therefore used as a great source of user recommendations. (Fun fact: when I was at Netflix I pitched Reed that we should buy Rotten Tomatoes. He laughed it off almost as much as when I recently pitched that Netflix should buy the local cinema theater 🙂

![](/blog/images/01-03.png)

Ratings have all other kinds of issues that have been described in many of my talks. E.g. They are aspirational (people rate as they want to be perceived, not as they act), and depend on cultural factors: Some cultures only rate extremes, while others rate using the middle of the scale as has been found in many studies on cultural patterns in survey responses (see this [paper](https://www.emerald.com/insight/content/doi/10.1108/02651330710741785/full/html) e.g.)

Finally, the reliability of a rating also depends on the “effort” required from the user. If you build a complicated interface that requires the user to evaluate a number of features, you will get much more reliable ratings. Of course, the flip side of this is that you will also get many LESS ratings!

This last point is extremely important, because it can be translated into the following: there is a direct tension between the amount of user feedback you will get and its quality! You can either get tons of crappy feedback from everyone, or very good feedback from a select few. Navigating that tension requires a lot of careful and dedicated UX research (you can read about such research every year in the proceedings of conferences such as [CHI](https://chi2023.acm.org/), Computer Human Interaction, and [IUI](https://iui.acm.org/2023/), Intelligent User Interfaces)

So, with all this context in mind, why did ratings go away at Netflix? It should not come as a surprise: they simply became useless for how the product was operating. We started talking about it in [this](https://netflixtechblog.com/netflix-recommendations-beyond-the-5-stars-part-1-55838468f429?gi=133a9bbebcac) blog post where we already hinted about how much more reliable implicit feedback was. In general, implicit feedback (i.e. consumption signals from users) is much more reliable (it is hard to fake watching a show) and readily available from every user. Plus, it “only” requires logging (i.e. engineering work). No hard UX work and experimentation.

So, that was the “easy” answer, not without a lot of prior AB testing and effort in trying to rescue them. Was it the “right” answer? Well, in a product that was fully transitioning to streaming mostly on devices like TVs and, less so, mobile, it became really hard to come up with a feedback mechanism that provided just enough friction to be reliable, but not as much to make this feedback only come from an “elite” of dedicated users. I am glad to see that Netflix is now adding the two thumbs up, which means they still have not given up on explicit feedback. Other online applications such as Facebook or LinkedIn have also invested recently in enabling more nuanced explicit feedback from users. When I was at Quora, we also invested in taking into account a lot of different explicit forms of feedback such as upvotes, downvotes, expands, or shares (see presentation [here](https://www.slideshare.net/xamat/machine-learning-for-qa-sites-the-quora-example)).

![](/blog/images/01-04.png)

My personal take is that explicit feedback is far more valuable than pure implicit signals that are usually only directly connected to overall engagement, but have no information about the value that users took away from the service. This is a much longer discussion though, that I will leave for a future post.

```
---
id: 90
title: "Migrating Medium and Blogger blogs to self-maintained Github Pages"
date: '2022-06-07T00:00:00+00:00'
author: xamat
permalink: /migrating-my-blog
header:
    teaser: https://amatriain.net/blog/images/90-05.png
reading_time:
    - ''
    - ''
categories:
    - Uncategorized
image: /blog/images/90-05.png

---
```

In case you are reading this post in my now Medium mirror, you can access the result of this migration [here](https://xamat.github.io/blog/).

For quite some time I have been pretty disappointed with my current [Medium blog](https://xamat.medium.com). It has been sad to see how much Medium has failed as a product. On the one hand, the tools and services for writers are horrible (as an example, they recently removed the ability to edit on mobile!). On the other hand, Medium distribution is completely dead. I get most of my views through Twitter and LinkedIn, and any of my Quora answers gets orders of magnitude more reads than my Medium post. Finally, they are not friendly to readers either (and they make it harder and harder to find the damn Friends Link). I have felt captive to Medium for the only reason that I did not have enough time to invest into alternatives. But, I thought that I just HAD TO. Here's what I learned in the process of "freeing" my posts from Medium. In doing so, I also thought it would be a good idea to bring along my old [Blogger posts](https://technocalifornia.blogspot.com/).

![](/blog/images/90-01.png)

## Alternatives

I have maintained my [webpage](https://xavier.amatriain.net) in Github Pages for some time now. Going all in and finding a way to publish a blog using basic html or some other tool was top of mind. However, I did consider a couple others. The first one is [Wordpress](https://www.wordpress.org), which I will cover at length later on. The other one is [Substack](https://substack.com/), which I have seen used quite a lot lately, even for people who do not use it for its primary purpose of sending newsletters. If I were more serious about monetizing my writing, or I even liked newsletters at all, I might have been tempted to give it a try. But, the reality is that in any comparison you read, you will see that Wordpress comes on top on almost anything, but probably easiness of use. That was something I was not prioritizing. But, if you are not a techie and are looking to earn cash from your posts, Substack might be worth looking into.

## Why not Wordpress

[Wordpress.org](https://www.wordpress.org) is a powerful publishing platform that has existed for many years (almost 20 to be precise). It is open source, and has a very strong plug-in ecosystem that you can use for almost anything you can imagine. So, why not go with Wordpress and call it a day? For most people it will come to a very simple reason: you need a host. Wordpress is a software, and it needs to be hosted somewhere. Unless you have a server laying around in your basement, you are going to need to pay someone to host your blog, and that costs some money. Now, to be clear, it is not a lot of money. Recommended hosts such as [Bluehost](https://www.bluehost.com/wordpress-hosting) have plans starting under $3 a month. And [Wordpress.com](https://www.wordpress.com) even has a barebones free hosting option.

In my case though, it was not about the money. I was looking for a truly open solution where I felt like I owned the pipeline end to end. Having to pay someone to host Wordpress did not feel like the way to go. Plus, as I mentioned before, I already had my webpage in Githubs Pages and was looking to continue using Github's infrastructure for publishing.

## Exporting your Medium (and Blogger) posts

Regardless of what your target platform is, you are going to have to start with exporting your existing blog posts. How do you do that? There are basically two options that I explored (and eventually got both to work): (1) Find a script that downloads your posts directly, and (2) Use Wordpress, and its many plugins, to export and aggregate your content.

### Exporting scripts

Here is the thing, there are many scripts out there that promise to do the job of exporting your posts from your blog to some common format (mostly markup or .md). Most of those scripts probably worked at some point, but they don't anymore. I spent a bunch of time trying out scripts that did not work until I found one that did. As a rule of thumb, any script that has not been updated in say the last year or so is unlikely to work.

I came up with the one that did work, [Medup](https://github.com/miry/medup/) a bit late in the game, when I had already figured things out through Wordpress. I still would recommend you to start there since it worked well, and the developer was super responsive. Now, if you are reading this in say 2 years, things might be different and the script might not work anymore. That's why the second approach is always a good fallback to have.

### Exporting and merging through Wordpress

This is the more complicated route that I ended up following. However, it was worth it for me to get to install Wordpress locally, understand how plugins worked, and use Wordpress as an intermediate step to integrate posts coming both from Medium and Blogger into a single platform, even if that one was temporary.

The first thing you need to get in order to follow this route is to install Wordpress. While Wordpress.org boasts about their "5 minute installation, this was probably the most time consuming part of the process. You can indeed install Wordpress in 5 minutes… if, and only if, you already have a working AMP (Apache, MySQL, and PhP) installation in your system. That is a breeze in any Linux system, but it took way more than expected on my recently upgraded Mac OS Monterey. I could not get it to work until I "cheated" my way through [XAMPP](https://www.apachefriends.org/index.html), a cross-platform AMPP installer. I came across XAMPP through some tutorials recommending it for Wordpress (see [this one](https://www.hostinger.com/tutorials/how-to-use-xampp-wordpress/) e.g.), and it did make everything much easier than trying to get AMP up and running directly.

Once you have XAMPP installed, you still need to install Wordpress. The most tricky part of that [“5 minutes installation”](https://wordpress.org/support/article/how-to-install-wordpress/) is to get the mySQL database started, but you can find good articles such as [this one](https://wordpress.org/support/article/creating-database-for-wordpress/) that will guide you through the process.

![](/blog/images/90-02.png)

![](/blog/images/90-03.png)

Once you are done with the installation, you should be ready to go. You will be able to use Wordpress locally. From Wordpress, you can install plugins to import content from Medium and Blogger (or wherever you have your old blog) into Medium, and from Wordpress to markup for then taking to your Jekyll blog. Basically you can use Wordpress as the ultimate tool to import and export content from anything to anything. I ended up using a few plugins including Blogger Importer, Blogger Importer Extended, Wordpress Importer, WP Smart Import, Wordpress Importer, Wordpress to Jekyll Exporter, and Auto Upload Images.

I should have kept better notes of what exactly worked, since I had to try a combination of different things, but for importing the Medium content, I basically had used the [following tool](https://mediumtowp.com/) as explained [here](https://themeisle.com/blog/migrate-medium-to-wordpress/) .I did have to add auto upload image plugin though, and even with that I struggled with images and had to use the script above. But, a combination of the above should more or less work, and you should soon have all your posts in Wordpress ready to edit and/or export to Jekyll markup.

![](/blog/images/90-04.png)

## Creating your Jekyll blog on Github Pages

The final step, although you could really start here particularly if you don’t have much old content to transfer, is to create a Jekyll blog on Github pages. [Jekyll] (https://jekyllrb.com/) is a wonderful little framework that enables you to create webpages and blogs fairly easily by using a simplified markup language and some configuration files. It has the added benefit of being fully and directly compatible with Github pages. So, all you need to do is start a Jekyll compatible repository on Github, and Github will render it!

Taking all of this into account, I decided that the easiest way to start would be to fork an existing Jekyll blog. Lo and behold, while reading [this howto guide](https://www.smashingmagazine.com/2014/08/build-blog-jekyll-github-pages/), I found [Jekyll-now](https://github.com/barryclark/jekyll-now), a popular Jekyll blog template that has been forked a whooping 34k times. I forked the project into [my github](https://github.com/xamat/blog), pushed the blogs that I had exported into it, and was up and running in no time. As a side note, because I did not want to host the blog directly in my Github Pages root (i.e. xamat.github.io), I had to create the fork into a gh-pages branch (see documentation [here](https://docs.github.com/en/pages/getting-started-with-github-pages/configuring-a-publishing-source-for-your-github-pages-site)).

## The result

![](/blog/images/90-05.png)

## The ugly

After going through all those steps, I did have to waste some time in formatting some of the older blog posts. The one thing that I spent the most time on was fixing image URL’s since moving posts from one place to another and using several plugins and scripts did not preserve the paths. That was a somewhat tedious process, but it did give me the opportunity to revisit content I had not read in a long time.

## What is next?

I have just edited my Jekyll posts with the bare minimum. I have not even added labels and categories, nor explored styling options. Similarly, Jekyll has a lot of extensions and plugins too, and I haven’t even scratched the surface. I plan on adding more capabilities to my new blog. In particular, I need to add stats to keep track of the visits as well as comments.

In the meantime, I will continue to cross-post in Medium just as an experiment. I am interested in seeing what happens on Medium if I don’t share any of my posts externally. Over time though I plan to shut it down. The only question is whether Medium itself will shut down sooner than that, judging by its lack of progress and promise.
---
id: 91
title: "Principles for data driven organizations"

```
date: '2022-07-09T00:00:00+00:00'
author: xamat
permalink: /data-driven
image: /blog/images/91-01.png
header:
    teaser: https://amatriain.net/blog/images/91-01.jpeg
reading_time:
    - ''
    - ''
categories:
    - Uncategorized

---
```

![](/blog/images/91-01.jpeg)

## What does it mean to be “data driven”?

Data driven is a somewhat controversial term that has many definitions and interpretations. In fact, many argue that being data driven can harm organizations, and instead argue that they should be [data informed](https://www.scuba.io/blog/data-driven-vs-data-informed#:~:text=This%20is%20the%20fundamental%20difference,a%20check%20on%20your%20intuition.) . I believe that this stems from an incorrect interpretation of the term. Because of this, it is worth to start by clearly stating what we mean by it:

A data driven organization is one that makes decisions that are informed by quality data whenever possible. In order to do this, the organization seeks data proactively, and invests in mechanisms and tools that enable quality data to be available whenever the decision will be made.

Note that being data driven does not mean that every single decision is made by looking at data. As much as we might seek quality data, there will be times that this data is unobtainable. unavailable, or of low quality. In those situations other mechanisms for ensuring the quality of the decision will be necessary.

I believe that other terms such as the aforementioned “data informed” are ill suited because they convey too much of a passive attitude towards data in which data is assumed to “exist” and needs to be interpreted subjectively, and even “mistrusted”. This leads to posthoc data reads that in the most extreme case derive in organizations deciding after the fact whether the data tells the story they want to hear or needs to be discarded.

## The pillars of a data driven decision culture

There are many aspects involved in making data driven decisions. I summarize them in the following four pillars of a data driven decision culture:

1. A hypothesis-driven approach to data in which the hypothesis and the metric that will be used to test it should be specified beforehand. In this approach, questions are asked and formulated ahead of time, and data is sought in the right way to answer the question at hand.
2. Metrics used to make decisions are explicitly communicated, known, and agreed upon by all stakeholders
3. Data that enables decision making processes is expected to be trustworthy, That means that data quality mechanisms that guarantee such quality need to be in place.
4. Data is easily accessible by stakeholders. This involves some level of investment in creating user interfaces to data such as dashboards and reports.

In the following sections, I describe the tenets or principles that drive each of those pillars. Note that these tenets should be operationalized by tools and mechanisms to support them in different teams. They are important because everyone in a data driven organization should agree to work together towards them and to call out whenever they are being compromised.

### Hypothesis-driven product development

It is very hard to answer questions using past data, particularly when that data is limited or questionable. When data is extracted and prepared after the fact, it is very likely that this data is much lower quality than if we had made sure that it was going to be generated appropriately beforehand. There is also the danger of over-manipulating past data to fit your current assumptions. That is why we should strive to ask questions first, and then proactively design experiments to gather the kind of data that will answer them.

For every project we should aspire to set a hypothesis by asking the question of “Why?” at the onset. For example: “Why are we changing the color of our login button?”. The answer to a question like this might be something like “Because we expect users to log in at a higher rate if we use a different color”. The “why” question and its response can be easily rephrased as an experimental hypothesis. In our example this would be “Hypothesis: Changing the color of our login button will increase log in rate”. Some readers might recognize this “why” question as somewhat related to the so-called causal reasoning (see Book of Why). Strictly speaking, the data driven mechanisms and tools most people use and are described in this post do not necessarily enable causal inference, but they do get us a step closer for sure.

An immediate consequence of formulating hypotheses in this manner is that we can quickly derive what metric we are hoping to move and in what direction. In our running example we mentioned “log in rate”. Note though that when turning this into a measurable metric, we are likely to run into more concrete questions. What does “log in rate” mean? We could decide what we expect is for users that have never logged in before to log in now for their first session. Or maybe we do expect for this change to impact both new and existing users who logged in before. It might even be that we expect the direction of the change to be different! Because of this deeper discussion, in many cases we might end up breaking a hypothesis into several sub-hypothesis, and identify different metrics or groups of users to track independently. In our case, we might end up with something like:

- Main Hypothesis: Changing the color of our login button will increase the overall log in rate of our users as defined by [Metric] percentage of logged in sessions per user.
  - Sub-hypothesis 1: Changing the color of our login button will increase first login probability of our new users as defined by [Metric] median number of logged out sessions before user logs in for the first time
  - Sub-hypothesis 2: Changing the color of our login button will slightly decrease successive log in rate existing users as defined by [Metric] percentage of logged in sessions per user who had already previously logged in. This slight decrease is expected to be more than compensated by the increase of new users that we are going to convert to logged in.

Very importantly, we can now make sure that we have the ability to track the metrics we have identified as necessary to validate or invalidate the hypothesis. If the metrics don’t exist or are questionable, we should remedy it before the experiment runs.

The most common, but not only, approach to running such experiments in a product is AB testing. It is beyond the scope of this post to go over the many aspects that are required to run a good AB test (I will instead direct the interested reader to Ronny Kohavi’s [Trustworthy Online Controlled Experiments] (https://www.amazon.com/Trustworthy-Online-Controlled-Experiments-Practical/dp/1108724264/)).

Hypothesis-driven experiments do not necessarily require an AB test. This is many times an excuse I have heard from people in other disciplines for not being strict with their experimental design. How can, for example, a UX research project be hypothesis driven? In our running example of the login button, we would show mocks of the site with the two colors to users and ask them how likely they are to log in. We could even develop two prototypes with the difference in color and let users play around with them in a controlled environment (although in this particular example this is probably not worth it vs. directly running a live AB test if our product is live already).

It is important to note that, in any case, sometimes reaching a conclusive answer will require several rounds of experimentation. This can be so for many reasons, but an important one is that we might not get data that conclusively and statistically proves or disprove our hypothesis. We understand that, and we might treat some answers as tentative in the meantime.

#### What about “exploratory” projects?

Whenever I describe the approach above, many folks react by asking “Do you mean that I can’t decide to prioritize an exploratory or discovery project for an area where I might not yet have a clear hypothesis?”. The answer is, of course, yes. Such projects might be important when entering a completely new space in order to uncover possible trends or generate ideas. However, these should be treated as exceptions, and should be explicitly called out. The corollary is simple: if your organization runs mostly exploratory projects or experiments, you are not in a data driven organization.

### Metrics

A data driven organization should clearly define, document, and publish its metrics. Those metrics should be known and agreed upon at the appropriate level of the organization. Metrics and their documentation should be kept up to date, and any change to metric definitions should be documented in some form of changelog so that they can be tracked over time.

Metrics should be organized in a hierarchy such that the top level metrics represent the most important business metrics that drive the organization goals (see example from Gartner below). Those higher level metrics should be agreed upon and approved by the executive teams. Lower level metrics will be defined and maintained by the appropriate team leads.

![](/blog/images/91-02.png)

The quantity of metrics an organization tracks does not define how much data driven they are. In fact, it is almost the other way around: A truly data driven organization should strive to have as few metrics as possible. Of course, the hierarchy described above means that different levels and parts of the the organization will care about a different subset of metrics. An important negative side effect of having too many metrics is that they end up not being important (“if everything is important, nothing is important). Another negative side effect is that maintaining too many metrics is hard, and it increases the chances that they become noisy and unreliable. This is what I call the “metrics death spiral”: too many metrics to maintain that become unimportant and therefore are maintained poorly, becoming even more unreliable, and therefore less important, and the organization responds by adding new metrics that feed into the problem, making it worse and worse.

An important litmus test for whether we should track and incorporate a given metric is whether it is actionable. In other words, will that metric inform something that can be actioned upon? Whenever anyone comes to me asking for adding a new metric, I ask them the following question: “How is that metric going to help you make a decision?”. We should be able to answer this question precisely for any metric we track. As an example, and going back to our login button, imagine that the percentage of logged in users is a metric we keep track of. What would happen if this metric all of the sudden drops without a clear cause? If the answer is that we would not do anything, we should then not track the metric. Very importantly, you should not track metrics just to satisfy someone’s curiosity (unless that someone is the person that is paying your organization, of course).

In this section so far we have described organizational metrics, but how do these relate to the experimental metrics mentioned in the previous section? Ideally, they fully overlap. In other words, when defining a hypothesis for a new experiment I would be wise to choose one that can be verified using a metric from the existing organizational metric hierarchy. Why? Well, because those metrics are supposed to represent what the organization cares about and prioritizes. Why would I want to run an experiment and change a metric my organization does not care about or prioritize?

In practice though, particularly if you are in a young organization (e.g. startup), it is likely that a new project with a new hypothesis will uncover a new metric.That is expected, and completely ok. My recommendation though is to then spend some time discussing whether that new metric is important enough to make it into the “select club” of the organization metric hierarchy. Make the actionability of the metric part of the discussion. And, remember that we don’t want to just continue adding every single metric we can think of to the hierarchy to avoid the metrics death spiral!

### Data quality

A data driven organization strives to have highest data quality standards, and understands that poor data quality drives poor decision making. Because of this, it will not accept poor data quality, and will point to it whenever and wherever it manifests, with the goal of increasing overall quality over time. Accepting poor data quality is, as we saw, an important contributing factor to the metrics death spiral.

That being said, data driven organizations also understand that optimal data quality is a journey that is never complete, and that needs to be traveled incrementally. Sub-optimal data quality does not imply that you cannot start the journey towards a better data driven culture. Too often have I heard organizations throwing their arms up and saying they’d love to be data driven, but simply cannot trust their data. Start small, call out bad data when you see it, and caveat any decision that is made by using questionable data, using the process itself as an excuse to prioritize improving the quality of the data. And, as underlined in the previous section, remember that the more data and metrics you maintain, the harder it is to guarantee their quality.

### Data accessibility

Everyone in a data driven organization should be able to access the data and metrics they need to make the optimal decisions. This means that we should give maximum visibility to metrics and data (taking into account of course possible privacy, security, and confidentiality constraints).

Access to data or metrics without context, and a minimum level of preparation might be counterproductive though (see example below). As a rule of thumb, an individual in a data driven organization should probably understand and care about the metrics that are directly connected to their team goals. This is probably a combination of a small subset of metrics at the appropriate level in the metrics hierarchy, and the top level metrics that are important for everyone.

![](/blog/images/91-03.jpeg)

## Appendix: The right tool for the job

As alluded to in the sections above, there are different tools for being data-driven. Those tools have pros and cons. This appendix is a brief summary with some rules of thumb of how to decide what tool to use. Of course, the answer might be more nuanced and complicated, but treat these rules of thumb as a good starting point.

The “data-driven grade” is a relative metric that attempts to quantify how much the tool enables or facilitates the tenets above.

Dashboards

Use when: There is an actionable metric that you need to keep track of and need to look at frequently (e.g. more than weekly).
Data-driven grade: C
Why: While end users tend to love dashboards because they are flexible and easy to use, they hardly ever lead to good data driven decisions, they are costly to maintain, and can easily lead to metrics death spiral.

Ad hoc Data Report

Use when: There is an actionable metric that you need to understand at a point in time to make a decision, but you do not anticipate having to track over time.
Data-driven grade: A
Why: While ad hoc reports are not generally flexible and require work from specialists, they can be a good tool for analyzing situations without generating unnecessary complexity and maintenance.

Periodic Report

Use when: There is an actionable metric that you need to understand periodically to make a decision, and anticipate having to track every now and then to revise the decision..
Data-driven grade: A-
Why: A periodical data report is simply a bit more cumbersome to maintain than an ad hoc report. However, it is a great tool to highlight trends and changes in the metrics that are already part of the hierarchy

A/B Test

Use when: You are making a change to the product/service for which you need a reliable response on whether the hypothesis is correct. Importantly, only to be used if the volume of data that will be gathered allows for the result to be significant (note that this depends on things like the duration of the experiment, but also whether you are measuring differences upstream of the funnel or deep in the product).
Data-driven grade: A+
Why: A/B tests promote hypothesis driven development and are a key component of most data driven organizations. They are costly to set up, and generally not very flexible for the end user who receives the results. However, they get us close to understanding the “why” and therefore enable us to make good data driven decisions.

Tracking metrics change due to a product launch

Use when: You want to understand the impact of a product change, but cannot afford to do an A/B test because you cannot meet necessary criteria (e.g. amount of data) or doing that could be too complex or costly. Note that you can track a product launch related metric using any of the first three tools described (dashboard, ad-hoc report, or periodic report). The pros/cons will depend on this choice.
Data-driven grade: B-

Others

There are other tools for data driven decision making that are beyond the scope of this post. For example, for experiments, besides A/B testing, you can use Quasi-experiments, interleaved experiments, or multi-armed bandits. If there is interest, and I have time, I might get to them in a second part.
```
---
id: 92
title: "An unnecessarily long and personal recount of a 70.3 Ironman in Andorra"
date: '2022-08-21T00:00:00+00:00'
author: xamat
permalink: /andorra
image: /blog/images/IronmanMedal.jpg
header:
    teaser: https://amatriain.net/blog/images/IronmanMedal.jpg
reading_time:
    - ''
    - ''
categories:
    - running

---
```

I have not written about my running/sports feats in a while. On the one hand I don’t think I have anything more interesting to say than what I said in my [10 Lessons I learned from running](https://amatriain.net/blog/10-little-lessons-for-life-that-i/). On the other hand, I usually prefer to answer [Quora questions](https://www.quora.com/profile/Xavier-Amatriain/answers) about running and triathlons rather than writing a post about “about me”. But, given that I am on vacation, and promised myself not to “work” too much, and given that my last “adventure” was somewhat epic, I thought it was a good occasion to go back to writing about this. Also, fwiw, this has somewhat of a therapeutic effect and distracts me from all the work I do in my day job.

First, a bit of context. Even though I got a late start in races, and I ran my first one in 2011, 11 years later, I have ran 15 Marathons, a few ultras (up to 100k), and one full Ironman.

<img src="/blog/images/IronmanMedal.jpg" width="300" height="200">

This [Half IM in Andorra](https://www.ironman.com/im703-andorra) was my fifth. This was not a “normal” half IM though. In fact, the organizers, who we ended up becoming pretty close with, advertise it as the “hardest half IM in the world”. Why? Well, to start with, the whole event is in altitude (In case you have not heard about [Andorra](https://en.wikipedia.org/wiki/Andorra) before, it is a so-called microstate in the

Pyrinees). [The swim](https://www.ironman.com/im703-andorra-course) happens in the [Engolasters Lake](https://en.wikipedia.org/wiki/Lake_Engolasters), at 1,600m (5k ft) of elevation. The rest of the event is also at high altitude. Not only that, the 90 km's of the bike course are brutal with 2,000m (6,500 ft) of elevation. During the ride you climb the "famous" [Ordino](https://www.bikecation.co.uk/climbs/coll-d-ordino.aspx) port twice, and that is not even the hardest part.

<img src="/blog/images/Location_Andorra_Europe.png" width="300">

[By Bosonic dressing - Own work, CC BY-SA 3.0](https://commons.wikimedia.org/w/index.php?curid=7163527)

You get the gist of it. This is no ordinary Half Ironman. In fact, I would argue that in order to finish it reasonably well, you need to be in shape to finish almost a standard Full Ironman. In fact, as I expected, [SPOILER ALERT] my final time was by far the worst of all the Half IMs I have done in the past. Now, importantly, I have done some tough races in the past: my first 70.3 was in [Donner Lake](https://donnerlaketri.com/), also in altitude and with elevation gain. I have run the [Triple Tahoe Marathon](https://www.laketahoemarathon.com/triple) in high altitude. I have done the [Grand Canyon R2R2R](https://www.trailrunproject.com/trail/7011064/rim-to-rim-to-rim-r2r2r)… and so on. Because of this, I was confident I would finish this one. A bit too confident maybe, as I struggled through this one quite a bit.

Before we move on, I should introduce some of the players. To start with, and above all, I should introduce my partners in crime: Joaquin and Alberto. Joaquin and Alberto are my long-standing training buddies (and friends). While we started about 10 years ago with a much larger group, most of the others dropped. A few because they moved. Others because they picked up the pace, and are now too fast for us. Many because they are no longer running. Alberto and Joaquin have been constantly by my side in most of the adventures. They train with me at least once if not twice weekly, and I would not be here without them. Then there is Andres, an old training buddy of ours who moved and now lives in Madrid. While he doesn't always sign up, he does come to our virtual Zwift meetups, and he agreed to do this adventure in a relay team. Last, but not least, is my sister Estibaliz. She lives near Barcelona. In fact, she is the captain of the Olympic Port in Barcelona. She is also a strong long distance swimmer, and more recently has gotten into running, and even some cycling. She was part of the Andorra 70.3 organization. She was in charge of the swimming portion, and ended up introducing us to most of the main organizers.

<img src="/blog/images/Relay2013.JPG" width="300" height="200">
<img src="/blog/images/TrainingWithAlbertoAndJoaquin.jpg" width="300" height="200">

# The preparation

I don't run as many races as I did before COVID, but I have definitely kept pretty active. I did the Santa Cruz 70.3 in 2021 in preparation for the California full IM that ended up being canceled because of the weather in October 2021. That was despite having done most of my biking indoors mostly on Zwift. In 2022, after I got a special invitation for this 70.3 in Andorra from the organizers (that included my sister), I did not up my mileage much. In 2021 I did a total of almost 7,000 km and 472 hours of training. In the 7 months of 2021 I have trained 273 hours and run/biked about 4,000 km.

Because we had done most of our bike training indoor, and we were expecting the bike portion to be brutal, we did a few long outdoor rides that mimicked the terrain as well as possible including the wonderful loop going to the Pacific from the Bay Area up Old La Honda, and back through Tunitas Creek, as well as the [Mount Umunhum climb](http://www.toughascent.com/blog/tag/mt-umunhum-road/), which is in my "backyard".

We did not prepare the swimming part too much, which, as we will see, was probably a mistake. However, in case you are not familiar with Ironmans, it is important to note that the swimming part is the least important of the three. If you are REALLY good at swimming, you can maybe beat mediocre swimmers by 10-15 minutes. In the bike, the difference can be 4 times larger or more. I did all my swimming in the local pool. I now count 15 1 hour long swimming sessions in 2022.

Regarding the running, that is usually my strong suit, and I did prepare for it as usual: train to be in shape for a good full marathon at least a couple of months before the IM. In fact, we did run the [Big Sur Marathon](https://www.bigsurmarathon.org/) in April, and I improved my PR from 6 years ago, beating it only by a few seconds. I did run another race the week after the marathon: the popular local ["The Great Race"](https://raceroster.com/events/2022/57933/2022-rotary-club-of-los-gatos-great-race). Running a fast short race immediately after a marathon, and in preparation for a 70.3 is pretty dumb, and would definitely not recommend it. Finally, I had also signed up for a [100k ultra](http://www.quicksilver-running.com/Quicksilver-100k/50k) three weeks after the marathon. I ended up running only about 65k. The weather was very hot in the Bay Area that day, and I worried that if I pushed forward until the end I would end up injured, and miss the 70.3.

<img src="/blog/images/BigSur2022.jpeg" width="200">
<img src="/blog/images/TheGreatRace.jpg" width="400" height="300">
<img src="/blog/images/Quicksilver100k2022.jpg" width="200">
<img src="/blog/images/Zwift.jpg" width="350" height="250">

# The days before

We all got to Andorra on different days. Alberto, was there for the whole week sizing up the situation, and sending us reports from the ground. But, by Thursday night, we were all there and ready to go (minus some jet lag for some, and a recent COVID situation for Joaquin).

On Friday, we went up the Ordino port with the bikes. It is a beautiful climb, and we enjoyed every bit of it. Plus, we all got used to the rented bikes (as a side note, mine was so good that it reminded me that I should probably get a new one at home).

<img src="/blog/images/Ordino.jpg" width="300" height="200">

On Saturday, we went up to the lake to drop the bikes and to “test the waters”. Water temperature had been a concern all along since we had heard that on Thursday the temperature was as low as 14C (X F). However, the organizers had told us that the had closed the gates of the lake and that would make the water temperature increase significantly. Interestingly, this also came at a huge cost for the Andorra government since they supposedly lost about $40k a day in lost electricity production. In any case, by Saturday the water was already 17C (62F), and it was clear that water temperature would not be an issue. In fact, I strongly considered using my sleeveless wetsuit for the race.

<img src="/blog/images/PracticingSwim.jpg" width="400">

During these days, I became somewhat of a celebrity. The fact that I was coming all the way from California to this small country in the Pirenees, and the fact that I was well connected with the organizers, got me to do a couple of media interviews and I ended up making it into a local radio program and the Sport newspaper, the goto sports newspaper in Barcelona when I was growing up.

Everything seemed ready and set, although a last piece of concern that loomed upon us those days was the heat. Europe has experienced a pretty terrible heatwave this summer, and Andorra has not been any different. The expected temperature by the time we would start the run in downtown Andorra on Sunday was over 90F. That’s really hot, but, again, we are no strangers to running in those temperatures. In fact, the first 70.3 we did in Donner Lake also ended up being extremely hot, and, to make matters worse, under the smoke of nearby fires in Yosemite.

# The race

## The swim

A bit more about the swimming course: it happens in the Engolasters lake at 5000 ft of altitude. The lake is small, so you can’t really get the 1.2 miles in a single loop. So, we had to do two loops to the lake and do an “Australian exit” twice. This means you have to get out of the water, run/walk under the timing device, and get back into the water.

As mentioned before, I was not well prepared for the swim, but I was pretty confident anyways. I have swam in altitude before, I have swam in colder water, I have swam longer…, hell, I have even done the [Escape from Alcatraz](https://www.escapealcatraztri.com/). What could go wrong?

Well, whatever could go wrong went VERY wrong. I struggled to do the most basic form of swimming for at least one third of the distance. Before I exited the first time my pace was a horrendous 3:30 min/100m. My usual time is around 2 min. During that time I experienced the closest thing to what I can imagine a panic attack being. I could not breathe. I could not give more than a couple of strokes before stopping and getting all my head out of the water. And, of course, I was drinking a lot of fresh mountain water.

Now, both my other friends struggled as much as I did, with one of them deciding to swim the full course backstroke (a last resource we had heard from my sister who is a strong open water swimmer and was in charge of the water section of the race). So, clearly it was not just a ME thing. But, what really happened? Our best bet is the altitude, combined with the stress from the race, made it really hard to get our breadth under control. Now again, we are no novices on this, so it was definitely something that caught us completely by surprise. Next time, if there is a next time, I will make sure to spend a lot more time in the water the days prior in the same conditions of the race.

In any case, my final swimming time was a horrible 52:46. My expected time was around 35 or 40 minutes, so I did end up “losing” over 10 minutes in the water. But, I made it out. So, now everything should be downhill from here. Well, not really, since we had to climb the almost 7,000 feet of elevation gain in the bike.

<img src="/blog/images/IronmanSwim.JPG" width="300" height="200">

## The ride

Now, the path to the bike wasn’t easy either. We had to run around half a kilometer, the first part in the wetsuit, the second one in the cycling cloth (no cleats) pushing the bike from the forest to the road. By the time I got my bike on the road, I was somewhat dizzy and tired of pushing it uphill. Plus, I did not remember that I was using a rental bike, with different pedals. There is a pathetic video of me trying to get on the bike. By looking at it you would say that it was the first time I was riding at all (LOL).

The bike ride starts with a pretty long and dangerous downhill. I have to say that I am horrible at downhills. I have never been great, but ever since I had a pretty bad bike accident about 4 years ago, I have become even worse since I am not very comfortable going fast. That ends up making a difference in this course. My fastest speed in the course was 63 km/h. Some of the other riders I know were able to hit almost 90 km/h.

Besides that, I enjoyed the brutal ride quite a lot. I loved the two climbs to Ordino, and ended up suffering on those “other climbs” that were not “on the map”. For example, after the second climb to Ordino, everything is supposed to be downhill. However, you do all of the sudden find some climbs over 5% that can break your legs.

<img src="/blog/images/AndorraRideTotals.png" width="200">

I ended up doing the ride in 4:44. This is a horrible time if you consider that on a “normal” 70.3 my goal is always to do under 3 hours. I did end up thinking that I could have gone a bit harder, given that I did the uphill segments at around 200W (for an FTP of 360W that’s pretty low). However, given that I was expecting a pretty tough half marathon in the heat, that was probably as hard as I should have gone.

<img src="/blog/images/IronmanBike.JPG" width="300" height="200">

## The run

And boy did I struggle through the run. By that time of the day we were already hitting 90F (30C). And, remember, we were still at altitude. As I mentioned above, I had run a similar 70.3 run in my first one, when we ran in similar conditions in Tahoe. So, I knew what to do, plow through it, one step at a time.

Fortunately, the organizers had prepared the course for the weather conditions. There were ice buckets and hoses in every aid station. I don’t know that I could have finished without this.

Another very unfortunate thing happened during the run. Halfway through, a guy on a bike got in my way in the middle of the race path. I almost tripped over his bike and yelled at him. He came after me on his bike and told me he was a course referee and that he was coming after me because I was wearing headphones. What the actual f**k! I told him to go f himself. I was very annoyed he had gotten in the way, and he was now making the excuse of the headphone. While I continued my run I explained to the guy that I was wearing an “open ear” headset, which I knew for a fact was allowed according to their own race rules. The guy wouldn’t have it and harassed me for a few miles until I literally thanked him for doing so: the “distraction” was actually helping me run faster and not feel the pain. Believe or not, the asshole went on to disqualify me after the race, despite the fact he did not even show me a yellow card at that time, violating yet again, the race rules. I was able to revoke the decision easily because I “know people at high places”, but this was my worst ever experience in a race.

<img src="/blog/images/IronmanRun.JPG" width="200">

## The finish

The finish made everything worth it. My sister was waiting for me with the medal. I don’t get to see my sister much since I left Spain. Spending a few days with her and share this moment was very special for both of us.

<img src="/blog/images/IronmanFinish.JPG" width="200">

While the day after I felt so good that I thought I should have pushed more, the truth is that I was exhausted when I finished. My legs felt ok, but I was pretty dizzy and could barely walk. When I think about that feeling, I don’t think there was much more in me that day.

# Meta thoughts

## More on the silly Ironman rules

Ironman is a franchise brand that has turned into an international success by selling regular amateur triathletes the illusion that they are competing “almost professionally”. The truth is that you are sharing the road with world-class athletes, but the same is true in most popular marathons and even ultras. However, Ironman takes this a step further by allocating a few qualifying slots for the world championship for every age/gender group. Because many folks who qualify are not interested in it, getting the qualifying slot is relatively easy. I qualified for the 2022 World Championship in St. George Utah. My poor performance was still below the qualifying time of 8:30h.

All this “qualifying circus” implies that IM races have to impose a set of rules that are stupid for anyone who is not competing for a living. Those include e.g. a rule that forbids cyclists from “drafting” (riding behind another rider). This rule is absolutely stupid for most part, and, for what is worth, I have only seen professionals trying to break it by drafting in places where they know the judges will have no visibility (e.g. the tunnels in this Andorra 70.3).

Besides the headset incident, I had a couple more “situations” that can give you more insight into what I am talking about. In one of them, someone from the organization called me out for having removed my bike helmet strap AFTER dismounting the bike and while walking the bike to the transition area. As crazy and unexplainable as it might seem, there is a rule in this case that does forbid competitors to untie their helmet’s strap before they leave their bikes in the specified location in the T2 transition. You can be disqualified for that. Nuts. More on the Ironman rules [here]
(https://cdn2.sportngin.com/attachments/document/5326-2664364/2022_IRONMAN_Competition_Rules_-_English_Version_-_30march2022_-_FINAL_.pdf#_ga=2.218352282.2045724340.1648425032-260277762.1569464052).

## Nutrition

Understanding nutrition is a huge part of endurance events. This is true of marathons, even more true of ultras, and even more so of ironmans. The rule of thumb is that the longer an event is, the more you will be limited by your ability to endure while ingesting the right amount of nutrients. You can be in the greatest of shapes, but if the event is going to last, say, 12 hours, you better understand how to drink and eat during the event. You won't make it otherwise.

Why are Ironman's even harder than ultramarathons in this respect? The reason is that there is an added difficulty added to the the duration of the event: the different disciplines. You cannot eat and drink the same while swimming, riding a bike, and running. There are books written on this topic, but just to summarize, you need to ingest most of your calories (around XX an hour) during the bike, and keep a reasonable balance between carbs (aka sugar), calories, electrolytes, and water.

In my only full IM, and despite having done several ultras before, I did not do things right. I ran half of the marathon with pretty strong stomach cramps, had to stop several times in the porta potties, and struggled to the end. My conclusion is that I did not drink enough for the carbs that I ingested. But, who knows…

I usually don't struggle at all with nutrition in the 70.3s that I have done. I can push through 6 hours of vigorous exercise mostly on gels, water with electrolytes, and an occasional nutrition bar. However, as already explained, this 70.3 was different. For starters, I exercised for over 8 hours. The temperature was also brutal, particularly during the run. And, we were doing all of that at altitude. I might have benefitted from "packing" a bit more nutrition and water during the bike, but all in all I thought my nutrition was pretty good. What I did:

* During the bike:
   * 1 Huma gel every hour
   * 3 Tahoe nutrition bars
   * 2 bottles of water + Nuun electrolytes
   * 2 bottles of water + Tailwind
   * Occasional water only in some aid stations
* During the run:
   * 1 Huma gel
   * 1 bottle of water + Tailwind
   * Water in all aid stations
   * Some orange slices in the aid stations

## What I like/dislike about Ironmans

What I like:

* They are endurance events, and I like endurance events in general
* Having to prepare 3 disciplines forces you to get in really good shape
* Ironmans are generally well organized, and located in very nice locations (this one was spectacular, and I would totally recommend it)
* Ironmans have good services. E.g. the rental bike service in this one was excellent

What I dislike:

* Ironman is a franchise, and therefore a business. That shows in many different ways. As an example, the organization was [sued](https://www.triathlete.com/culture/news/class-action-lawsuit-filed-against-ironman-over-lack-of-refunds/) on how they handled COVID-related refunds (although, in their defense, the judge ruled in their favor).
* The stupid rules (see above)
* While I like cycling, I don't like that cyclist are so obsessed with their bikes. Most IMs spend thousands of $$ on their machines. Plus, they then have to ship them around to all the events. Compare that to running, where you only need to remember to grab your shoes.
* Somewhat related to the above, I enjoy the vibes in ultras much more than in Ironmans. Ultras feel like a community of outcasts getting together to hang out and go on a long hike. Ironmans feel like what happens when you put together a bunch of type A rich folks and ask them to show off. (Not lost in irony the fact that I might fit that description pretty accurately).
* The swim. I have really tried to like swimming. At this point, I enjoy a nice open water swim on my own or with friends. I even enjoyed some of the most epic swims I have done, in particular the Escape from Alcatraz. But, this latest experience was awful, and it makes me wonder about how much I want to do it again (the answer is probably yes, but we'll see).

# The aftermath

The days after the Andorra 70.3 I felt very well. So much, that I continued doing almost my usual training. While I did slow my pace a bit, I continued running around 6 miles a day, and doing occasional biking. That was obviously a mistake, one that I have done many times before. No matter how well you feel after a race, you need to give your body a rest. The end result was that I had a hamstring strain a few days later that forced me to stop for a little while.

<img src="/blog/images/RunningBarcelona.jpg" width="300" height="200">

At this point, a few weeks after the event, my hamstring is totally ok, but I am still struggling with a minor achilles tendon issue that will likely take some time to heal. Make sure to recover after any event!

# What now?

While I mentioned above that I was unlikely to do more Ironmans, I already have two on the horizon. The first one is the California 140.6 IM that was postponed last year because of the rain, and the year prior because of COVID. This race is in October, and as I get closer and closer to it, I feel I am not ready. To be clear, the Andorra 70.3 was part of the training plan. And, I think that if I was able to finish the Andorra race, I should be able to finish (even if struggling) to the end of a reasonably flat 140.6 like the California one. That being said, the recent injuries are somewhat worrisome. If I am not able to hit a good training pace and e.g. do an 8 hour training day combining biking and running in the next few weeks, I will end up dropping.

The other IM that I have “been volunteered” for by my sister and other organizers is the [140.6 in Vitoria, Spain, next July](https://www.ironman.com/im-vitoria). That would be a very nice one. It would be in my country, and right after my 50th birthday. However, my friends have other proposals that include (oh no!) doing the Andorra 70.3 again. We’ll see how this all shakes out.

Besides, as I hinted to above, at this point I’d rather do different kinds of events. I love running and ultras, so that would be my first choice. I would love to run a marathon under 3 hours, but I think it is too late for me at this point. Running is also a hard sport that is more prone to injuries. Biking is nice in that regard, but is dangerous to train outdoors, and requires you to invest quite a lot in your equipment. Some multi day biking events such as the [Haute Route](https://www.hauteroute.org/) are “being considered”.

What I am pretty sure is that this one won’t be the last thing I do. If you have any suggestions, feel free to send my way. And, if you want to see what I am up to in my sports, please follow me on [Strava](https://www.strava.com/athletes/370350).

```
---
id: 93
title: "A new chapter"
date: '2022-10-11T00:00:00+00:00'
author: Xmas
permalink: /newchapter
image: /blog/images/NewChapter1.jpg
header:
    teaser: https://amatriain.net/blog/images/NewChapter1.jpg
reading_time:
    - ''
    - ''
categories:
    - personal
    - career
    - curai

---
```

(Read my cofounder Neal Khosla's version in [Curai's blog](https://medium.com/curai/making-bets-at-curai-30609d84d16f))

After more than 5 years as the CTO of [Curai](https://www.curaihealth.com/), the healthcare startup I co-founded, I am taking a step back and transitioning to an advisory role while remaining on the board of directors. A few days back I let my colleagues know that I will be moving on from my full time day-to-day role at the company, and moving on to my next professional chapter.

As someone who is almost 50 years old and has moved across oceans a couple of times for work, this is obviously not my first transition. However, after spending more than five years building something wonderful from scratch, this one is particularly heartfelt.

# The Curai story

I must have told the story of why I started Curai hundreds of times since that is something that I have been asked by most candidates I have interviewed. It might be good to write it here “for the record”:

I got connected to my co-founder and now CEO Neal through his dad, who I had met a couple of times before. Neal told me about a vague idea of using AI to make healthcare very accessible and high quality for everyone, at a fraction of the cost of what was possible without technology. I immediately thought that was a great mission. The mission was the first thing that attracted me, and it still does today. Could we use technology (and particularly AI) to help millions of people get the care they needed, when they needed it? Besides the mission, the challenge itself also attracted me. I love challenges. However, I also love to address challenges that I think I have a reasonable answer for, and that took some time for me to figure out. The AI component was there: even six years ago it was easy to see that AI was not only progressing very quickly, but that the rate of innovation was accelerating exponentially. I was convinced (and I still am) that AI will revolutionize healthcare completely. A bigger concern for me was the data: where would we get data from to bootstrap our algorithms? Over months of conversations with Neal, I realized that we could get access to very unique partnerships to "cold-start the system". We were able to get exclusive access to knowledge bases from Massachusetts General Hospital and Vanderbilt University, and we also partnered with institutions like Stanford Healthcare to access their Electronic Health Records. When all those pieces fell together, I realized this was a once in a lifetime opportunity: an amazing mission, a big challenge, and a big opportunity. If I ever was to start a company, it **had to** be Curai.

<img src="/blog/images/NewChapter1.jpg" width="300" height="200">
<img src="/blog/images/NewChapter2.jpg" width="300" height="200">

We started in a temporary office space where we brought in our first engineer (Viggy) and we started doing what startups do: writing some throwaway code (particularly mine), and hiring some more people onto the team. We soon added Anitha, who now leads our AI/ML efforts, straight from Facebook AI Research, and Geoff, an amazing doctor and software engineer who leads our Clinical Innovation and Informatics efforts and still practices at Stanford. Curai was born! (To be historically accurate, Curai was not our name at that time, but I will spare you from the original name… insert blushing face emoji here).

<img src="/blog/images/NewChapter3.png" width="300" height="200">

# What we built

We have accomplished so many things since then! Many of those things were not in our initial roadmap for sure. We were not thinking about developing an entire end-to-end health system that is operational nationwide. We did not plan to hire an amazing care team of licensed physicians and other team members distributed across different continents. We were not thinking that we would ever acquire another startup! Or that we would build our own provider operating system from scratch. All of that has been possible thanks to many incredible team members and a lot of grinding along the way.

<img src="/blog/images/NewChapter4.jpg" width="300" height="200">
<img src="/blog/images/NewChapter5.jpg" width="300" height="200">

At this point, I am proud and confident to say that our tech is not only state-of-the-art but also way ahead of anyone else working in this space. How do I know? Well, we hear that every time we show our service and demos to people with deep knowledge of the health tech industry. Even [Dr. Bob Wachter](https://en.wikipedia.org/wiki/Robert_M._Wachter), author of [The Digital Doctor](https://www.amazon.com/dp/0071849467) and member of our medical advisory board, said in a recent meeting: *"It's really impressive. The tool that you showed us is really cool. I haven't seen anything nearly that good in the virtual primary care, app-based care space."* That was a highlight of my time at Curai, not only for the small role I played in putting the tech together but also because I was the one putting together our amazing [medical advisory board](https://medium.com/curai/curais-medical-advisory-board-f48b1e0ce549) and convincing Dr. Wachter to join.

<img src="/blog/images/NewChapter6.png" width="500" height="400">

It is hard to pick just a few technical milestones to highlight here, but one that definitely makes the list would be the moment when we surpassed typical human primary care physician diagnosis accuracy on a public dataset. We have also deployed a few of those algorithms into production, such as our "next question" algorithm that suggests optimal questions to ask to a patient given what we know about them in their history and current encounter. This algorithm does not only "suggest" questions to the providers, but also directly steps in to point out whenever conditions are met. Finally, it is also worth mentioning that our research playground is even more impressive than the technology that we have rolled out into the product. Our prototype is an end-to-end AI driven medical solution that drives the entire patient workflow from history gathering to prescription recommendations. We expect everything that is in this prototype to make it to the product in the next few quarters. It has been highly rewarding to watch the future of personalized AI medicine develop before my eyes.

<img src="/blog/images/NewChapter7.png" width="500" height="400">

Something that **was** on our original roadmap, and that has driven many of the features I mentioned, is our research into state of the art medical AI. I started my career as a researcher after my Ph.D., and I view using research and publications as an effective way to inject an open innovation culture and attract talent. I am very happy that since 2017 [we have published](https://medium.com/curai-tech/research-publications-at-curai-ee22a820b807) more than a dozen peer-reviewed papers in areas such as medical diagnosis algorithms, NLP, and medical image classification.

<img src="/blog/images/NewChapter8.jpg" width="300" height="200">
<img src="/blog/images/NewChapter9.jpg" width="300" height="200">

And while I feel very proud of the service, the technology, and the research, I even feel more proud of the people we have attracted at Curai and the culture we have managed to set across the company. Attracting talent across diverse disciplines like clinical operations, product, engineering, and ML research is no small feat. Striking the right balance between growing and pushing people to their best performance while being supportive and understanding, particularly during a pandemic and a move to fully remote, is not easy either. While we have made our fair share of mistakes (oh, yes, there were lots of them), I feel very proud of where I will leave the team and the culture. We grinded and delivered and had fun all the while (I fondly remember the talent shows, the many remote games we played, the group runs on the Bayland Loop Trail, and, of course, the Las Vegas offsite).

<img src="/blog/images/NewChapter10.jpg" width="300" height="200">
<img src="/blog/images/NewChapter11.jpg" width="300" height="200">

As a final fun fact, I started a “to-do” document on the first day of Curai. That document is simply a bullet list of stuff that I simply struck through and moved down to a “done” section as I went through them. As of this writing, the document is 145 pages long, and most (but not all) of the items are stricken through!

# Curai’s next chapter

As I take this step back, I could not leave Curai in better hands. Neal has been a great partner, and is the perfect CEO for Curai given his deep understanding of everything from the technology to the healthcare business and industry. Meghan is an awesome CPO who leads not only the product, but also our operations team. She comes with an amazing healthcare product background and also with a great (Canadian) attitude. Similarly, Nicole, who joined more recently from Amazon, is the perfect Chief Business Officer. A lot of the upcoming commercial success Curai is going to have in the next few months is due to her talent and hard work. And Ally, who leads our People team, has been an incredible force behind our culture and operations. We wouldn’t be here without her and her team. There are so many others to include in the list of people to thank, but in an effort to be succinct, I want to at least mention some of the other very early folks that made Curai’s foundation so special; Jenae, who has been with us almost since the beginning, and has played an immense role in our success, Jen-Jen, another of those amazing minds who studied Data Science after becoming a Pediatrician and now leads our Clinical Product efforts, and there is Jack, who I hired as a Data Scientist coming from the Oil industry and now leads our Automation Product efforts. Also, as I am ramping down from Curai, we are just confirming Davis as our first ever full-time Chief Medical Officer. He will be leading our physician team and clinical innovation. He is taking the reins of a strong team that was built and led by our amazing Clinical Director Yvonne; together they will take our clinical operations to world-class status. I have known and followed Davis’ career for a long time, and I am convinced he is going to be hugely impactful at Curai.

You might have noticed even from the names and the backgrounds that there is a very high level of diversity in our executive team, and that is another aspect I am very proud of. Over 60% of Curai employees are women, and that number is also over 40% in my Technology department.

As we looked at who would take my place in this new chapter for Curai, it was clear to us that we had enough internal talent to fill open positions from within the company. Viggy, who as I mentioned has been building the tech since day one, will lead engineering; and Anitha, who has been not only “the brain” but also the one putting the work behind most of our AI and ML, will be leading the ML team. I am 100% positive that they are ready for this next challenge, and they will take our Curai technology to the next level.

So, as I move on to my next chapter for personal reasons, I could not be more thankful for the opportunity to be part of Curai’s history. I am bullish on what is to come for the company, and I am committed to continue helping from the sidelines as an advisor and member of the board of directors. I will also be cheering the many successes that are to come very soon and will continue to recommend Curai to everyone I know!

<img src="/blog/images/NewChapter12.jpg" width="500" height="400">
```
---
id: 94
title: "ChatGPT: hype or revolution"
date: '2022-12-11T00:00:00+00:00'
author: Xmas
permalink: /chatGPT
image: /blog/images/ChatGPT1.png
header:
    teaser: https://amatriain.net/blog/images/ChatGPT1.png
reading_time:
    - ''
    - ''
categories:
    - personal
    - career
    - curai

---
```

“We tend to overestimate the effect of a technology in the short run and underestimate the effect in the long run.” - Amara’s law

[ChatGPT](https://openai.com/blog/chatgpt/) was released by OpenAI a couple of weeks back. In a few days,

this next generation chatbot
has caused an incredible and mostly unexpected impact. Responses range from “[ChatGPT will kill Google and revolutionize the world](https://twitter.com/jdjkelly/status/1598021488795586561)” to
“[ChatGPT is not only bad, but dangerous](https://medium.com/@guruduth.banavar/chatgpts-deep-fake-text-generation-is-a-threat-to-evidence-based-discourse-c096164207e0)” .
But, how much of it is hype, reality, or fear mongering? Let me try to go over some of these questions. Inevitably, the answer will point us to Amara’s law.

# What is ChatGPT?

ChatGPT is a fine tuned version of [GPT-3](https://amatriain.net/blog/transformer-models-an-introduction-and-catalog-2d1e9039f376/#GPT3).
GPT-3 is an LLM (Large Language Model) developed by OpenAI and published in May 2020. It is a decoder-only Transformer with 175B parameters.
To be more precise, ChatGPT is a fine tuned version of the so-called GPT-3.5, a family of models that derives from the so-called [code-davinci-002](https://beta.openai.com/docs/model-index-for-researchers).
Note that there is no publications explaining the differences between GPT-3 and GPT-3.5 models (except for the variations introduced in the
[InstructGPT paper](https://arxiv.org/abs/2203.02155)). I won’t hypothesize here on what other improvements could have been introduced in recent models,
but anyone following the LLM literature closely will probably have good intuitions of what those might be.

# What can ChatGPT be used for?

ChatGPT has been explicitly designed and fine-tuned for chatbot applications. Because of this it can, for example, better track the state of a
conversation than previous models. Very importantly, ChatGPT has an easy-to-use interface that has been released to the public and has enabled
thousands of people to experiment with LLMs for the first time. In fact, ChatGPT surpassed all expectations by registering over 1 million users during
the first week. OpenAI had to close access to new users shortly after.

Although the main use of ChatGPT is, as mentioned above, open ended conversations, people have quickly found creative ways to use it. Here are some examples:

* [Answer StackOverflow questions](https://twitter.com/christophmolnar/status/1598250997939597312)
* [Replace Google](https://twitter.com/jdjkelly/status/1598021488795586561)
* [Generate cooking recipes](https://twitter.com/stephsmithio/status/1598920887029628928)
* [Solve complex programming tasks](https://twitter.com/goodside/status/1598129631609380864)
* [Generate prompts for Dall-e/StableDiffusion](https://twitter.com/guyp/status/1598020781065527296)
* [Build apps from scratch](https://twitter.com/packym/status/1598405769669771264)

It is important to keep in mind again, that ChatGPT was not optimized for any of those uses, nor was it considered to be general purpose. However, some of the results on those specific tasks have been pretty remarkable, which has given many people a peak into what might be coming soon.

# What are ChatGPT’s limitations?

ChatGPT, as all LLMs from its generation, are not great at factual information retrieval on their own. The problem is not that they
lack knowledge, but rather that they have been trained on all kinds of information, some of which can be very unreliable. LLMs such as
ChatGPT do not have a sense of source reliability or their own confidence on some specific information. Because of this they can easily,
and very confidently, hallucinate responses to questions by combining unreliable information they have been trained on. A way to understand
this is the following. Imagine a LLM like ChatGPT has been trained on the best medical literature, but also Reddit threads that discuss health
issues. The AI can sometimes respond by retrieving and referring to high quality information, but it can other times respond by using the completely
unreliable Reddit information. In fact, if the information is not available in the medical literature (e.g. a very rare condition), it is much more
likely it will make it up (aka hallucinate). In that case it will still “sound authoritative”, particularly because it is “generally correct”.

<img src="/blog/images/CharGPT2.png" width="300" height="200">

Because of the above, there have been plenty of examples of ChatGPT “being wrong”. This has pushed some renowned researchers and academics
to claim things such as ChatGPT being a [threat to evidence-based discourse](https://medium.com/@guruduth.banavar/chatgpts-deep-fake-text-generation-is-a-threat-to-evidence-based-discourse-c096164207e0).
They are not wrong, ChatGPT as such, and as Sam Altman explained,
is not designed for evidence-based discourse. In fact, ChatGPT is trained on data that is over a year old! So, at the very least it won’t have
evidence of recent events. ChatGPT does not have basic information such as what is the current day or where

you are located. If you ask a simple
question that Siri can answer such as what is the weather like today in San Francisco, it will acknowledge that it doesn’t know.

<img src="/blog/images/ChatGPT3.png" width="300" height="200">

It is important to understand that many of these issues can be solved, some more easily than others, of course. As an example,
access to current information such as weather in San Francisco is something that very basic systems such as Siri, Alexa or Google Assistant
have no problems with.

Other limitations require a bit more work, but we have reasons to believe that they can be addressed. Let’s look at another critique: ChatGPT
[does not know math](https://hariprasadblogs.medium.com/testing-chatgpt-in-mathematics-can-chatgpt-fe73c20b2c6d). There are plenty of examples of math mistakes by ChatGPT. This clearly means that ChatGPT on its own is pretty bad at math
(although it is fair to say that even with that [it gets]
(https://twitter.com/davidtsong/status/1598767389390573569) a 500 score in the math portion of
the SAT). But, hold on, it has not been designed for that purpose! The truth is that how much you can rely on ChatGPT’s response depends on your prompt.
You can in fact [teach a generic LLM math](https://arxiv.org/abs/2211.09066) by feeding algorithmic reasoning in the prompt (aka zero shot learning).

What about broader concerns about safety, trust, and truthfulness? There are also very promising avenues for solving those, even in a general sense.
Google’s [LaMDA](https://blog.google/technology/ai/lamda/) model already included some of those aspects.
More recently, Deepmind’s [Sparrow](https://www.deepmind.com/blog/building-safer-dialogue-agents)
includes a number of improvements in this area. It is very important to keep in mind that these improvements are still not perfect. However, one of the reasons for this is that they all attempt to provide safe/trustworthy general purpose dialog agents. However, in narrower domains, this is much easier to accomplish, even if by adding artificial rule-based guardrails. Yes, those will not scale to a general-purpose AI, but they can be extremely useful in a specific domain or application very soon.

# Conclusion

ChatGPT has taken the world by storm. Having such a high quality general-purpose chatbot available to anyone has
sparked our wildest imaginations, and there have been crazy applications in a few days. However, this LLM has very obviou
s limitations, particularly related to factualness and truthfulness. Those limitations have solutions, some easier than other'
s, that make me think that high quality and trustworthy LLMs are around the corner, particularly in narrow domains or specific applications.

As an end note I will say that I have been working with language models for quite some time now, particularly in the healthcare domain
(see e.g. [our publication](https://proceedings.mlr.press/v149/chintagunta21a.html) on using GPT3 for medical data labeling). In my end-of-year recap of AI in 2018 I already highlighted the importance of Language Models. I could not have foreseen such an incredibly fast development in this space. I am not at all dazzled by ChatGPT. I think it is a very incremental improvement over GPT3.5 and most of the hype is due to it being open to so many people. However, as always, we overestimate progress in the short term, and underestimate in the long term. In fact, I would replace “long term” with “mid term” in this last sentence. My strong prediction is that LLM applications (not ChatGPT) will revolutionize almost every digital vertical in the next 18 to 24 months. You better get ready for it!

```
---
id: 95
title: "Prompt Engineering 101: Introduction and resources"
date: '2023-01-04T00:00:00+00:00'
author: Xavier
permalink: /PromptEngineering
image: /blog/images/FatCrocodile.png
header:
    teaser: https://amatriain.net/blog/images/FatCrocodile.png
reading_time:
    - ''
    - ''
categories:
    - generativeAI
    - machine learning
    - AI
    - LLMs

---
```

# TOC

# 1. What is a prompt?

Generative AI models interface with the user through mostly textual input. You tell the model what to do through a textual interface, and the model tries to accomplish the task. What you tell the model to do in a broad sense is the prompt.

In the case of image generation AI models such as DALLE-2 or Stable Diffusion, the prompt is mainly a description of the image you want to generate.

<img src="/blog/images/FatCrocodile.png">

a fat crocodile with a gold crown on his head wearing a three piece suit, 4k, professional photography, studio lighting, LinkedIn profile picture, photorealistic

In the case of large language models (LLMs) such as GPT-3 or ChatGPT the prompt can contain anything from a simple question ("Who is the president of the US?") to a complicated problem with all kinds of data inserted in the prompt (note that you can even input a CSV file with raw data as part of the input). It can also be a vague statement such as "Tell me a joke. I am down today.".

Even more generally, in generative task oriented models such as Gato, the prompt can be extremely high level and define a task you need help with ("I need to organize a one week trip to Greece").

For the rest of this document, and for now, we will focus on the specific use case of prompts for LLMs.

# 2. Elements of a prompt

Generally speaking, and at a high level, a prompt can have any of the following:

1. Instructions
2. Question
3. Input data
4. Examples

# 3. Basic prompt examples

In order to obtain a result, either 1 or 2 must be present. Everything else is optional. Let's see a few examples (all of them using ChatGPT).

## Instructions + Question

Beyond asking a simple question, possibly the next level of sophistication in a prompt is to include some instructions on how the model should answer the question. Here I ask for advice on how to write a college essay, but also include instructions on the different aspects I am interested to hear about in the answer.

> *"How should I write my college admission essay? Give me suggestions about the different sections I should include, what tone I should use, and what expressions I should avoid."*

<img src="/blog/images/CollegeEssayAdvice.png" >

## Instructions + Input data

Continuing with the previous example, it does not escape anyone that if you can ask a LLM for advice on how to write an essay, you can also directly ask it to write the essay itself.

Note: I am not advocating for this to be an ethical use of these tools, but it is important to be aware that this possibility exists and is already being used by students around the world. It is beyond the scope of this introductory guide to discuss all the possible ethical, legal, or moral concerns that LLMs or generative AI as a whole introduces, but I thought it would be important to at least call out in an introductory example. The fact that you CAN do something with a generative model does not mean that it is the right thing to do! On the other hand, if you are on the receiving end, you better prepare yourself and your organization for all kinds of AI-generated content to come your way. Fortunately, for situations like the one outlined in this example, there are already efforts underway such as GPTZero to detect AI generated content.

Back to the example, let’s see what happens when I input some data about me and give some instructions:

> *“Given the following information about me, write a 4 paragraph college essay: I am originally from Barcelona, Spain. While my childhood had different traumatic events, such as the death of my father when I was only 6, I still think I had quite a happy childhood.. During my childhood, I changed schools very often, and attended all kinds of schools, from public schools to very religious private ones. One of the most “exotic” things I did during those years is to spend a full school year studying 6th grade in Twin Falls, Idaho, with my extended family.*
>
>I started working very early on. My first job, as an English teacher, was at age 13. After that, and throughout my studies, I worked as a teacher, waiter, and even construction worker.”*

<img src="/blog/images/XavierCollegeEssay.png">

## Question + Examples

You can also feed examples into a language model. In the example below I include some of the shows I like and don’t like to build a “cheap” recommender system. Note that while I added only a few shows, the length of this list is only limited by whatever token limit we might have in the LLM interface.

> *“Here are some examples of TV shows I really like: Breaking Bad, Peaky Blinders, The Bear. I did not like Ted Lasso. What other shows do you think I might like?”*

<img src="/blog/images/TVShows.png">

# 4. So, what is prompt engineering anyways?

Now that we know what a prompt is, and we have seen a few examples of it, let’s discuss what is prompt engineering.

Prompt engineering is a very recent but rapidly growing discipline that has the goal of designing the optimal prompt given a generative model and a goal. Prompt engineering is growing so quickly that many believe that it will replace other aspects of machine learning such as feature engineering or architecture engineering for large neural networks.

Prompt engineering requires some domain understanding to incorporate the goal into the prompt (e.g. by determining what good and bad outcomes should look like). It also requires understanding of the model. Different models will respond differently to the same kind of prompting.

Generating prompts at some scale requires a programmatic approach. At the most basic level you want to generate prompt templates that can be programmatically modified according to some dataset or context. As a basic example, if you had a database of people with a short blurb about them similar to the one used in the college essay above. The prompt template would become something like “Given the following information about [USER], write a 4 paragraph college essay: [USER_BLURB]“. And the programmatic approach to generating college letters for all users would look something like:

```python
    for user,blurb in students.items():
      prompt = “Given the following information about {}, write a 4 paragraph college essay: 
{}”.format(user, blurb)
      callGPT(prompt)
```

Finally, prompt engineering, as any engineering discipline, is iterative and requires some exploration in order to find the right solution. While this is not something that I have heard of, prompt engineering will require many of the same engineering processes as software engineering (e.g. version control, and regression testing).

# 5. Some more advanced prompt examples

It is important to note that given the different options to combine components and information in a prompt, you can get as creative as you want. Keep in mind that the response is stochastic and will be different every time. But, the more you constraint the model in one direction, the most likely you will get what you are looking for. Here are some interesting examples that illustrate the power of prompt engineering.

## Chain of thought prompting

In chain of thought prompting, we explicitly encourage the model to be factual/correct by forcing it to follow a series of steps in its “reasoning”.

In the following example, I use the prompt:

> “What European soccer team won the Champions League the year Barcelona hosted the Olympic games?
>
> Use this format:
>

```
>  Q: <repeat_question>
>  A: Let’s think step by step. <give_reasoning> Therefore, the answer is <final_answer>.”

<img src="/blog/images/BarcelonaOlympics.png">

I now ask ChatGPT to use the same format with a different question by using the prompt:

> *"What is the sum of the squares of the individual digits of the last year that Barcelona F.C. won the
Champions League? Use the format above."*

<img src="/blog/images/FCBarcelonaSumOfSquare.png">

## Encouraging the model to be factual through other means

One of the most important problems with generative models is that they are likely to hallucinate knowledge
that is not factual or is wrong. You can push the model in the right direction by prompting it to cite the
right sources.  (**Note:** I have seem examples of more obscure topics where sources are harder to find in
which this approach will not work since the LLM will again hallucinate non-existing sources if it can't find
them. So treat this with the appropriate care)

> *“Are mRNA vaccines safe? Answer only using reliable sources and cite those sources. “*

<img src="/blog/images/mRNAVaccines.png">

## Use the AI to correct itself

In the following example I first get ChatGPT to create a “questionable” article. I then ask the model to
correct it.

> *“Write a short article about how to find a job in tech. Include factually incorrect information.”*

<img src="/blog/images/JobInTech.png">

> *“Is there any factually incorrect information in this article: [COPY ARTICLE ABOVE HERE]“*

<img src="/blog/images/FactChecking.png">

## Generate different opinions

In the following example, I feed an article found online and ask ChatGPT to disagree with it. Note the use
of tags <begin> and <end> to guide the model.

> “The text between \<begin> and \<end> is an example article.
>
> \<begin>
> From personal assistants and recommender systems to self-driving cars and natural language processing,
machine learning applications have demonstrated remarkable capabilities to enhance human decision-making,
productivity and creativity in the last decade. However, machine learning is still far from reaching its
full potential, and faces a number of challenges when it comes to algorithmic design and implementation. As
the technology continues to advance and improve, here are some of the most exciting developments that could
occur in the next decade.
>
> 1. Data integration: One of the key developments that is anticipated in machine learning is the
integration of multiple modalities and domains of data, such as images, text and sensor data to create
richer and more robust representations of complex phenomena. For example, imagine a machine learning system
that can not only recognize faces, but also infer their emotions, intentions and personalities from their
facial expressions and gestures. Such a system could have immense applications in fields like customer
service, education and security. To achieve this level of multimodal and cross-domain understanding, machine
learning models will need to leverage advances in deep learning, representation learning and self-supervised
learning, as well as incorporate domain knowledge and common sense reasoning.
> 2. Democratization and accessibility: In the future, machine learning may become more readily available to
a wider set of users, many of whom will not need extensive technical expertise to understand how to use it.
Machine learning platforms may soon allow users to easily upload their data, select their objectives and
customize their models, without writing any code or worrying about the underlying infrastructure. This could
significantly lower the barriers to entry and adoption of machine learning, and empower users to solve their
own problems and generate their own insights.
> 3. Human-centric approaches: As machine learning systems grow smarter, they are also likely to become more
human-centric and socially-aware, not only performing tasks, but also interacting with and learning from
humans in adaptive ways. For instance, a machine learning system may not only be able to diagnose diseases,
but also communicate with patients, empathize with their concerns and provide personalized advice. Systems
like these could enhance the quality and efficiency of healthcare, as well as improve the well-being and
satisfaction of patients and providers
> \<end>
>
> Given that example article, write a similar article that disagrees with it. “
```

<img src="/blog/images/SecondOpinion.png">

## Keeping state + role playing

Language models themselves don’t keep track of state. However, applications such as ChatGPT implement the notion of “session” where the chatbot keeps track of state from one prompt to the next. This enables much more complex conversations to take place. Note that when using API calls this would involved keeping track of state on the application side.
In the example below, based on [a Tweet](https://twitter.com/goodside/status/1608701900735803394) by Scale’s Staff Prompt Engineer Riley Goodside, I make ChatGPT discuss worst-case time complexity of the bubble sort algorithm as if it were a rude Brooklyn taxi driver.

<img src="/blog/images/AngryBrooklyCab1.png">
<img src="/blog/images/AngryBrooklyCab2.png">
<img src="/blog/images/AngryBrooklyCab3.png">

## Teaching an algorithm in the prompt

The following example is taken from the appendix in Teaching Algorithmic Reasoning via In-context Learning where the definition of parity of a list is fed in an example.

> “The following is an example of how to compute parity for a list
> Q: What is the parity on the list a=[1, 1, 0, 1, 0]?
> A: We initialize s=
> a=[1, 1, 0, 1, 0]. The first element of a is 1 so b=1. s = s + b = 0 + 1 = 1. s=1.
> a=[1, 0, 1, 0]. The first element of a is 1 so b=1. s = s + b = 1 + 1 = 0. s=0.
> a=[0, 1, 0]. The first element of a is 0 so b=0. s = s + b = 0 + 0 = 0. s=0.
> a=[1, 0]. The first element of a is 1 so b=1. s = s + b = 0 + 1 = 1. s=1.
> a=[0]. The first element of a is 0 so b=0. s = s + b = 1 + 0 = 1. s=1.
> a=[] is empty. Since the list a is empty and we have s=1, the parity is 1
>
> Given that definition, what would be the parity of this other list b= [0, 1, 1, 0, 0, 0, 0, 0]”

<img src="/blog/images/ListParity.png">

# 6. Resources

## Videos

* [CMU Advanced NLP Course: Prompting](https://www.youtube.com/watch?v=5ef83Wljm-M) (2022)
* [Prompt Engineering 101: Autocomplete, Zero-shot, One-shot, and Few-shot prompting] (https://www.youtube.com/watch?v=v2gD8BHOaX4) (2022)

## Posts

* [The biggest bottleneck for large language model startups is UX] (https://medium.com/innovationendeavors/the-biggest-bottleneck-for-large-language-model-startups-is-ux-ef4500e4e786)
** Post about the broader UX implications of LLMs, with a section on prompting
* [Prompt injection attacks against GPT-3](https://simonwillison.net/2022/Sep/12/prompt-injection/)
** About prompt injections attacks, where the goal is to craft malicious inputs so that GPT-3 ignores previous directions

## Papers

* [Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing] (https://arxiv.org/abs/2107.13586) (2019)
** A bit dated (3 years old) survey of prompting. It includes a fairly reasonable taxonomy of prompting methods, but some of them are not very practical
* [Chain of Thought Prompting Elicits Reasoning in Large Language Models](https://arxiv.org/abs/2201.11903) (2022)
** Forcing the LLM to reason step by step by giving the right prompt improves results
* [Language Models are Zero Shot Reasoners](https://arxiv.org/abs/2205.11916) (2022)
** Fascinating paper that, as continuation of the previous, shows how LLMs reason better if you simply tell them to “reason step by step”
* [Teaching Algorithmic Reasoning via In-context Learning](https://arxiv.org/abs/2211.09066) (2022)
** More advanced prompting. In this case the authors show how you can prompt standard LLMs to do complex algorithmic computations given the right prompt. They also show how skills can not only be taught, but also composed in the prompt.
* [Ask Me Anything: A simple strategy for prompting language models] (https://github.com/HazyResearch/ama_prompting)
** Interesting approach to prompting in which instead of trying to come up with the perfect prompt at the input, the authors propose multiple imperfect input prompts and output aggregation through weak supervision

## Tools

```
* [Microsoft’s Prompt Engine](https://github.com/microsoft/prompt-engine)
** utility library for creating and maintaining prompts for Large Language Models
* [Ice](https://github.com/oughtinc/ice)
** Interactive Composition Explorer: a Python library for compositional language model programs

## Other lists of resources

* [DAIR’s Prompt Engineering Guide and resources](https://github.com/dair-ai/Prompt-Engineering-Guide)
* [Prompt engineering resources github repo](https://github.com/sw-yx/prompt-eng#top-prompt-engineering-
reads)



---
id: 35
title: "Transformer models: an introduction and catalog — 2023\_Edition"
date: '2023-01-16T00:00:00+00:00'
author: xamat
##layout: post
permalink: /transformer-models-an-introduction-and-catalog-2d1e9039f376/
header:
    teaser: https://amatriain.net/blog/images/02-05.png
reading_time:
    - ''
    - ''
categories:
    - machine learning
    - NLP
    - Transformers
image: /blog/images/02-05.png
---

![](/blog/images/02-01.jpeg)

This post is now an [ArXiV paper](https://arxiv.org/abs/2302.07730) that you can print and cite.


**Update 05/2023**

Another pretty large update after 4 months. I was invited to submit the [article]
(https://arxiv.org/abs/2302.07730) to a journal, so I decided to enlist some help from some LinkedIn
colleages and completely revamp it. First off, we added a whole lot of new models, including e.g. many from
the Llama family. Because of this, we also included a new image that includes all the newer models we have
included since February 2023. Since some of those models are fine-tuned or intstruction-tuned, we went into
explaining a bit more what is the difference between a fine-tuned and a pre-trained model, and added a
section on that discussion. We also fixed some details on the catalog itself and added a field on the
license status of each model, which has become very relevant recently. I also added quite a few links to
similar surveys at the end of this post. Finally, there was a lot of editing throughout the paper that I
have incorporated here too. Hope this makes it more useful!

Also, in case you are wondering why the catalog does not include GPT-4 or PALM-2, I explicitly decided not
to include models for which there are no public details of the different elements needed to classify and
understand them even at the most basic level.

Thanks so much to Ananth Sankar, Jie Bing, Praveen Kumar Bodigutla, Timothy J. Hazen, and Michaeel Kazi from
LinkedIn for their help with this latest version!

**Update 01/16/2023**

Six months after my last update, it is clear that the world has been taken by storm by Transformers.
Everyone is talking about [ChatGPT](https://amatriain.net/blog/chatGPT), so I thought I needed to add the
models that got us there. I had already talked about GPTInstruct before, but I added [GPT3.5](#gpt35) and
[ChatGPT](#chatgpt) as independent models although they don’t add too much to the former. I also added a
couple of models from [Eleuther.ai](https://www.eleuther.ai/) and [Anthropic](https://www.anthropic.com/),
the only two startups that seem to be even ready to challenge the OpenAI/Facebook/Google supremacy in
language models. Because of what is happening with ChatGPT, I thought I should add the main competitors from
the big labs: [Sparrow](#Sparrow) from Deepmind/Google, and Blenderbot3 from Facebook. Speaking of startups
though, there has been a lot of talk of [Stability.ai](https://stability.ai/), so I felt I needed to add a
reference to [StableDiffusion](#stablediffusion). Finally, and while not many details are known about
[AlphaFold](#alphafold)'s architeccture, I thought I should add a reference to it since the problem of
protein folding is very important, and Deepmind’s accomplishment in this regard is huge.

Also, there are two concepts that are becoming more and more important in the recent success of
Transformers: On the one hand, [RLHF](#rlhf) (Reinforcement Learning with Human Feedback) for language
models. On the other hand, [Diffusion models](#diffusion). I added a brief section on both these topics.

Now that I am including over 50 Transformers I thought I should highlight those that for some reason I
```

```
consider to be noteworthy. I hope the others don't feel bad about it :-) I also felt that very often I was
searching for Transformer model timelines and none was comprehensive enough, so I bit the bullet and added a
[timeline view](#Timeline) to the catalog.

Enjoy! And, as always, human feedback is welcomed.

**Table of Contents**

- [Catalog Index](#CatalogIndex)
- [What are Transformers](#Transformers)
     - [Encoder/Decoder Architecture](#encoderdecoder)
     - [Attention](#attention)
     - [Foundation vs Fine-tuned models](#foundation)
     - [Reinforcement Learning with Human Feedback](#rlhf)
     - [Diffusion Models](#diffusion)
- [The Transformers Catalog](#TransformersCatalog)
    - [Catalog Table](#CatalogTable)
    - [Catalog Family Tree](#FamilyTree)
    - [Catalog Timeline](#Timeline)
    - [Catalog List](#CatalogList)


<a name="CatalogIndex"></a>**Catalog Index**

Click on the list to access a Tranformer model directly, or keep reading below for more context and
explanations. I am adding an * to those that I consider noteworthy in case you want to start with those.

- [ALBERT](#albert)
- [AlexaTM 20B](#alexa)
- [Alpaca](#alpaca)
- [AlphaFold*](#alphafold)
- [Anthropic Assistant](#anthropicassistant)
- [BART*](#BART)
- [BERT*](#BERT)
- [Big Bird](#BIGBIRD)
- [BlenderBot3*](#blenderbot3)
- [BLOOM*](#BLOOM)
- [ChatGPT*](#chatgpt)
- [Chinchilla](#CHINCHILLA)
- [CLIP](#CLIP)
- [CM3](#CM3)
- [CTRL](#CTRL)
- [DALL-E](#DALLE)
- [DALL-E-2*](#DALLE2)
- [Deberta](#deberta)
- [Decision Transformers](#DECISION)
- [DialoGPT](#DIALOGPT)
- [DistilBERT](#DISTILBERT)
- [DQ-BART](#DQBART)
- [Dolly](#dolly)
- [E5](#e5)
- [ELECTRA](#ELECTRA)
- [ERNIE](#ERNIE)
- [Flamingo](#FLAMINGO)
- [Flan-T5](#flant5)
- [Flan-PALM](#flanpalm)
- [Galactica](#galactica)
- [Gato*](#GATO)
- [GLaM](#GLAM)
- [GLIDE](#GLIDE)
- [GLM](#glm)
- [GC-ViT](#GCVIT)
- [Gopher](#GOPHER)
- [GopherCite](#gophercite)
- [GPT](#GPT)
- [GPT-2](#GPT2)
- [GPT-3](#GPT3)
- [GPT-3.5](#GPT35)
- [GPT-J](#gptj)
- [GPT-Neo](#GPTNEO)
- [GPT-NeoX 20B](#GPTNEOX)
- [GPTInstruct](#GPTINSTRUCT)
- [HTLM](#htlm)
- [Imagen](#IMAGEN)
- [InstructOR](#instructor)
- [Jurassic-1](#JURASSIC1)
```

# Why this post

I have a terrible memory for names. In the past few years we have seen the meteoric appearance of dozens of models of the Transformer family, all of which have funny, but not self-explanatory, names. The goal of this post is to offer a short and simple catalog and classification of the most popular Transformer models. In other words, I needed a Transformers cheat-sheet and couldn't find a good enough one online, so I thought I'd write my own. I hope it can be useful to you too.

# <a name="Transformers"></a>What are Transformers

Transformers are a class of deep learning models that are defined by some architectural traits. They were first introduced in he now famous [Attention is All you Need](https://arxiv.org/abs/1706.03762) paper by Google researchers in 2017 (the paper has accumulated a whooping 38k citations in only 5 years) and associated [blog post](https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html)) by Google researchers in 2017. The paper has accumulated a whopping 38k citations in only 5 years.

The original Transformer architecture is a specific instance of the [encoder-decoder models](https://machinelearningmastery.com/encoder-decoder-long-short-term-memory-networks/) that had become popular just over the 2--3 years prior. Up until that point however, attention was just one of the mechanisms used by these models, which were mostly based on LSTM (Long Short Term Memory) and other RNN (Recurrent Neural Networks) variations. The key insight of the Transformers paper, as the title implies, was that attention could be used as the only mechanism to derive dependencies between input and output.

The input to the Transformer is a sequence of tokens. The output of the encoder is a fixed-dimensional representation for each of the tokens along with a separate embedding for the sequence as a whole. The decoder takes the output of the encoder as input, and spits out a sequence of tokens as its output. In natural language processing (NLP), the tokens can be words or subwords. Subwords are used in all popular Transformer NLP models because they enable us to address the out-of-vocabulary (OOV) issue that is inherent in a word-based system. For simplicity, we will use the term \"token\" to refer to the items in the input and output sequences, understanding that these tokens are subwords for NLP systems. When Transformers are used for processing images or video, the tokens can represent sub-images or objects.

Since the publication of the paper, popular models like BERT and GPT have used only the encoder or decoder aspects of the original architecture. The core commonality of these models is, thus, not the encoder-decoder aspect, but, rather, the architecture of the individual

layers in the encoders and decoders. The layer architecture of
Transformers is based on a self-attention mechanism and a feed-forward
layer, the core aspect of this being that each input token flows through
the layers in its own path, while, at the same time, being directly
dependent on every other token in the input sequence. This enables
parallel and direct computation of contextual token representations
which was previously not possible with sequential models like RNNs.

It is beyond the scope of this paper to go into all the details of the
Transformer architecture. For that, we will refer you to the original
paper or to [The Illustrated Transformer](https://jalammar.github.io/illustrated-transformer/) post. That
being said, we
will briefly describe the most important aspects since we will be
referring to them in the catalog below. Let's start with the basic
architectural diagram from the original paper, and describe some of the
components.

<a name="fig:transformer"></a>![](/blog/images/02-02.png)

## <a name="encoderdecoder"></a> Encoder/Decoder architecture

A generic encoder/decoder architecture (see
Figure [2](#fig:transformer) is composed of two models. The encoder
takes the input and encodes it into a fixed-length vector. The decoder
takes that vector and decodes it into the output sequence. The encoder
and decoder are jointly trained to maximize the conditional
log-likelihood of the output given the input. Once trained, the
encoder/decoder can generate an output given an input sequence or can
score a pair of input/output sequences.

In the case of the original Transformer architecture, both encoder and
decoder had 6 identical layers. In each of those 6 layers the Encoder
had two sub layers: a multi-head self attention layer, and a simple feed
forward network. The self attention layer computes the output
representation of each of its input tokens based on *all the input
tokens*. Each sublayer also has a residual connection and a layer
normalization. The output representation size of the Encoder was 512.
The multi-head self-attention layer in the decoder is slightly different
than that in the encoder. It masks all tokens to the right of the token
whose representation is being computed so as to ensure that the decoder
can only attend to tokens that come before the token it is trying to
predict. This is shown in
Figure [2](#fig:transformer) as \"masked multi-head attention.\" The
Decoder also added a third sublayer, which is another multi-head
attention layer over all the outputs of the Encoder. Note that all those
specific details have since been modified in the many Transformer
variations we will discuss. For example, as we noted before, models like
BERT and GPT are based on only the encoder or decoder.

## <a name="attention"></a>Attention

It is clear from the description above that the only "exotic" elements
of the model architecture are the multi-head attention layers, but, as
described above, that is where the whole power of the model lies! So,
what is attention anyway? An attention function is a mapping between a
query and a set of key-value pairs to an output. Each token in the input
to the attention layer is converted to a query, key and value using
three corresponding matrices. The output representation of each token is
computed as a weighted sum of the values of all the tokens, where the
weight assigned to each value is computed by a compatibility function of
its associated key and the query of the token whose representation is
being computed. The compatibility function used in Transformers is just
a scaled dot product. A key aspect of this attention mechanism in
Transformers is that each token flows through its own computation path,
thus lending itself to parallel computation of the representation of all
the tokens in the input sequence. Now that we understand how attention
works, what is multi-head attention? Well, that is just multiple
attention blocks independently computing representations for each token.
All these representations are then aggregated to give the final
representation of the token. We will refer you again to the [The Illustrated Transformer]
(https://jalammar.github.io/illustrated-transformer/)
for many more details on how the
attention mechanism works, but will reproduce the diagram from the
original paper in Figure [3](#fig:attention) so you get the main idea.

<a name="fig:attention"></a>![The Attention Mechanism from. (left) Scaled Dot-Product Attention,

(right) Multi-Head Attention](/blog/images/02-03.png)

There are several advantages of attention layers over recurrent and
convolutional networks, the two most important being their lower
computational complexity and their higher connectivity, especially
useful for learning long-term dependencies in sequences.

## Foundation vs Fine-tuned models

A foundation model is defined as \"any model that is trained on broad
data (generally using self-supervision at scale) that can be adapted
(e.g., fine-tuned) to a wide range of downstream tasks" (see paper [here](https://arxiv.org/abs/2108.07258))
. When the
foundation model is further trained on a small amount of target-specific
data, it is called a [fine-tuned model](https://huggingface.co/docs/transformers/training) because it has
been fine-tuned
to the specifics of the task at hand.

The [BERT paper](https://export.arxiv.org/abs/1810.04805) popularized this approach of pretraining and
finetuning
for natural language processing, resulting in many researchers using
this approach for many different tasks. As a consequence, most of the
leaderboards for any language-related machine leartning (ML) task became
completely dominated by some version of the Transformer architecture
(see for example the well known [SQUAD leaderboard](https://rajpurkar.github.io/SQuAD-explorer) for question
answering or the [GLUE leaderboard](https://gluebenchmark.com/leaderboard) for general language
understanding, where all systems at the top employ Transformer-based
models).

In its original usage, \"fine-tuning\" referred to tweaking a foundation
model for a specific task, such as spam classification or question
answering. Models, such as BERT, produce representations of the input
tokens, but do not, by themselves, accomplish any task. Thus, it is
necessary to fine-tune them by adding extra neural layers on top of the
foundation model and training the model end to end.

With generative models like GPT, things are a little different. GPT is a
decoder language model trained to predict the next token of a sentence
given all the previous tokens. By training on huge amounts of web
corpora covering almost any topic one can think about, it was found that
GPT could actually produce reasonable outputs to input queries or
prompts. GPT accomplished this by simply predicting the next token given
the input prompt sequence and the output sequence GPT had already
predicted. This language generation actually did a somewhat reasonable
job of tasks like answering questions about general web knowledge,
writing poems etc. Notwithstanding, GPT's outputs were often untruthful
or really not very helpful to the user. To address this, OpenAI
researchers came up with the idea of training GPT to [follow human
instructions](https://arxiv.org/abs/2203.02155) . The resulting models are called InstructGPT. The authors
did this by using a small amount of human-labeled data from a large
variety of tasks to further train GPT. As before, this is a
\"fine-tuning\" process, but the resulting Instruct GPT model is capable
of doing a wide range of tasks, and is, in fact, the class of models
used by the popular ChatGPT engine. Since these models can accomplish a
myriad of tasks, we refer to them as foundation models.

Such additional fine-tuning has been used to generate other general
purpose model variants as well, specifically designed for uses cases
beyond language modeling (predicting the next token in a sequence). For
example, there is a subclass of models fined-tuned to learn text string
embeddings optimized for semantic-relatedness, making them directly
useful for higher-level semantic tasks (e.g. text classification,
clustering, search retrieval, etc.). Examples include [OpenAI's text
embedding models](https://platform.openai.com/docs/guides/embeddings/what-are-embeddings}, [E5]
(https://huggingface.co/intfloat/e5-large),
and [InstructOR](https://huggingface.co/hkunlp/instructor-xl). Transformer encoders
have also been successfully fined-tuned within multi-task learning
frameworks to be able to perform [multiple different semantic tasks using
a single shared Transformer model](https://arxiv.org/abs/2101.11038).

Thus, as we see, while originally foundation models were fine-tuned for
very specific target tasks for specific groups of users, today
fine-tuning is used to also create further versions of foundation models
that can be used by a huge number of users. The process used by ChatGPT
and similar dialog agents, like BlenderBot3 or Sparrow, is fairly

simple: Given a pretrained language model like GPT, we use it to
generate different responses to input prompts (or instructions) and have
humans rank the results. We then use those rankings (aka preferences or
feedback) to train a reward model. The reward model attaches a score to
each output for a given input instruction. After this, a reinforcement
learning with human feedback [(RLHF) process](https://arxiv.org/abs/1706.03741) is used to train the model
on more input instructions, but, rather than use a human to generate the
feedback, the reward model is used to rank the outputs of the model. You
can read much more in these two wonderful posts by [Huggingface](https://huggingface.co/blog/rlhf) and
[Ayush Thakur](https://wandb.ai/ayush-thakur/RLHF/reports/Understanding-Reinforcement-Learning-from-Human-Feedback-RLHF-Part-1--VmlldzoyODk5MTIx).

<a name="fig:rlhf"></a>![Reinforcement Learning with Human Feedback. From HuggingFace's RLHF
blog post at <https://huggingface.co/blog/rlhf>](/blog/images/rlhf.png)

## The impact of Transformers

The application demonstrated in the original Transformer paper was
language translation. This seminal work also showed the architecture
generalized well to other language tasks. Over the next several months,
researchers figured out that Transformers could be used to capture a lot
of inherent knowledge about language by pretraining them on a very large
amount of unsupervised text. The knowledge captured in these models
could then be transferred to target tasks by training on a small amount
of labeled data.

While original Transformers were designed for language tasks, the same
Transformer architecture has been applied to many other applications
like the generation of images, audio, music, or even actions. Because of
that, Transformers are considered a key, if not the key, component to
the new wave of the so-called \"Generative AI\". Generative AI and its
many applications are already revolutionizing many aspects of society (see [here]
(https://www.nature.com/articles/d41586-023-00340-6)
and [here](https://papers.ssrn.com/sol3/Papers.cfm?abstract_id=4337484), for example).

Of course all these applications would not have been possible but for
the myriad of tools that made them readily available to anyone that
could write a few lines of code. Not only were Transformers quickly
integrated into the main AI frameworks (namely [Pytorch]
(https://pytorch.org/tutorials/beginner/transformer_tutorial.html) and
[TensorFlow](https://www.tensorflow.org/text/tutorials/transformer)), but they even enabled the creation of
an entire
company around them. [Huggingface](https://huggingface.co/docs), a startup that has raised over \$
60M to this day, is almost entirely built around the idea of
commercializing their open source [Transformers library](https://github.com/huggingface/transformers).

Transformer model adoption is further accelerated as specialized
hardware is developed by commercial players to improve model training
and inference speed. NVIDIA's [Hopper Tensor Cores](https://resources.nvidia.com/en-us-tensor-core/nvidia-tensor-core-gpu-datasheet) can apply mixed
FP8 and FP16 precisions to dramatically accelerate AI calculations for
Transformers.

Last but not least, we would be remiss if we did not mention the impact
of ChatGPT on the popularization of Transformers. ChatGPT was released
by OpenAI in November 2022, and became the fastest growing app in
history, reaching 1 million users in less than a month, and 100 million
in [less than two](https://www.ubs.com/us/en/wealth-management/insights/article.1585717.html). ChatGPT was
originally a chatbot application built on
top of the [Instruct-GPT model](https://arxiv.org/abs/2203.02155) also called GPT-3.5. Not much later,
OpenAI announced the release of the more powerful [GPT-4](https://openai.com/research/gpt-4), which
[achieves human level capabilities in tasks such as passing the USMLE exam for
medical doctors or the bar exam for lawyers](https://arxiv.org/abs/2303.08774) .

## <a name="diffusion"></a>A Note on Diffusion Models

Diffusion models have become the new state-of-the-art in image
generation, clearly pushing aside the previous approaches such as GANs
(Generative Adversarial Networks). It is important to note, though, that
the diffusion mechanism is not dependent on the Transformer
architecture. However, most modern diffusion approaches do include a
Transformer backbone .

Diffusion models are a class of latent variable models trained through

variational inference. What this means in practice is that we train a
deep neural network to denoise images blurred with some sort of noise
function. Networks that are trained this way are in fact learning the
latent space of what those images represent (see
Figure [4](#fig:diffusion).

<a name="fig:diffusion"></a>![Probabilistic diffusion model architecture from "Diffusion Models: A
Comprehensive Survey of Methods and Applications,\"](/blog/images/diffusion.png)

Diffusion models have relation to other generative models like Denoising
Autoencoders and the famous Generative Adversarial Networks [Generative Adversarial Networks (GAN)]
(https://en.wikipedia.org/wiki/Generative_adversarial_network),
which they have mostly replaced in many applications. Some [authors]
(https://benanne.github.io/2022/01/31/diffusion.html)
will go as far as saying that Diffusion models are just a specific
instance of autoencoders. However, they also admit that the small
differences do transform their application, from the latent
representation of autoencoders to the pure generative nature of
Diffusion models.

# <a name="TransformersCatalog"></a>The Transformers catalog

## Features of a Transformer

So hopefully by now you understand what Transformer models are, and why they are so popular and impactful.
In this section we will introduce a catalog of the most important
Transformer models that have been developed to this day. We will
categorize each model according to the following properties: Family,
Pretraining Architecture, Pretraining or Fine-tuning Task, Extension,
Application, Date (of first known publication), Number of Parameters,
Corpus, License, and Lab. Some are relative simple to understand:
Family represents what original foundation model the specific model is
extending, *extension* describes what the model is adding to the one it
is deriving from, *Date* is when the model was firts published, *Number
of parameters* of the pretrained model, *Corpus* is what data sources
the model was pre-trained or fine-tuned on, *License* describes how the
model can be legally used, and *Lab* lists the institution that
published the model. The remaining propterties deserve a bit more
explanation. We do that in the paregraphs that follow:

### Pretraining Architecture

We described the Transformer architecture as being made up of an Encoder
and a Decoder, and that is true for the original Transformer. However,
since then, different advances have been made that have revealed that in
some cases it is beneficial to use only the encoder, only the decoder,
or both.

Encoder Pretraining

These models, which are also called bi-directional or auto-encoding,
only use the encoder during pretraining, which is usually accomplished
by masking tokens in the input sentence and training the model to
reconstruct those tokens. At each stage during pretraining,
self-attention layers can access all their input tokens. This family of
models are most useful for tasks that require understanding complete
sentences or passages, such as text classification, entailment, and
extractive question answering.

Decoder Pretraining

Decoder models use only the decoder during a pretraining. They are also
called auto-regressive language models because they are trained to
predict the next token based on the previous sequence of tokens.

The self-attention layers can only access the tokens positioned before a
given token in the sentence. They are best suited for tasks involving
text generation.

Transformer (Encoder-Decoder) Pretraining

Encoder-decoder models, also called sequence-to-sequence, use both parts
of the Transformer architecture.

Self-attention layers of the encoder can access all their input tokens,
while the self-attention layers of the decoder can only access the

tokens positioned before a given token. As explained before, the additional attention layer in the decoder enables access to all encoder token representations.

An encoder-decoder model can be pre-trained by optimizing denoising objectives or a combination of denoising and causal language modeling objectives . These objective functions are complex in comparison to the ones used to pretrain encoder only or decoder only models. The encoder-decoder models are best suited for tasks revolving around generating new sentences depending on a given input, such as summarization, translation, or generative question answering.

### Pretraining Task

When training a model we need to define a task for the model to learn on. Some of the typical tasks, such as predicting the next word or learning to reconstruct masked words were already mentioned above. “[Pre-trained Models for Natural Language Processing: A Survey](https://arxiv.org/abs/2003.08271)” includes a pretty comprehensive taxonomy of pretraining tasks, all of which can be considered self-supervised:

When training a model we need to define an objective, or task, for the model to learn on. Some of the typical tasks, such as predicting the next token or learning to reconstruct masked tokens were already mentioned above. "Pre-trained Models for Natural Language Processing: A Survey" includes a pretty comprehensive taxonomy of pretraining tasks, all of which can be considered self-supervised:

1. **Language Modeling (LM):** Predict the next token (in the case of unidirectional LM) or the previous and next token (in the case of bidirectional LM).
2. **Causal Language Modeling (Causality-masked LM):** Autoregressively (left-to-right, in general) predict a text sequence, similar to unidirectional LM.
3. **Prefix Language Modeling (Prefix LM):** In this task, a separate 'prefix' section is separated from the main sequence. Within the prefix, any token can attend to any other token (non-causal). Outside of the prefix, decoding proceeds autoregressively.
4. **Masked Language Modeling (MLM):** Mask out some tokens from the input sentences and then train the model to predict the masked tokens using the surrounding context.
5. **Permuted Language Modeling (PLM):** Same as LM, but on a random permutation of input sequences. A permutation is randomly sampled from all possible permutations. Then some of the tokens are chosen as the target, and the model is trained to predict these targets.
6. **Denoising Autoencoder (DAE):** Take a partially corrupted input and aim to recover the original, undistorted input. Examples of corrupted input include randomly sampling tokens from the input and replacing them with \"\[MASK\]\" elements, randomly deleting tokens from the input, or shuffling sentences in random order.
7. **Replaced Token Detection (RTD):** Using a \"generator\" model, randomly replace certain tokens in the text. The \"discriminator\" is tasked to predict whether a token comes from the original text, or the generator model.
8. **Next Sentence Prediction (NSP):** Train the model to distinguish whether two input sentences are continuous segments from the training corpus.

Note that in the case of fine-tuned models, this property is used to describe the task the model was fine-tuned to, not how it was pre-trained.

### Application

Here we will note what are the main practical applications of the Transformer model. Most of these applications will be in the language domain (e.g. question answering, sentiment analysis, or entity recognition). However, as mentioned before, some Transformer models have also found applications well beyond NLP and are also included in the catalog.

### <a name="CatalogTable"></a>Catalog table

Note: For all the models available in Huggingface, I decided to directly link to the page in the documentation since they do a fantastic job of offering a consistent format and links to everything else you might need, including the original papers. Only a few of the models (e.g. GPT3) are not included in Huggingface.

![](/blog/images/02-04.png)

You can access the original table [here]
(https://docs.google.com/spreadsheets/d/1ltyrAB6BL29cOv2fSpNQnnq2vbX8UrHl47d7FkIf6t4/edit#gid=0) for easier
browsing across the different model features. The full list of models is included below.

### <a name="FamilyTree"></a>Family Tree

The diagram below is a simple view that should highlight the different families of transformers and how they
relate to each other.

<a name="fig:familytree"></a>![](/blog/images/02-05.png)

### <a name="Timeline"></a>Chronological timeline

Another interesting perspective of the catalog is to see it as a chronological timeline. Here you will find
all the transformers in the catalog sorted by their date of publication. In this first visualization, the Y-
axis is only used to cluster transformers of related heritage/family.

<a name="fig:timeline1"></a>![](/blog/images/02-06.png)

In this next visualization, the Y-axis represents model size in millions of parameters. You won't be able to
see all the models in the catalog since many fall right on the same time and size, so please refer to the
previous image for that.

<a name="fig:timeline2"></a>![](/blog/images/02-09.png)

Since the introduction of chatGPT, the LLM open-source community has
experienced a significant surge in activity. With each passing week, we
have observed a proliferation of refined models fine-tuned using the
latest technologies. As a result, these models are continuously
improving, growing more robust and powerful.
Figure [10](#fig:finetunedModels) demonstrates the recent emerged models
since Feb, 2023.

<a name="fig:timeline3"></a>![Recently published LLMs](/blog/images/02-10.png)

### <a name="Catalog List"></a>Catalog List

Finally, here is the full list of all the models in the catalog:

## <a name="albert"></a>ALBERT

-   **Link:**
    <https://huggingface.co/docs/transformers/model_doc/albert>

-   **Family:** BERT

-   **Pretraining Architecture:** Encoder

-   **Pretraining Task:** MLM/NSP

-   **Extension:** Compressed version of BERT using parameter sharing,
    which is much more efficient given the same number of parameters

-   **Application:** Same as BERT

-   **Date (of first known publication):** 09/2019

-   **Num. Params:** Base = 12M, Large = 18M, XLarge = 60M

-   **Corpus:** Same as BERT

-   **License:** Open, Apache-2.0

-   **Lab:** Google

## <a name="alexa"></a>AlexaTM 20B

-   **Link:** <https://github.com/amazon-science/alexa-teacher-models>

-   **Family:** Transformer

-   **Pretraining Architecture:** Encoder/Decoder

-   **Pretraining Task:** Optimizes denoising (80%) and Prefix LM
    (20%)

-   **Extension:** Derived from BART and layernorms located exactly at
    the beginning of each layer. Encoder initialized with internal 10B
    pre-trained encoder.

-   **Application:** Summarization, multi-lingual machine translation
    and NLU tasks

-   **Date (of first known publication):** 08/2022

-   **Num. Params:** 20B

-   **Corpus:** Wikipedia and mC4 datasets in 12 languages.

-   **License:** Limited, non-commercial

-   **Lab:** Amazon

## <a name="alpaca"></a>Alpaca

-   **Link:** <https://github.com/tatsu-lab/stanford_alpaca>

-   **Family:** LLaMA

-   **Pretraining Architecture:** Decoder

-   **Fine-tuning Task:** human instructions

-   **Extension:** Alpaca is fine-tuned from a 7B LLaMA model.

-   **Application:** Evaluated on a variety of text generation and
    classification tasks.

-   **Date (of first known publication):** 03/2023

-   **Num. Params:** 7B

-   **Corpus:** 52K instruction-following data generated using
    self-instruct mechanism, from 175 human-written instruction-output
    pairs.

-   **License:** Limited, Non-commercial bespoke license

-   **Lab:** Stanford

## <a name="alphafold"></a> AlphaFold

-   **Link:** <https://github.com/deepmind/alphafold>

-   **Family:** SE(3) Transformer

-   **Pretraining Architecture:** Encoder

-   **Pretraining Task:** Protein folding prediction of BERT using
    parameter sharing, which is much more efficient given the same
    number of parameters

-   **Extension:** The original Alphafold used a BERT-style Transformer.
    The details of Alphafold's Transformer are not known, but it is
    believed it is an extension of the SE(3)-Tranformer, a 3-D
    equivariant Transformer

-   **Application:** Protein folding

-   **Date (of first known publication):** 09/2019

-   **Num. Params:**b12M, Large = 18M, XLarge = 60M*

-   **Corpus:** Same as BERT

-   **License:** the code is open sourced, with Apache-2.0

-   **Lab:** Deepmind

## <a name="anthropicassistant"></a> Anthropic Assistant

```
-   **Link:** <https://arxiv.org/abs/2112.00861>

-   **Family:** Transformer

-   **Pretraining Architecture:** Decoder

-   **Pretraining Task:** LM

-   **Extension:** These models do not introduce novelties at the
    architecture/pretraining level and they are similar to GPT-3, but
    they focus on how to improve alignment through fine-tuning and
    prompting. Note that the Anthropic Assistant includes several models
    optimized for different tasks. The work often focus on the benefits
    of RLHF. Latest versions of this work study using an LLM to critique
    the model output for harmlessness, and provide feedback data for RL
    this way (RLHF -\> RLAIF).

-   **Application:** Different models with different applications from
    general dialog to code assistant.

-   **Date (of first known publication):** 12/2021

-   **Num. Params:**10M to 52B

-   **Corpus:** 400B tokens from filtered Common Crawl and Books, and
    10% python code. They also create several Dialogue Preference
    datasets for the RLHF training.

-   **License:** N/A

-   **Lab:** Anthropic

## <a name="BART"></a>BART

-   **Link:** <https://huggingface.co/docs/transformers/model_doc/bart>

-   **Family:** BERT for encoder, GPT for Decoder

-   **Pretraining Architecture:** Encoder/Decoder

-   **Pretraining Task:** DAE

-   **Extension:** It can be seen as a generalization of BERT and GPT in
    that it combines ideas from both in the encoder and decoder

-   **Application:** Mostly text generation but also some text
    understanding tasks

-   **Date (of first known publication):** 10/2019

-   **Num. Params:** Base = 140M, Large = 400M. In general, roughly 10%
    larger than BART for equivalent architectures.

-   **Corpus:**Same as RoBERTa (160Gb of news, books, stories)

-   **License:** Open, Apache-2.0

-   **Lab:**Facebook

## <a name="BERT"></a>BERT

-   **Link:** <https://huggingface.co/docs/transformers/model_doc/bert>

-   **Family:** BERT

-   **Pretraining Architecture:** Encoder

-   **Pretraining Task:** MLM/NSP

-   **Extension:It can be seen as a generalization of BERT and GPT in
    that it combines ideas from both in the encoder and decoder**

-   **Application:**General Language Understanding and Question
    Answering. Many other language applications followed

-   **Date (of first known publication):** 10/2018
```

-   **Num. Params:**Base = 110M, Large = 340MT

-   **Corpus:**Toronto Book Corpus and Wikipedia (3.3B Tokens)

-   **License:** Open, Apache-2.0

-   **Lab:**Google

## <a name="BIGBIRD"></a>Big Bird

-   **Link:**
    <https://huggingface.co/docs/transformers/model_doc/big_bird>

-   **Family:** BERT

-   **Pretraining Architecture:** Encoder

-   **Pretraining Task:** MLM

-   **Extension:** Big Bird can extend other architectures such as BERT,
    Pegasus, or RoBERTa by using a sparse attention mechanism that
    elminates the quadratic dependency thus making it more suitable for
    longer sequences

-   **Application:**Particularly well suited for longer sequences, not
    only in text but also e.g. in genomics

-   **Date (of first known publication):** 07/2020

-   **Num. Params:**Depends on the overall architecture

-   **Corpus:**Books, CC-News, Stories and Wikipedia)

-   **License:** Open, Apache-2.0

-   **Lab:**Google

## <a name="blenderbot3"></a>BlenderBot3

-   **Link:** <https://parl.ai/projects/bb3/>

-   **Family:** GPT

-   **Pretraining Architecture:** Decoder

-   **Pretraining Task:** LM

-   **Extension:** BlenderBot 3 is based on a pre-trained OPT. It adds
    features needed for a dialog agent such as long-term memory or the
    ability to search the internet. It is also fine-tuned for some
    specific tasks given human feedback on them.

-   **Application:** Same as GPT-3

-   **Date (of first known publication):** 08/2022

-   **Num. Params:** 3B, 30B, and 175B

-   **Corpus:** 180B tokens = RoBERTa + the Pile + PushShift.io Reddit
    item **License:** Limited, non-commercial, research only

-   **Lab:** Facebook

## <a name="BLOOM"></a>BLOOM

-   **Link:** <https://huggingface.co/docs/transformers/model_doc/bloom>

-   **Family:** GPT

-   **Pretraining Architecture:** Decoder

-   **Pretraining Task:** LM

-   **Extension:** Main difference to GPT-3 is that it uses full
    attention instead of sparse attention

-   **Application:** Same as GPT-3

-   **Date (of first known publication):** 07/2022

-   **Num. Params:**176B

-   **Corpus:** 366B tokens (1.5 TB of text data) multilingual dataset

-   **Lab:** Big Science/Huggingface

-   **License:** Open, but need to follow restrictions in Attachment A,
    BigScience RAIL License v1.0

## <a name="chatgpt"></a>ChatGPT

-   **Link:** <https://chat.openai.com>

-   **Family:** GPT

-   **Pretraining Architecture:** Decoder

-   **Pretraining Task:** LM

-   **Extension:** ChatGPT takes a GPT3.5 (aka GPT3 Davinci-003)
    pretrained model and uses RLHF to finetune the model mostly like
    described in InstructGPT but with slight differences in the data
    collection. ChatGPT is also more than a model since it includes
    extensions for Memory Store and retrieval similar to BlenderBot3

-   **Application:** Dialog agents

-   **Date (of first known publication):** 10/2022

-   **Num. Params:** Same as GPT3

-   **Corpus:** Same as GPT3 + datasets generated for RLHF

-   **License:** Closed source, accessible through API

-   **Lab:** OpenAI

## <a name="CHINCHILLA"></a>Chinchilla

-   **Link:** <https://arxiv.org/abs/2203.15556>

-   **Family:** GPT

-   **Pretraining Architecture:** Decoder

-   **Pretraining Task:** LM

-   **Extension:** Same as Gopher but with optimizations to reduce model
    size and therefore training/inference time with equal or superior
    performance

-   **Application:** Same as Gopher/GPT3

-   **Date (of first known publication):** 03/2022

-   **Num. Params:**70B

-   **Corpus:** Massive Text

-   **License:** Closed source.

-   **Lab:** Deepmind

## <a name="CLIP"></a>CLIP

-   **Link:** <https://huggingface.co/docs/transformers/model_doc/clip>

-   **Family:** CLIP (Also using Resnet, ViT, and vanilla Transformer
    for text)

-   **Pretraining Architecture:** Encoder

-   **Pretraining Task:** predict which of the N × N possible (image,
    text) pairings across a batch actually occurred

-   **Extension:** Combines Resnet and ViT for the visual encoding with
    Transformer for the Textual encoder

-   **Application:** Image/object classification

-   **Date (of first known publication):** 02/2021

-   **Num. Params:** N/A

-   **Corpus:** WIT (WebImageText) - 400 million text,image pairs

-   **License:** Open, MIT license

-   **Lab:** OpenAI

## <a name="CM3"></a>CM3

-   **Link:** <https://arxiv.org/abs/2201.07520>

-   **Family:** HTLM

-   **Pretraining Architecture:** Decoder

-   **Pretraining Task:** Causality-masked LM

-   **Extension:** This is somewhat similar to HTML in its use of
    structured training data. However, it is a different architecture
    and uses causal masking, which makes the model predict, at the end
    of the sequence, an entire missing span of text. It also includes
    image input via Vector Quantized Variational Autoencoding (VQ-VAE)
    tokens.

-   **Application:** Multimodal language model with the ability to do
    structured prompting, zero-shot captioning, image generation, and
    entity linking (via target text prediction of hyperlinks)

-   **Date (of first known publication):** 01/2022

-   **Num. Params:**13B (largest)

-   **Corpus:** CC-News, English Wikipedia

-   **License:** N/A

-   **Lab:** Facebook

## <a name="CTRL"></a>CTRL

-   **Link:** <https://huggingface.co/docs/transformers/model_doc/ctrl>

-   **Family:**

-   **Pretraining Architecture:** Decoder

-   **Pretraining Task:**

-   **Extension:** model can generate text conditioned on control codes
    that specify domain, style, topics, dates, entities, relationships
    between entities, plot points, and task-related behavior

-   **Application:** Controllable text generation

-   **Date (of first known publication):** 09/2019

-   **Num. Params:**1.63B

-   **Corpus:** 140 GB of text including: Wikipedia (En, De, Es, Fr),
    Project Gutenberg, 45 subreddits, OpenWebText2, Amazon Reviews,
    Europarl and UN data from WMT, question-answer pairs from ELI5, and
    the MRQA shared task3, which includes the Stanford Question
    Answering Dataset, NewsQA, TriviaQA, SearchQA, HotpotQA , and
    Natural Questions

-   **License:** Open, BSD-3-Clause license

-   **Lab:** Salesforce

## <a name="DALLE"></a>DALL-E

-   **Link:** <https://openai.com/blog/dall-e>

-   **Family:** GPT

-   **Pretraining Architecture:** Decoder

-   **Pretraining Task:** Caption prediction

-   **Extension:** A differential variational auto-encoder is used to
    learn the visual codebook. The Transformer is a variation of GPT-3

-   **Application:** Text to image

-   **Date (of first known publication):** 01/2021

-   **Num. Params:**12B

-   **Corpus:** 250 million text-images pairs from the internet

-   **License:** N/A

-   **Lab:** OpenAI

## <a name="DALLE2"></a>DALL-E 2

-   **Link:** <https://openai.com/dall-e-2>

-   **Family:** CLIP, GLIDE

-   **Pretraining Architecture:** Encoder/Decoder

-   **Pretraining Task:** Caption prediction

-   **Extension:** Combines CLIP encoder and Diffusion decoder similar
    to GLIDE

-   **Application:** Text to image

-   **Date (of first known publication):** 04/2022

-   **Num. Params:**3.5B

-   **Corpus:** Combination of the DALL-E and CLIP datasets

-   **License:** Closed source, accessible through API

-   **Lab:** OpenAI

## <a name="deberta"></a>DeBERTa

-   **Link:** <https://huggingface.co/microsoft/deberta-large>

-   **Family:** BERT

-   **Pretraining Architecture:** Encoder

-   **Pretraining Task:** MLM

-   **Extension:** Separate positional embedding vector independent from
    the content embedding using disentangled attention matrices for
    contents and relative positions

-   **Application:** Same as BERT

-   **Date (of first known publication):** 06/2020

-   **Num. Params:** 134M (base), 384M (large), 750M (xlarge)

-   **Corpus:** English Wikipedia, BookCorpus, OPENWEBTEXT and STORIES

```
-   **License:** Open, MIT license

-   **Lab:** Microsoft

## <a name="DECISIONTRANSFORMERS"></a>Decision Transformers

-   **Link:** <https://github.com/kzl/decision-transformer>

-   **Family:** GPT, Control Transformers" (not per se a family, but
    grouping here those Transformers that try to model more general
    control, RL-like, tasks)

-   **Pretraining Architecture:** Decoder

-   **Pretraining Task:** Next action prediction

-   **Extension:** Decision Transformers use a GPT architecture and
    extend it by encoding trajectories in a way that they can be learned
    by an auto-regressive task

-   **Application:** General RL (reinforcement learning tasks)

-   **Date (of first known publication):** 06/2021

-   **Num. Params:**Same as GPT

-   **Corpus:** Different corpus for different experiments

-   **License:** Open, MIT license

-   **Lab:** Google/UC Berkeley/Facebook

## <a name="DIALOGGPT"></a>DialoGPT

-   **Link:**
    <https://huggingface.co/docs/transformers/model_doc/dialogpt>

-   **Family:** GPT

-   **Pretraining Architecture:** Decoder

-   **Pretraining Task:** LM

-   **Extension:** GPT-2 architecture trained on dialog data

-   **Application:** Text generation in dialog settings

-   **Date (of first known publication):** 10/2019

-   **Num. Params:**1.5B

-   **Corpus:** 140M Reddit conversations

-   **License:** Open, MIT license

-   **Lab:** Microsoft

## <a name="DISTILLBERT"></a>DistilBERT

-   **Link:**
    <https://huggingface.co/docs/transformers/model_doc/distilbert>

-   **Family:** BERT

-   **Pretraining Architecture:** Encoder

-   **Pretraining Task:** MLM/NSP

-   **Extension:** Compressed version of BERT using distillation, which
    is much more efficient given the same number of parameters

-   **Application:** Same as BERT

-   **Date (of first known publication):** 10/2019
-   **Num. Params:**66M

-   **Corpus:** Same as BERT

-   **License:** Open, Apache-2.0

-   **Lab:** Huggingface

## <a name="DQBART"></a>DQ-BART

-   **Link:** <https://github.com/amazon-science/dq-bart>

-   **Family:** BART

-   **Pretraining Architecture:** Encoder/Decoder

-   **Pretraining Task:** DAE

-   **Extension:** Adds quantization and distillation to a BART model to
    improve performance and model size

-   **Application:** Text generation and understanding

-   **Date (of first known publication):** 03/2022

-   **Num. Params:**Up to 30x reduction in parameters compared to
    standard BART

-   **Corpus:** CNN/DM, XSUM, ELI5, WMT16 En-Ro ( 1M tokens)

-   **License:** Open, Apache-2.0

-   **Lab:** Amazon

## <a name="dolly"></a>Dolly

-   **Link:** <https://huggingface.co/databricks/dolly-v1-6b>

-   **Family:** GPT

-   **Pretraining Architecture:** Decoder

-   **Fine-tuning Task:** human instructions

-   **Extension:** fine-tuned based on the GPT-J-6B (V1) and Pythia
    model (V2)

-   **Application:** Similar to Alpaca

-   **Date (of first known publication):** 03/2023

-   **Num. Params:** V1: 6B, V2: 12B

-   **Corpus:** V1: Instruction corpus same as Alpaca, V2: databricks
    own dataset.

-   **License:** Open

-   **Lab:** Databricks, Inc

## <a name="e5"></a>E5

-   **Link:** <https://huggingface.co/intfloat/e5-large>

-   **Family:** BERT

-   **Pretraining Architecture:** Encoder

-   **Fine-tuning Task:** Semantic similarity using contrastive loss

-   **Extension:** Fine-tunes BERT-based models to create text string
    embeddings optimized for semantic relatedness.

-   **Application:** Text embeddings for semantic relatedness tasks such
    as text clustering or search retrieval.
```

-   **Date (of first known publication):** 12/2022

-   **Num. Params:** 300M (large version)

-   **Corpus:** MS-MARCO, NQ, NLI

-   **License:** Open, MIT license

-   **Lab:** Microsoft

## <a name="ELECTRA"></a>ELECTRA

-   **Link:**
    <https://huggingface.co/docs/transformers/model_doc/electra>

-   **Family:** BERT

-   **Pretraining Architecture:** Encoder

-   **Pretraining Task:** RTD

-   **Extension:** Applied new training techniques including Replaced
    Token Detection

-   **Application:** 03/2020

-   **Date (of first known publication):** 2020

-   **Num. Params:**Base = 110M, Large = 330M

-   **Corpus:** Same as BERT except for Large which is same as XLNet

-   **License:** Open, Apache-2.0

-   **Lab:** Stanford/Google

## <a name="ERNIE"></a>ERNIE

-   **Link:** <https://arxiv.org/abs/1905.07129>

-   **Family:** BERT

-   **Pretraining Architecture:** Encoder

-   **Pretraining Task:** MLM

-   **Extension:** Uses BERT for Encoder architecture, but stacks and
    aggregates two of them for text and entities. This architecture
    could be understood as BERT for text + knowledge graphs

-   **Application:** Knowledge intensive related tasks that might
    benefit from knowledge graphs or entities such as entity recognition

-   **Date (of first known publication):** 05/2019

-   **Num. Params:** Ernie-ViLG 2.0 = 10B, Ernie 3.0 Titan = 260B

-   **Corpus:** English Wikipedia + Wikidata for entitites (note that
    they initialize model to original BERT parameter values

-   **License:** Closed source

-   **Lab:** Baidu, Pengcheng Lab

## <a name="FLAMINGO"></a>Flamingo

-   **Link:** <https://arxiv.org/abs/2204.14198>

-   **Family:** Chinchilla

-   **Pretraining Architecture:** Decoder

-   **Pretraining Task:** Log likelihood of text given some visual input

-   **Extension:** It uses a frozen textual language model (like
    Chinchilla) conditioned on the visual representation, which is

    encoded from a Normalizer-Free ResNet

-   **Application:** Text to image

-   **Date (of first known publication):** 04/2022

-   **Num. Params:**80B (largest)

-   **Corpus:** MultiModal MassiveWeb (M3W): 185 million images and 182
    GB text + a number of text paired with image datasets: ALIGN + LTIP
    (Long Text & Image Pairs) = 312 million images, and VTP (Video &
    Text Pairs) = 27 million short videos (approximately 22 seconds on
    average)

-   **License:** Closed source

-   **Lab:** Deepmind

## <a name="flant5"></a>Flan-T5

-   **Link:**
    <https://huggingface.co/docs/transformers/model_doc/flan-t5>

-   **Family:** T5

-   **Pretraining Architecture:** Encoder/Decoder

-   **Fine-tuning Task:** Instructions for zero-shot and few-shot tasks

-   **Extension:** Flan-T5 is generated by \"Flan Finetuning\" the T5
    models: (1) scaling the number of tasks to 1,836, (2) scaling the
    model size, and (3) finetuning on chain-of-thought data.

-   **Application:** The primary use is to underestand how to improve
    large language models with the right kind of instruction
    fine-tuning. The focus is research on zero-shot and in-context
    few-shot learning NLP tasks, such as reasoning, and question
    answering; advancing fairness and safety research, and understanding
    limitations of current large language models

-   **Date (of first known publication):** 11/2022

-   **Num. Params:** 80M(small), 250M(base), 780M(large), 3B(xl),
    11B(xxl)

-   **Corpus:** Flan finetuned with tasks in Muffin, T0-SF, NIV2, and
    CoT.

-   **License:** Open, Apache-2.0

-   **Lab:** Google

## <a name="flanpalm"></a>Flan-PaLM

-   **Link:** <https://arxiv.org/abs/2210.11416>

-   **Family:** PaLM

-   **Pretraining Architecture:** Decoder

-   **Fine-tuning Task:** Instructions for zero-shot and few-shot tasks

-   **Extension:** Flan-PaLM is generated by \"Flan Finetuning\" the
    PaLM models: (1) scaling the number of tasks to 1,836, (2) scaling
    the model size, and (3) finetuning on chain-of-thought data.

-   **Application:** Same as Flan-T5. The goal is to show Flan
    finetuning can even improve on the largest Google LMs (+9.4%
    improvement average across tasks), with improvements to chain of
    thought, self consistency, multilingual tasks, arithmetic reasoning

-   **Date (of first known publication):** 11/2022

-   **Num. Params:** 8B, 62B, 540B

-   **Corpus:** Flan finetuned with tasks in Muffin, T0-SF, NIV2, and

    CoT.

-   **License:** Closed source

-   **Lab:** Google

## <a name="galactica"></a>Galactica

-   **Link:** <https://galactica.org>

-   **Family:** Transformer

-   **Pretraining Architecture:** Decoder

-   **Pretraining Task:** LM for scientific domain

-   **Extension:** Transformer based architecture in a decoder-only
    setup with a few modifications. Data extensions include special
    tokens for working memory, citations, genetic data, and a few other
    biology related tasks.

-   **Application:** The models are designed to perform scientific
    tasks, including but not limited to citation prediction, scientific
    QA, mathematical reasoning, summarization, document generation,
    molecular property prediction and entity extraction.

-   **Date (of first known publication):** 11/2022

-   **Num. Params:** mini: 125M, base: 1.3B, standard: 6.7B, large: 30B,
    huge: 120B

-   **Corpus:** Trained on 106 billion tokens of open-access scientific
    text and data. This includes papers, textbooks, scientific websites,
    encyclopedias, reference material, knowledge bases, and more

-   **License:** Limited, non-commerical CC BY-NC 4.0 license

-   **Lab:** Meta

## <a name="GATO"></a>Gato

-   **Link:** <https://www.deepmind.com/blog/a-generalist-agent>

-   **Family:** "Control Transformers" (not per se a family, but
    grouping here those Transformers that try to model more general
    control, RL-like, tasks)

-   **Pretraining Architecture:** Decoder

-   **Pretraining Task:** MLM (where tokens are either text or agent
    actions)

-   **Extension:** The standard decoder-only Transformer architecture is
    preceded by an embedding layer that can embed text and images, plus
    add position encodings to add spatial information when applicable.

-   **Application:** Gato presents a generalizable agent that can be
    used beyond text to tasks such as playing Atari or controlling a
    robot arm.

-   **Date (of first known publication):** 05/2022

-   **Num. Params:**1.2B

-   **Corpus:** 1.5T tokens including standard text (e.g. MassiveText),
    vision (e.g. ALIGN), and simulation environments (e.g. ALE Atari, or
    RGB Stacking Real Robot)

-   **License:** Closed source

-   **Lab:** Deepmind

## <a name="GLAM"></a>GLaM

-   **Link:** See blog post[^25]

-   **Family:** Transformer

-   **Pretraining Architecture:** Decoder

-   **Pretraining Task:** LM

-   **Extension:** GLaM introduces a Mixture of 64 Experts to increase parameter count and generalization properties in a somewhat standard decoder-only. Transformer architecture. Only two experts get activated at a time per token, which makes the model also more efficient in training and inference.

-   **Application:** General language modeling

-   **Date (of first known publication):** 12/2021

-   **Num. Params:**1.2T across 64 experts, but only 96B get activated for inference

-   **Corpus:** 1.6T tokens including web pages filtered by Wikipedia and books for quality

-   **License:** Closed source

-   **Lab:** Google

## <a name="GLIDE"></a>GLIDE

-   **Link:** <https://github.com/openai/glide-text2im>

-   **Family:** Diffusion models

-   **Pretraining Architecture:** Encoder

-   **Pretraining Task:** Caption prediction

-   **Extension:** GLIDE can be seen as an extension of the ADM (Ablated Diffusion Model) by the same authors. However, ADM is not per se a Transformer architecture although it does resemble one in some of the configurations the authors use. Given that ADM is by the same authors and was quickly followed up by GLIDE, we think it is fair to consider GLIDE as the first of its kind.

-   **Application:** Text to image

-   **Date (of first known publication):** 12/2021

-   **Num. Params:**3.5B diffusion model (2.3B for visual encoding, 1.2B for textual) + 1.5B for model for upsampling

-   **Corpus:** Same as DALL-E

-   **License:** Open, MIT license

-   **Lab:** OpenAI

## <a name="glm"></a>GLM

-   **Link:** <https://github.com/THUDM/GLM-130B>

-   **Family:** GLM (General Language Model)

-   **Pretraining Architecture:** Encoder and decoder

-   **Pretraining Task:** Auto regressive blank infilling

-   **Extension:** GLM has a bidirectional encoder and a unidirectional decoder in a unified model.

-   **Application:** a General Language Model pretrained with an autoregressive blank-filling objective and can be finetuned on various natural language understanding and generation tasks.

-   **Date (of first known publication):** 03/2022

-   **Num. Params:** Base = 110M, Large = 335M, and also 2B, 10B, 130B

-   **Corpus:** Pile, GLM-130B Chinese corpora, P3, DeepStruct
    finetuning dataset

-   **License:** Open, MIT license

-   **Lab:** Tsinghua

## <a name="GCVIT"></a>Global Context ViT

-   **Link:** <https://github.com/NVlabs/GCVit>

-   **Family:** ViT

-   **Pretraining Architecture:** Encoder

-   **Pretraining Task:** Image classification

-   **Extension:** hierarchical ViT architecture consisting of local and
    global self-attention modules

-   **Application:** Image generation

-   **Date (of first known publication):** 06/2022

-   **Num. Params:** 90M

-   **Corpus:** Imagenet-1K and other task dependent dataasets

-   **License:** Limited, non-commercial license CC-BY-NC-SA-4.0

-   **Lab:** NVidia

## <a name="GOPHER"></a>Gopher

-   **Link:** See blog post[^26]

-   **Family:** GPT

-   **Pretraining Architecture:** Decoder

-   **Pretraining Task:** LM

-   **Extension:** Same as GPT-2 but use RSNorm instead of LayerNorm and
    relative positional encoding rather than absolute

-   **Application:** Mostly Language Modeling and NLU, but also
    extensible like GPT

-   **Date (of first known publication):** 12/2021

-   **Num. Params:**280B

-   **Corpus:** Massive Text (2.35 billion documents, or about 10.5 TB
    of text including Massive Web, Books, Github, News, C4, and
    Wikipedia.

-   **License:** Closed source

-   **Lab:** Deepmind

## <a name="gophercite"></a>GopherCite

-   **Link:** <ttps://www.deepmind.com/blog/gophercite-teaching-language-models-to-support-answers-with-verified-quotes>

-   **Family:** GPT

-   **Pretraining Architecture:** Decoder

-   **Pretraining Task:** LM

-   **Extension:** GopherCite is based on Gopher but adds a step using
    RLHP (Reinforcement Learning from Human Preferences) to learn
    whether not only a response is plausible but also supported

```
-   **Application:** Dialog systems, Q&A, general language generation
    tasks

-   **Date (of first known publication):** 03/2022

-   **Num. Params:**280B

-   **Corpus:** Same as Gopher plus specific dataset generated in the
    RLHP process

-   **License:** Closed source

-   **Lab:** Deepmind

## <a name="GPT"></a>GPT

-   **Link:**
    <https://huggingface.co/docs/transformers/model_doc/openai-gpt>

-   **Family:** GPT

-   **Pretraining Architecture:** Decoder

-   **Pretraining Task:** LM

-   **Extension:**

-   **Application:** Text generation, but adaptable to many other NLP
    tasks when fine tuned.

-   **Date (of first known publication):** 06/2018

-   **Num. Params:**117M

-   **Corpus:** Unsupervised Pretraining on BookCorpus dataset.
    Supervised Finetuning on several task-specific datasets including
    SNLI, RACE, Quora...

-   **License:** N/A

-   **Lab:** OpenAI

## <a name="GPT2"></a>GPT-2

-   **Link:** <https://huggingface.co/docs/transformers/model_doc/gpt2>

-   **Family:** GPT

-   **Pretraining Architecture:** Decoder

-   **Pretraining Task:** LM

-   **Extension:** Minor extensions to the GPT architecture (e.g. layer
    normalization moved to the input of each sub-layer, or increased
    context size from 512 to 1024)

-   **Application:** Text generation, but adaptable to many other NLP
    tasks when fine tuned.

-   **Date (of first known publication):** 02/2019

-   **Num. Params:** 124M, 355M, 774M, 1.5B

-   **Corpus:** 8 million web pages (40 GB). 10X GPT . WebText dataset
    is created by crawling all links at Reddit with at least 3 Karma
    points.

-   **License:** Open, Modified MIT license

-   **Lab:** OpenAI

## <a name="GPT3"></a>GPT-3

-   **Link:** <https://github.com/openai/gpt-3>

-   **Family:** GPT
```

-   **Pretraining Architecture:** Decoder

-   **Pretraining Task:** LM

-   **Extension:** Same as GPT-2 with the only addition of alternating
    dense and locally banded sparse attention patterns, inspired by the
    Sparse Transformer

-   **Application:** Initially text generation, but has over time been
    used for a large range of applications in areas such as code
    generation, but also image and audio generation

-   **Date (of first known publication):** 05/2020

-   **Num. Params:**175 B

-   **Corpus:** 500B tokens including CommonCrawl (410B), WebText2
    (19B), Books1 (12B), Books2 (55B), and Wikipedia (3B)

-   **License:** Closed source

-   **Lab:** OpenAI

## <a name="GPT35"></a>GPT-3.5

-   **Link:**
    <https://platform.openai.com/docs/model-index-for-researchers/models-referred-to-as-gpt-3-5>

-   **Family:** GPT

-   **Pretraining Architecture:** Decoder

-   **Pretraining Task:** LM

-   **Extension:** The GPT3.5 series includes a number of models like
    Davinci-003. They are basically versions of the InstructGPT model.
    See blog post[^28] for details on the comparison of the performance
    to older GPT3 models.

-   **Application:** Dialog and general language, but there is a code
    specific model - codex

-   **Date (of first known publication):** 10/2022

-   **Num. Params:**175B

-   **Corpus:** Same as InstructGPT

-   **License:** Closed source, accessible through API

-   **Lab:** OpenAI

## <a name="gpt3"></a>GPT-J

-   **Link:** <https://huggingface.co/EleutherAI/gpt-j-6B>

-   **Family:** GPT

-   **Pretraining Architecture:** Decoder

-   **Pretraining Task:** LM

-   **Extension:** GPT-J 6B is a Transformer model trained using Mesh
    Transformer JAX and same tokenizer as GPT2/3

-   **Application:** Same as GPT-3

-   **Date (of first known publication):** 05/2021

-   **Num. Params:** 6B

-   **Corpus:** Pile corpus, a large-scale curated dataset created by
    EleutherAI.

-   **License:** Open, Apache-2.0

```
-   **Lab:** EleutherAI

## <a name="GPTNEO"></a>GPT-Neo

-   **Link:** <https://github.com/EleutherAI/gpt-neo>

-   **Family:** GPT

-   **Pretraining Architecture:** Decoder

-   **Pretraining Task:** LM

-   **Extension:** Similar to GPT-2 but uses local attention in every
    other layer with a window size of 256 tokens

-   **Application:** Text generation, but adaptable to many other NLP
    tasks when fine tuned

-   **Date (of first known publication):** 03/2021

-   **Num. Params:** 5B, 2.7B (XL)

-   **Corpus:** Pile --- 840 GB open source text dataset that combines
    22 pre existing datasets

-   **License:** Open, MIT license

-   **Lab:** EleutherAI

## <a name="GPTNEOX20"></a>GPT-NeoX-20B

-   **Link:** <https://huggingface.co/EleutherAI/gpt-neox-20b>

-   **Family:** GPT

-   **Pretraining Architecture:** Decoder

-   **Pretraining Task:** LM

-   **Extension:** Similar to GPT-3 with rotary encoders instead of
    positional, parallel attention and feed forward layers, different
    initialization, and all dense layers instead of alternate
    dense/sparse

-   **Application:** same as GPT-3

-   **Date (of first known publication):** 04/2022

-   **Num. Params:**20B

-   **Corpus:** Pile --- 840 GB open source text dataset that combines
    22 pre existing datasets

-   **License:** Open, Apache-2.0

-   **Lab:** EleutherAI

## <a name="htlm"></a>HTLM

-   **Link:** <https://arxiv.org/abs/2107.06955>

-   **Family:** BART

-   **Pretraining Architecture:** Encoder/Decoder

-   **Pretraining Task:** DAE

-   **Extension:** As opposed to BART, they don't do sentence shuffling

-   **Application:** General purpose language model that allows
    structured HTML prompting

-   **Date (of first known publication):** 07/2021

-   **Num. Params:**400M
```

-   **Corpus:** 23TB of simplified HTML extracted from CommonCrawl

-   **License:** N/A

-   **Lab:** Facebook

## <a name="IMAGEN"></a>Imagen

-   **Link:** <https://imagen.research.google>

-   **Family:** T5, CLIP, Diffusion models

-   **Pretraining Architecture:** T5 (or CLIP or BERT) for frozen text
    encoder + U-net architecture for cascaded diffusion models for text
    to image

-   **Pretraining Task:** image/text pair prediction

-   **Extension:** Imagen adds a few extensions to the U-net diffusion
    architecture (pooled embedding vector, cross attention over text
    embeddings, and Layer Normalizations)

-   **Application:** Text to image

-   **Date (of first known publication):** 06/2022

-   **Num. Params:**2B

-   **Corpus:** a combination of internal datasets, with 460M image-text
    pairs, and the publicly available Laion dataset, with 400M
    image-text pairs

-   **License:** Closed source

-   **Lab:** Google

## <a name="INSTRUCTGPT"></a>InstructGPT

-   **Link:**
    <https://github.com/openai/following-instructions-human-feedback>

-   **Family:** GPT

-   **Pretraining Architecture:** Decoder

-   **Pretraining Task:** LM

-   **Extension:** GPTInstruct starts off with a pretrained GPT3 model
    and adds reward modeling through reinforcement learning after a
    supervised finetuning

-   **Application:** Knowledge-intensive dialog or language tasks

-   **Date (of first known publication):** 01/2022

-   **Num. Params:** Same as GPT3

-   **Corpus:** Same as GPT3 for pretraining, but finetuned and
    optimized using labeler data and prompts

-   **License:** Closed source, Accessible through API

-   **Lab:** OpenAI

## <a name="instructor"></a>InstructOR

-   **Link:** <https://huggingface.co/hkunlp/instructor-xl>

-   **Family:** T5

-   **Pretraining Architecture:** Encoder/Decoder

-   **Fine-tuning Tasks:** Wide variety of instruction based
    text-to-text tasks

```
-   **Extension:** Fine-tunes T5 explicitly to optimize encoder to
    produce a general purpose text string embedding useful for many NLU
    tasks.

-   **Application:** Any NLU task requiring a single text string
    embedding. As of April 2023 InstructOR is the top-ranked system on
    the Massive Text Embedding Benchmark (MTEB).[^29]

-   **Date (of first known publication):** 12/2022

-   **Num. Params:** 330M

-   **Corpus:** Finetuned on MEDI

-   **License:** Open, Apache-2.0

-   **Lab:** University of Hong Kong, University of Washington, META AI

## <a name="JURASSIC1Å"></a>Jurassic-1

-   **Link:** <https://github.com/ai21labs/lm-evaluation>

-   **Family:** GPT

-   **Pretraining Architecture:** Decoder

-   **Pretraining Task:** LM

-   **Extension:** Very similar to GPT-3, but far more parameters and
    improved training efficiency mostly because of the improved
    tokenizer. Also, different ratio of depth to breadth

-   **Application:** Similar to GPT-3

-   **Date (of first known publication):** 09/2021

-   **Num. Params:** 178B (Jumbo), 17B (Grande), 7.5B (Large)

-   **Corpus:** 300B tokens (same as GPT-3)

-   **License:** Closed source, accessible through API

-   **Lab:** AI21

## <a name="LAMDA"></a>LAMDA

-   **Link:** See blog post[^30]

-   **Family:** Transformer

-   **Pretraining Architecture:** Decoder

-   **Pretraining Task:** LM

-   **Extension:** LAMDA focuses on how to improve safety, quality, and
    groundeness using different fine-tuning strategies

-   **Application:** General language modeling, such as translation,
    summarization, question and answers.

-   **Date (of first known publication):** 01/2022

-   **Num. Params:**137B

-   **Corpus:** 1.56T words from public dialog data and other public web
    documents

-   **License:** Closed source

-   **Lab:** Google

## <a name="llama"></a>LLaMA

-   **Link:**
    <https://huggingface.co/docs/transformers/main/model_doc/llama>
```

-   **Family:** Transformer

-   **Pretraining Architecture:** Decoder

-   **Pretraining Task:** LM

-   **Extension:** LLaMA uses a Transformer architecture, and with
    extensions: Pre-normalization, SwiGLU activations, RoPE embeddings,
    reduced memory usage and runtime through efficient implementation of
    the causal multi-head attention, checkpointing to reduce the amount
    of activations that are recomputed during the backward pass, model
    and sequence parallelism to reduce memory usage of the model, and
    uses 1.4T BPE tokens after tokenization.

-   **Application:** Zero and few shot Commonsense reasoning, Question
    answering, Code generation and Reading comprehension.

-   **Date (of first known publication):** 02/2023

-   **Num. Params:** 7B, 13B, 33B and 65B

-   **Corpus:** English CommonCrawl + C4 + Github + Wikipedia +
    Gutenberg and Books3 + ArXiv + Stack Exchange

-   **License:** Limited, Non-commercial bespoke license

-   **Lab:** Meta

## <a name="MBART"></a>mBART

-   **Link:** <https://huggingface.co/docs/transformers/model_doc/mbart>

-   **Family:** BART

-   **Pretraining Architecture:** Encoder/Decoder

-   **Pretraining Task:** DAE

-   **Extension:** Extends BART to multilingual capability

-   **Application:** Translation

-   **Date (of first known publication):** 01/2020

-   **Num. Params:** Same as BART

-   **Corpus:** CC25 Corpus includes 25 monolingual corpuses in
    different languages. Largest corpuses are English (300 GB) and
    Russian (280GB)

-   **License:** Open, MIT license

-   **Lab:** facebook

## <a name="MEGATRON"></a>Megatron

-   **Link:** <https://github.com/NVIDIA/Megatron-LM>

-   **Family:** GPT/BERT/T5

-   **Pretraining Architecture:** Encoder or Decorder, depending on the
    base model

-   **Pretraining Task:** Same as base model

-   **Extension:** Megatron is a family of models that extend previously
    known architectures (namely GPT-2 and BERT originally, but also T5
    more recently) by introducing model parallelism primitives. In the
    case of BERT, the authors also replace the next sentence prediction
    head with sentence order prediction and use whole word n-gram
    masking.

-   **Application:** Same as base model

-   **Date (of first known publication):** 03/2020

-   **Num. Params:** 8.3B (GPT-like), 3.9B (BERT-like)

-   **Corpus:** Original paper uses an aggregate dataset consisting of
    Wikipedia), CC-Stories), RealNews, and OpenWebtext

-   **License:** Limited, Non-commercial usage

-   **Lab:** NVidia

## <a name="MINERVA"></a>Minerva

-   **Link:** See blog post[^31]

-   **Family:** PaLM

-   **Pretraining Architecture:** Decoder

-   **Pretraining Task:** LM

-   **Extension:** Extends PaLM by fine-tuning on the mathematical
    dataset

-   **Application:** Mathematical reasoning

-   **Date (of first known publication):** 06/2022

-   **Num. Params:**540B

-   **Corpus:** Same as PaLM + 118GB dataset of scientific papers from
    the arXiv preprint server and web pages that contain mathematical
    expressions using LaTeX, MathJax, or other mathematical typesetting
    formats

-   **License:** Closed source

-   **Lab:** Google

## <a name="MTNLG"></a>MT-NLG (Megatron TuringNLG)

-   **Link:** <https://developer.nvidia.com/blog/using-deepspeed-and-megatron-to-train-megatron-turing-nlg-530b-the-worlds-largest-and-most-powerful-generative-language-model/>

-   **Family:** GPT

-   **Pretraining Architecture:** Decoder

-   **Pretraining Task:** LM

-   **Extension:** Uses parallelization similar to Megatron to train a
    LM double the size of GPT-3

-   **Application:** Language generation and others (similar to GPT-3)

-   **Date (of first known publication):** 10/2021

-   **Num. Params:**530B

-   **Corpus:** The Pile (800GB dataset) + 2 Common Crawl snapshots

-   **License:** Limited, Non-commercial usage

-   **Lab:** NVidia

## <a name="openassistantllama"></a>OpenAssistant LLaMa

-   **Link:** <https://open-assistant.io/>

-   **Family:** LLaMA

-   **Pretraining Architecture:** Decoder

-   **Extension:** Supervised fine-tuning on crowd sourced
    conversation/assistant data.

-   **Application:** Same as ChatGPT, but open source. Compared to
    alternatives, it uses human generated conversation data

-   **Date (of first known publication):** 04/2023

-   **Num. Params:** 30B for LLaMa

-   **Corpus:** Conversations collected by volunteers available at
    <https://huggingface.co/datasets/OpenAssistant/oasst1>

-   **License:** Limited, Non-commercial bespoke license. There is also
    a version based on Pythia which is Apache licensed.

-   **Lab:** Various open source contributors

## <a name="OPT"></a>OPT

-   **Link:** See blog post[^34]

-   **Family:** GPT

-   **Pretraining Architecture:** Decoder

-   **Pretraining Task:** LM

-   **Extension:** Basically same architecture as GPT-3 but with some
    training improvements introduced in Megatron-LM

-   **Application:** Same as GPT-3

-   **Date (of first known publication):** 05/2022

-   **Num. Params:** 175B (and other smaller versions)

-   **Corpus:** 180B tokens = RoBERTa + the Pile + PushShift.io Reddit

-   **License:** Limited, non-commercial license

-   **Lab:** Facebook

## <a name="PALM"></a>PalM

-   **Link:** <https://ai.googleblog.com/2022/04/pathways-language-model-palm-scaling-to.html>

-   **Family:** Transformer

-   **Pretraining Architecture:** Decoder

-   **Pretraining Task:** LM

-   **Extension:** Palm uses a typical decoder-only Transformer
    architecture, but adds quite a few extensions: SwiGLU activations,
    parallel layers, multi-query attention, RoPE embeddings, Shared
    Input-Output Embeddings, no biases, and a 256k SentencePiece
    vocabulary generated from the training data.

-   **Application:** PalM is designed as a general purpose language
    model with applicability to hundreds of different language tasks

-   **Date (of first known publication):** 04/2022

-   **Num. Params:** 540B

-   **Corpus:** 780B tokens from filtered webpages, books, Wikipedia,
    news articles, source code, and social media conversations. Code
    includes 24 programming languages.

-   **License:** Closed source, Accessible through API

-   **Lab:** Google

## <a name="PEGASUS"></a>Pegasus

-   **Link:**
    <https://huggingface.co/docs/transformers/model_doc/pegasus>

-   **Family:** Transformer

-   **Pretraining Architecture:** Encoder/Decoder

-   **Pretraining Task:** DAE (more concretely GSG) and MLM

-   **Extension:** Extends vanilla Transformer by using a different
    pretraining task (GSG: Gap Sentence Generation) that is better
    suited for summarization

-   **Application:** Summarization

-   **Date (of first known publication):** 12/2019

-   **Num. Params:** Base = 223M, Large = 568M

-   **Corpus:** C4 (750GB) + HugeNews (3.8 TB)

-   **License:** N/A

-   **Lab:** UCL/Google

## <a name="pythia"></a>Pythia

-   **Reference:** [](https://arxiv.org/abs/2304.01373)

-   **Link:** <https://github.com/EleutherAI/pythia>

-   **Family:** Pythia

-   **Pretraining Architecture:** Decoder

-   **Extension:** Trained with the library GPT-NeoX

-   **Application:** Research on language model's behavior,
    functionality, and limitations.

-   **Date (of first known publication):** 04/2023

-   **Num. Params:** 70M, 160M, 410M, 1B, 1.4B, 2.8B, 6.9B, 12B

-   **Corpus:** Pile

-   **License:** Open, Apache-2.0

-   **Lab:** Eleuther AI

## <a name="ROBERTA"></a>RoBERTa

-   **Link:**
    <https://huggingface.co/docs/transformers/model_doc/roberta>

-   **Family:** BERT

-   **Pretraining Architecture:** Encoder

-   **Pretraining Task:** MLM (Dynamic)

-   **Extension:** Extension of BERT with optimized training procedure
    and more data

-   **Application:** Same as BERT

-   **Date (of first known publication):** 07/2019

-   **Num. Params:** 356M

-   **Corpus:** Same as BERT + CC News + OpenWebText + Stories (33B
    Tokens)

-   **License:** N/A

-   **Lab:** UW/Google

## <a name="SEEKER"></a>SeeKer

-   **Link:** <https://parl.ai/projects/seeker>

-   **Family:** GPT (but can extend any family)

-   **Pretraining Architecture:** Encoder/decoder or decoder only,
    depending on the base model it's extending

-   **Pretraining Task:** LM training, Dialogue training

-   **Extension:** SeeKer is an extension that can be applied to any
    Transformer architecture by introducing "search", "knowledge", and
    "response" modules that are introduced during pretraining

-   **Application:** Same as base models

-   **Date (of first known publication):** 03/2022

-   **Num. Params:** SeeKeR Dialogue: 400M, 3B; SeeKeR LM: 365M, 762M,
    1.5B, R2C2 BlenderBot: 400M, 3B

-   **Corpus:** Wizard of the Internet/Wikipedia, PersonaChat, Blended
    Skill Talk, Empatheic Dialogues, Multi-Session Chat, MS MARCO,
    Natural questions, SQuAD, TriviaQA

-   **License:** the code is open sourced.

-   **Lab:** Facebook

## <a name="Sparrow"></a>Sparrow

-   **Link:** <https://arxiv.org/abs/2209.14375>

-   **Family:** GPT

-   **Pretraining Architecture:** Decoder

-   **Pretraining Task:** LM

-   **Extension:** Starts from the Chinchilla 70B model but adds RLHF
    (Reinforcement Learning with Human Feedback). It also adds inline
    evidence a la GopherCite

-   **Application:** Dialog agents and general language generation
    applications like Q&A

-   **Date (of first known publication):** 09/2022

-   **Num. Params:** 70B

-   **Corpus:** Same as Chinchilla + interactive data gathering with
    human annotators during the RLHF process

-   **License:** Closed source

-   **Lab:** Deepmind

## <a name="stablediffusion"></a>StableDiffusion

-   **Link:** <https://huggingface.co/CompVis/stable-diffusion>

-   **Family:** Diffusion

-   **Pretraining Architecture:** Encoder/Decoder

-   **Pretraining Task:** Caption prediction

-   **Extension:** Stable diffusion is basically the Latent Diffusion
    model developed by LMU Munich researchers + some learnings on
    conditional diffusion from DALL-e and Imagen

-   **Application:** Text to image

-   **Date (of first known publication):** 12/2021

-   **Num. Params:** 890M (although there are different, smaller,
    variants)

-   **Corpus:** LAION-5B, a publicly available dataset derived from

```
    Common Crawl

-   **License:** open, CreativeML Open RAIL++-M License

-   **Lab:** LMU Munich + Stability.ai + Eleuther.ai

## <a name="SWINTRANSFORMER"></a>Swin Transformer

-   **Link:** <https://github.com/microsoft/Swin-Transformer>

-   **Family:** ViT

-   **Pretraining Architecture:** Encoder

-   **Pretraining Task:** Same as ViT

-   **Extension:** Extends ViT by replacing the standard multi-head self
    attention (MSA) module by a module based on shifted windows (Swin)
    allowing ViT-like architectures to generalize to higher resolution
    images

-   **Application:** Image (object detection, image classification..)

-   **Date (of first known publication):** 03/2021

-   **Num. Params:** 29M-197M

-   **Corpus:** Imagenet and Imagenet-22k

-   **License:** the code is open sourced, with MIT-license

-   **Lab:** Microsoft

## <a name="SWITCH"></a>Switch

-   **Link:** <https://github.com/google-research/t5x>

-   **Family:** T5

-   **Pretraining Architecture:** Encoder/Decoder

-   **Pretraining Task:** DAE

-   **Extension:** Goal to increase parameter count while keeping FLOP
    operations constant by using efficient routing of MoE (Mixture of
    Experts)

-   **Application:** General language tasks (e.g. question answering)

-   **Date (of first known publication):** 01/2021

-   **Num. Params:** 1T

-   **Corpus:** Colossal Clean Crawled Corpus

-   **License:** Open, Apache-2.0

-   **Lab:** Google

## <a name="t0"></a>T0

-   **Link:** <https://huggingface.co/bigscience/T0>

-   **Family:** T5

-   **Pretraining Architecture:** Encoder/Decoder

-   **Fine-tuning Task:** Natural language prompts

-   **Extension:** T0 stands for \"T5 for Zero Shot\", obtained by
    fine-tuning the T5 model on multitask mixture covering many
    different NLP tasks. Compared with T0, T0p and T0pp were fine-tuned
    with more datasets. T0pp is recommended as it leads (on average) to
    the best performances on a variety of NLP tasks.

-   **Application:** Perform zero-shot inference tasks by specifying the
```

query in natural language, and the models will generate a
    prediction.

-   **Date (of first known publication):** 03/2022

-   **Num. Params:** T0-3B: 3 billion, T0, T0p, T0pp: 11 billion

-   **Corpus:** T0 (Multiple-choice QA, Extractive QA, Closed-Book QA,
    Structure-To-Text, Sentiment, Summarization, Topic Classification,
    Paraphrase Identification. T0p (same as T0, with additional datasets
    from GPT-3's evaluation suite). T0pp (same as T0p, with additional
    datasets from SuperGLUE, excluding NLI sets)

-   **License:** Open, Apache-2.0

-   **Lab:** BigScience

## <a name="T5"></a>T5

-   **Link:** <https://huggingface.co/docs/transformers/model_doc/t5>

-   **Family:** Transformer

-   **Pretraining Architecture:** Encoder/Decoder

-   **Pretraining Task:** DAE

-   **Extension:** Same as original Transformer with some additions such
    as relative positional embeddings like Transformer XL

-   **Application:** General language tasks including machine
    translation, question answering, abstractive summarization, and text
    classification

-   **Date (of first known publication):** 10/2019

-   **Num. Params:** 11 B (up to)

-   **Corpus:** Colossal Clean Crawled Corpus (C4) --- Cleaned up
    version of the Common Crawl dataset --- 750 GB

-   **License:** Open, Apache-2.0

-   **Lab:** Google

## <a name="TRAJECTORYTRANSFORMERS"></a>Trajectory Transformers

-   **Link:** <https://trajectory-transformer.github.io>

-   **Family:** GPT, Control Transformers" (not per se a family, but
    grouping here those Transformers that try to model more general
    control, RL-like, tasks)

-   **Pretraining Architecture:** Decoder

-   **Pretraining Task:** predict most likely sequence

-   **Extension:** Similarly to the Decision Transformers, the main
    extension introduced by Trajectory Transformers is a way to encode a
    trajectory (state, actions, rewards)

-   **Application:** General RL (reinforcement learning tasks)

-   **Date (of first known publication):** 06/2021

-   **Num. Params:** Smaller architecture than GPT

-   **Corpus:** D4RL dataset and other RL datasets depending on the task
    at hand

-   **License:** Open, MIT license

-   **Lab:** UC Berkeley

## <a name="TRANSFORMERXL"></a>Transformer XL

-   **Link:**
    <https://huggingface.co/docs/transformers/model_doc/transfo-xl>

-   **Family:** Transformer

-   **Pretraining Architecture:** Decoder

-   **Pretraining Task:** LM

-   **Extension:** Relative positioned embeddings enable longer-context
    attention when compared to vanilla Transformer model

-   **Application:** General language tasks

-   **Date (of first known publication):** 01/2019

-   **Num. Params:** 151M

-   **Corpus:** Different training datasets depending on experiments,
    but baseline is Wikitext-103

-   **License:** N/A

-   **Lab:** CMU/Google

## <a name="TURINGNLG"></a>Turing-NLG

-   **Link:** <https://www.microsoft.com/en-us/research/blog/turing-nlg-a-17-billion-parameter-language-model-by-microsoft>

-   **Family:** GPT

-   **Pretraining Architecture:** Decoder

-   **Pretraining Task:** LM

-   **Extension:** Optimized version of GPT2 with optimal
    hyperparameters and software/hardware platform to improve training

-   **Application:** Same as GPT-2/3

-   **Date (of first known publication):** 02/2020

-   **Num. Params:** 17B originally, up to 530B more recently

-   **Corpus:** Highest quality subset from The Pile + 2 CC snapshots
    (339B tokens)

-   **License:** N/A

-   **Lab:** Microsoft

## <a name="ul2"></a>UL2

-   **Link:**
    <https://github.com/google-research/google-research/tree/master/ul2>

-   **Family:** Transformer

-   **Pretraining Architecture:** Encoder/Decoder

-   **Pretraining Task:** Mixture-of-Denoisers, which combines diverse
    pre-training paradigms together

-   **Extension:** UL2-20B (Unifying Language Learning) can be
    interpreted as a model that is quite similar to T5 but trained with
    a different objective and slightly different scaling knobs.

-   **Application:** A unified framework for pre-training models that
    are universally effective across datasets and setups.

-   **Date (of first known publication):** 05/2022

-   **Num. Params:** 20B

-   **Corpus:** 1 trillion tokens on C4

-   **License:** Open, Apache-2.0

-   **Lab:** Google

## <a name="vicuna"></a>Vicuna

-   **Link:** <https://vicuna.lmsys.org>

-   **Family:** LLaMA

-   **Pretraining Architecture:** Decoder

-   **Fine-tuning Task:** human instructions

-   **Extension:** LLaMA fine-tuned on user-shared conversations
    collected from ShareGPT.

-   **Application:** Same as ChatGPT

-   **Date (of first known publication):** 03/2023

-   **Num. Params:** 13B

-   **Corpus:** Conversations collected from ShareGPT

-   **License:** Limited, Non-commercial bespoke license

-   **Lab:** UC Berkeley, CMU, Stanford, UC San Diego, and MBZUAI

## <a name="VIT"></a>ViT

-   **Link:** <https://huggingface.co/docs/transformers/model_doc/vit>

-   **Family:** BERT

-   **Pretraining Architecture:** Encoder

-   **Pretraining Task:** Image classification

-   **Extension:** Extension of BERT architecture to train on patches of
    images

-   **Application:** Image classification

-   **Date (of first known publication):** 10/2020

-   **Num. Params:** 86M(Base) to 632M (Huge)

-   **Corpus:** From standard Imagenet to JFT-300M (large inhouse
    dataset)

-   **License:** N/A

-   **Lab:** Google

## <a name="WUDAO2"></a>Wu Dao 2.0

-   **Link:** <https://en.wikipedia.org/wiki/Wu_Dao>

-   **Family:** GLM (General Language Model)

-   **Pretraining Architecture:** Decoder

-   **Pretraining Task:** Autoregressive blank infilling

-   **Extension:** Similar to GPT in that it uses a
    Decoder/autoregressive architecture but applies a different
    pretraining task proposed in the GLM family of models. Besides, Wu
    Dao uses a Fast Mixture of Experts (see
    <https://github.com/laekov/fastmoe>) approach to scale training to
    trillions of parameters

-   **Application:** Language and multimodal (particularly image)

-   **Date (of first known publication):** 06/2021

-   **Num. Params:** 1.75T

-   **Corpus:** 4.9 TB of high quality images and texts in both English
    and Chinese

-   **License:** Closed source

-   **Lab:** Beijing Academy of Artificial Intelligence

## <a name="XLMROBERTA"></a>XLM-RoBERTa

-   **Link:**
    <https://huggingface.co/docs/transformers/model_doc/xlm-roberta>

-   **Family:** RoBERTa

-   **Pretraining Architecture:** Encoder

-   **Pretraining Task:** MLM (Dynamic)

-   **Extension:** An extension of RoBERTa that introduces parameter
    tuning insights in the context of multilingual applications

-   **Application:** Translation and other cross-lingual language tasks

-   **Date (of first known publication):** 10/2019

-   **Num. Params:** Base = 270M, Large = 550M

-   **Corpus:** Cleaned Common Crawl in 100 languages

-   **License:** Open, MIT license

-   **Lab:** Facebook

## <a name="XLNET"></a>XLNet

-   **Link:** <https://huggingface.co/docs/transformers/model_doc/xlnet>

-   **Family:** Transformer XL

-   **Pretraining Architecture:** Decoder

-   **Pretraining Task:** PLM

-   **Extension:** This model basically adapts Transformer XL
    architecture to permutation-based LM

-   **Application:** General language tasks

-   **Date (of first known publication):** 05/2019

-   **Num. Params:** Base=117M, Large=360M

-   **Corpus:** Same as BERT + Giga5 (16GB text) + and aggressively
    filtered ClueWeb 2012-B (19GB), Common Crawl (110 GB)

-   **License:** Open, MIT license

-   **Lab:** CMU/Google

### Further reading

Most of the following references have already been mentioned in the post. However, it is worth listing them here in case you need more details:

- The Huggingface Transformers [documentation](https://huggingface.co/course/chapter1/1?fw=pt) and course is extremely good and comprehensive. I have used myself in this post, and I can’t recommend enough as a natural follow up to what you will find here.
- [A survey of transformers](https://arxiv.org/abs/2106.04554) (Lin et al. 2021) includes a 40 page long survey wit over 170 references and a full blown taxonomy.

There are a few surveys specificially focused on LLMs. Here a afew worth checking out:

- [Pre-trained Models for Natural Language Processing: A Survey](https://arxiv.org/abs/2003.08271) (Quiu et al. 2021) is another 30+ pages long survey that focuses on pretrained models for NLP.

![](/blog/images/02-08.png)

- [Stanford Ecosystem Model Zoo](https://crfm.stanford.edu/ecosystem-graphs/index.html?mode=table)

![](/blog/images/02-11.png)

- [Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond]
(https://arxiv.org/pdf/2304.13712.pdf)

![](/blog/images/02-12.png)

- [A Survey of Large Language Models](https://arxiv.org/abs/2303.18223)

![](/blog/images/02-13.png)

```
---
id: 97
title: "2022: A Year in Books"
date: '2023-01-29T00:00:00+00:00'
author: Xavier
permalink: /2022Books
image: /blog/images/2022Books.png
header:
    teaser: https://amatriain.net/blog/images/2022Books.png
reading_time:
    - ''
    - ''
categories:
    - Books
---
```

I have always loved reading. For the past 3 years I have also gotten into audiobooks, which has increased my book consumption dramatically.
I listen to many audiobooks, and then buy in paper those that I really like. For fiction novels, I still prefer paper, but in some cases
I have found the audio extremely good too.

<img src="/blog/images/2022Books.png">

Here is the list of my 10 favorite books that I read in 2022 (Thanks to [Necip Fazil Ayan]
(https://www.linkedin.com/in/necip-fazil-ayan-3241206a) for prompting me!):

1. [If Nietzsche Were a Narwhal: What Animal Intelligence Reveals About Human Stupidity]
(https://www.amazon.com/Nietzsche-Were-Narwhal-Intelligence-Stupidity/dp/0316388068) - by Justin Gregg- a fascinating and radical look at what intelligence is by comparing humans and animals. Among other things, it discusses the idea of why causal reasoning is not only not needed but it might lead to human extinction. Very thought provoking.
2. [The Culture Map: Breaking Through the Invisible Boundaries of Global Business]
(https://www.amazon.com/The-Culture-Map-Erin-Meyer-audiobook/dp/B07YL7JTT5) - by Erin Meyer - One of my all time favorite books to understand people and the world. I just hoped this became a series of book because there are so many other things to explore in that dimension of cultural background impact on behavior
3. [The Boys](https://www.amazon.com/Audible-The-Boys-A-Novel/dp/B09W62TRPX/) - by Katie H. - My fiction recommendation is a novel that I absolutely loved. It is the first "post pandemic" story that I have read, and it made me cry and think hard. I know the author, which may be played a role in how much I absolutely loved it.
4. [The Dichotomy of Leadership: Balancing the Challenges of Extreme Ownership to Lead and Win]
(https://www.amazon.com/The-Dichotomy-of-Leadership-audiobook/dp/B07BN5NGQ5/) - by Jocko Willink and Leif Babin - I am a fan of Extreme Ownership, which I have included in my top management book [recommendations]
(https://amatriain.net/blog/the-7-1-habits-of-highly-successful-leaders-91cdcd37765/). This is not such a great book as the original one, but it is a necessary companion that clarifies some misunderstandings, and provides practical examples. Only read it as a companion, not on its own.
5. [The Body: A Guide for Occupants](https://www.amazon.com/The-Body-Bill-Bryson-audiobook/dp/B07RB2C1LV/) - by Bill Bryson An amazing trip into what we know, and, more importantly, what we don't know about the human body
6. [Atomic Habits: An Easy & Proven Way to Build Good Habits & Break Bad Ones]
(https://jamesclear.com/atomic-habits) - by James Clear - I believe in the power of habits and have read other books about them. This is a popular one that I had not read. It really lived up and even exceeded my expectations.
7. [Superforecasting: The Art and Science of Prediction](https://www.amazon.com/Superforecasting-audiobook/dp/B0131HGPQQ/ref=sr_1_1?crid=344ZQ8Z5T1W8I) - by Philip Tetlock One of those books that you have to read if you care about how (good) decisions are made.
8. [Accelerate: Building and Scaling High Performing Technology Organizations]
(https://www.amazon.com/Accelerate-Building-Performing-Technology-Organizations/dp/B07BMBYHXL) - by Nicole

Forsgren, Jez Humble, and Gene Kim - Not a book that I enjoyed in the literary sense in any way (in fact, it could probably be a blog post), but it includes really important technical team management lessons and research. I have already applied some of them in practice, and I continue referencing it often.
9. [A Romance on Three Legs: Glenn Gould's Obsessive Quest for the Perfect Piano](https://www.amazon.com/Romance-Three-Legs-Obsessive-Perfect/dp/1596915242) - by Erin Meyer - As I mentioned, I absolutely loved "The Boys". In talking to the author, when she learned that I played the piano she told me that she had written a non-fiction book about Glenn Gould and his obsession. I jumped onto this book and thought it was absolutely brilliant too.
10. [The Selfish Gene](https://www.amazon.com/The-Selfish-Gene-Richard-Dawkins-audiobook/dp/B004U8NB2M/) - by Richard Dawkins -This book had been recommended to me many times. It was also references in "If Nietzsche Were a Narwhal" and Andrej Karpathy also mentioned it in a recent interview. The book lived up to its hype. This is also a great book to listen to since the narration by Richard Dawkins himself and Lalla Ward is fantastic.

```
---
id: 98
title: "Blueprints for recommender system architectures: 10th anniversary edition"
date: '2023-01-29T00:00:01+00:00'
author: Xavier
permalink: /RecsysArchitectures
image: /blog/images/2022Books.png
header:
    teaser: https://amatriain.net/blog/images/NewArchitecture.png
reading_time:
    - ''
    - ''
categories:
    - Recommender Systems
    - Machine Learning
    - Netflix
---
```

Ten years ago, we published [a post](https://netflixtechblog.com/system-architectures-for-personalization-and-recommendation-e081aa94b5d8) in the Netflix tech blog explaining our three-tier architectural approach to building recommender systems (see below). A lot has happened in the last 10 years in the recommender systems space for sure. That’s why, when a few months back I designed [a Recsys course for Sphere](https://www.getsphere.com/cohorts/designing-state-of-the-art-recommender-systems), I thought it would be a great opportunity to revisit the blueprint.

In this blog post I summarize 4 existing architectural blueprints, and present a new one that, in my opinion, encompasses all the previous ones.

At a very high-level, any recommender system has items to score and/or rank, and a machine learned model that does that. This model needs to be trained on some data, which is obtained from the service where the recommender operates in some form of feedback loop. The architectural blueprints that we will see below connect those components (and others) in a general way while incorporating some best practices and guidelines.

# The Netflix three tier architecture

In our post ten years ago, we focused on clearly distinguishing the components that can be executed offline (i.e. not when the recommendations need to be served but rather e.g. once a day), those that need to be computed online (i.e. when the user visits the site and the recommendation is being served) and those somewhere in the middle called nearline (i.e. components that are executed when the user visits the site, but do not need to be served in real-time). At that time, and still today in many cases, most of the big data training of the algorithm was performed offline using systems such as Hadoop or Spark. The nearline layer included things like filtering in response to user events, but also some retraining capabilities such as e.g. folding-in and incremental matrix factorization training (see [here](https://towardsdatascience.com/building-an-incremental-recommender-system-8836e30afaef) for a practical introduction to the topic).

<img src="/blog/images/NetflixArchitecture.png">

Ten years ago I felt like this architecture was state of the art. A lot of things have happened since then in the machine learning world, but this three tiered approach is still pretty relevant. Let’s fast forward.

# Eugene Yan’s 2 x 2 blueprint

Amazon’s Eugene Yan does an amazing job of compiling many industry posts in [his june 2021 post](https://eugeneyan.com/writing/system-design-for-discovery/). He cites and describes systems from Alibaba, Facebook, JD, Doordash, and LinkedIn. If you are interested in this space, you should totally read his post (and the rest of his blog btw). After this amazing compilation and distillation work, he presents the following 2x2 blueprint:

<img src="/blog/images/EugeneArchitecture.png">

There are a few things worth noting here. First off, as opposed to our previous blueprint, this one only distinguishes between online and offline. That being said, the “Build Approx. NN index” is close to the boundary, so almost could be considered nearline. Second, this blueprint very much focuses on the latest trend of neural network and embedding-based recommender systems, particularly on the retrieval side of things. While that does exclude “older” approaches, it is a fair assumption since most recommender systems nowadays have replaced the matrix factorization approaches with newer embedding based dimensionality reduction approaches.

Finally, and very importantly, what’s with those “retrieval” components? Why weren’t they even present in our original blueprint? I am glad you asked. It turns out that at Netflix the catalog of items was so small that we did not have to select a subset for ranking. We could literally rank the whole catalog for every member. However, in most other situations, as I quickly learned at Quora, you cannot rank all items for all users all the time. Therefore, this two phase approach where you first select candidates using some retrieval approach, and then you rank them, is pretty much general purpose.

# Nvidia’s 4 stage blueprint

A few months later, Even and Karl from NVidia’s Merlin team published [a new architectural blueprint](https://medium.com/nvidia-merlin/recommender-systems-not-just-recommender-models-485c161c755e) that they acknowledge extended Eugene’s.

<img src="/blog/images/NVidiaArchitecture.png">

Indeed, it is clear that this is an extension of the previous blueprint where they added a filtering step, and they decomposed ranking into scoring and ordering. While I think those two changes make sense, I do think the way they are named and described is confusing, and not general purpose enough. The key aspect to keep in mind is that both before and after the machine learning model is applied (either for scoring or ranking), many systems apply business logic or some other kind of logic for filtering or re-ranking. However, filtering can also be done after scoring or even ranking. And, as mentioned, ranking or ordering is not necessarily done following some business logic. For example, a multi-armed bandit approach can be used at this stage to learn the optimal exploration/exploitation strategy.

# Fennel.ai’s 8 stage blueprint

Finally, my friends at Fennel.ai recently published [a three posts series](https://fennel.ai/blog/real-world-recommendation-system/) describing an 8-stage recommender systems architecture.

<img src="/blog/images/FennelAIArchitecture.png">

While this might seem simpler than the previous two, there are few things I like. First off, it is more generic than the other two, which are very much focused on neural network/embedding-based systems. Second, and importantly, their blueprint highlights the importance of the feedback loop where the data generated by the system flows back into the training data.

# My latest proposal

Given all the above, and in the context of a course on recommender systems that I recently prepared with Deepak Agarwal, I am proposing the new expanded blueprint below:

<img src="/blog/images/NewArchitecture.png">

I am not going to go into the details of all the components in this post, and might do so in a second part if there is enough interest, but I will point out some important aspects and differences when compared to some of the previous ones:

1. This blueprint highlights the central role of data, and the feedback loop in recommender systems
2. It includes two potential ML models: one for retrieval (e.g. embeddings) and the other one for scoring or ranking.
3. It does not prescribe components having to be offline, online, or nearline but rather introduces the notion that some components are more likely to be online while others are more likely to be offline. This is important given the trend of more and more components transitioning to being more online in modern recommender systems.
4. It includes several components that are optional (dashed lines). Those include even candidate selection. While that component is extremely important in services with large catalogs it is not necessary in others like Netflix.
5. Among those optional components, there are two post-processing/filtering components both after scoring and after ranking. To be clear, scoring and ranking are not necessarily two separate components since an ML model can be optimized directly for ranking, and that is why scoring is also optional. But it is important to note that post-processing and/or filtering can be introduced at almost any step (e.g. it could also be included after candidate generation).

Let me know what you think about this new proposal and whether you would be interested in a more detailed follow up.

References

* [System Architectures for Personalization and Recommendation](https://netflixtechblog.com/system-architectures-for-personalization-and-recommendation-e081aa94b5d8)(2013)
* [Deep Neural Networks for YouTube Recommendations](https://research.google.com/pubs/pub45530.html?authuser=1) (2016)
* [Wide & Deep Learning for Recommender Systems](https://arxiv.org/abs/1606.07792) (2016)
* [Practical Lessons for Building Machine Learning Models in Production](https://www.datacouncil.ai/talks/practical-lessons-for-building-machine-learning-models-in-production) (2017)
* [Related Pins at Pinterest: The Evolution of a Real-World Recommender System](https://arxiv.org/abs/1702.07969) (2017)
* [Powered by AI: Instagram’s Explore recommender system](https://instagram-engineering.com/powered-by-ai-instagrams-explore-recommender-system-7ca901d2a882) (2019)
* [Billion-scale Commodity Embedding for E-commerce Recommendation in Alibaba](https://arxiv.org/abs/1803.02349) (2018)
* [System Design for Recommendations and Search](https://eugeneyan.com/writing/system-design-for-discovery/) (2021)
* [Evolving the Best Sort for Reddit’s Home Feed](https://www.reddit.com/r/blog/comments/o5tjcn/evolving_the_best_sort_for_reddits_home_feed/) (2021)
* [On YouTube’s recommendation system](https://blog.youtube/inside-youtube/on-youtubes-recommendation-system/) (2021)
* [Recommender Systems, Not Just Recommender Models](https://medium.com/nvidia-merlin/recommender-systems-not-just-recommender-models-485c161c755e) (2022)

```
---
id: 99
title: "A generation defining AI moment"
date: '2023-03-25T00:00:01+00:00'
author: Xavier
permalink: /AImoment
image: /blog/images/99-00.png
header:
  teaser: https://amatriain.net/blog/images/99-00.png
reading_time:
    - ''
    - ''
categories:
    - Artificial Intelligence
    - Machine Learning
    - Transformers
    - LLMs
    - Society
---
```

<img src="/blog/images/99-00.png">

Let's take a moment to reflect on everything that is going on with generative AI because what we are experiencing right now will define generations to come.

In my previous post [“ChatGPT: hype or revolution?”](https://amatriain.net/blog/chatGPT), I discussed that while there was some short-term hype with ChatGPT, a revolution was clearly brewing. Well, I am here to tell you that the revolution is already in full swing. Today.
The launch of GPT-4 marks a big milestone for many reasons. Not only GPT4 shows better than human capabilities at many tasks, but it is also multi-modal enabling mind-boggling applications that we just did not think possible only a few months ago.
Let’s take a look at some of the amazing accomplishments by GPT-4 out of the box:

* It scores in the 90th percentile for the BAR exam to become a lawyer
* It scores in the 90th percentile for the SAT both in Math and Reading and Comprehension
* It exceeds the USMLE score required to graduate in medicine in the US [by 20 points](https://www.microsoft.com/en-us/research/publication/capabilities-of-gpt-4-on-medical-challenge-problems/)
* It shows [“sparks of general artificial intelligence”](https://arxiv.org/abs/2303.12712) according to some
* It is much better at coding than ChatGPT

<img src="/blog/images/99-01.png">
[Results of GPT-4 one several standard exams](https://openai.com/research/gpt-4)

<img src="/blog/images/99-02.png">
[GPT-4 is much better in intricate coding tasks than ChatGPT](https://arxiv.org/abs/2303.12712)

<img src="/blog/images/99-04.png">
[GPT-4 can solve real world problems](https://arxiv.org/abs/2303.12712)

I have been using GPT-4 for months, and I have been very impressed. When I was at [Curai](https://www.curaihealth.com), we saw huge improvements in all our medical tasks when moving from GPT-3.5 to GPT-4. And, I have continued to be impressed while using it at LinkedIn in the context of exploratory work and some of our recently announced product launches. Yes, it still has many shortcomings, but most of them are being addressed and will get much better soon. As someone who has been working on AI for over twenty years I can say with no hesitation that what has happened in the field in recent months/years really represents a huge leap. In the following sectiosn I dive into some of the questions regarding this point in time and what it means for the near future.

# What will be disrupted by AI?

Everything. AI will transform almost any vertical and industry in the next few years. From healthcare to education or transportation. White and blue collar jobs will be transformed. A good way to think about it is to compare it to the Internet plus the mobile phone. Think about how both of these have impacted our world in the past few years. AI will impact as much as both of them combined.

This is of course both exciting and scary. As with any powerful technology, it will enable good people to do much more good, and bad people to do more evil. The solution is not to stop technological advancements, but to anticipate bad consequences to prevent them. With AI, it feels like for the first time in history we are doing that. For years there has been a big focus on responsible AI and adjacent topics such as transparency or bias. That is not to say that all those problems have been avoided or solved, but did you ever hear about “responsible internet” when the internet revolution was happening?

So, I think there are more reasons to be excited than to be scared. But, it is important to anticipate and prevent bad outcomes. And, it is also important to get society ready for the changes that are happening and coming. This includes lots of education, and, yes, some regulation.

# So, what has happened now?

As many scientists will try to convince you, nothing really revolutionary has happened on the technical side. The current revolution has happened mostly because of the accumulation of several recent (and not so recent) discoveries coming into place at once.

The first one is Deep Learning. Deep learning, which some dismissed at first as “just overfitted large neural networks” when it re-appeared in the spotlight ten years ago, went from [classifying cats](https://www.wired.com/2012/06/google-x-neural-network/) to much more impressive tasks like voice recognition or beating Go and Poker professional players.

Only five years ago, a novel deep learning architecture, the Transformer, grabbed everyone’s attention (pun intended). In only five years, there has been an explosion of Transformer models. You can read all about them in my now popular [Transformer catalog](https://amatriain.net/blog/transformer-models-an-introduction-and-catalog-2d1e9039f376/).

<img src="/blog/images/99-03.png">

The exponential growth in number of models and model size in only five years has been accompanied by an exponential growth in their capabilities. It is beyond the scope of this post to detail all the innovations that have enabled these models to become so much better in such a short time, but I will list the main ones:

* The fundamental attention mechanism introduced in the original paper
* Improvement in training processes (and in GPU hardware) have enabled training much larger models on much larger datasets. Most recent models are trained on almost all of the internet… and some.
* Reinforcement learning with human feedback, which was more recently introduced, has quickly improved “reasoning” capabilities and alignment.

Now it is important to remember that these large language models have still only been pre-trained to predict the next token (~word). However, we still don’t know why, but there are clearly some [“emerging abilities”](https://arxiv.org/abs/2206.07682) once those models get large enough. The addition of the RLHF has furthermore introduced another level of emergent abilities that are still hard to explain.

In summary, there have been a few discoveries that have all come together and enabled this revolution. However, we still don’t fully understand why all those things work so well together.

# Is GPT-4 AGI? Is it conscious?

The obvious answer to those questions is of course not. The not-so-obvious answer is “who cares?”. In my opinion AGI (Artificial General Intelligence) is a misnomer. We usually talk of general intelligence when we mean “human intelligence”. Human intelligence is actually not as general as we might want to think and is very much constrained by our physical experience, senses, and worldview. One can easily think of AI that will be far less constrained and therefore more “general”.

On the other hand, while we do probably want AI models over time to be more general purpose, the reality is that we might be better off with different models who are each “superhuman” at whatever they do. Think about it: if we had an AI that was better than the best human doctors at medical reasoning and another AI that was better than the best lawyers at legal reasoning why would we care about having an AI that is both of them?

The question about consciousness is philosophically very interesting, and I don't want to dismiss it. But, in order to answer a question like that, we would first need to agree on what consciousness is. Read any book about philosophy of mind and you will quickly realize that there is no commonly accepted definition (in fact, there is a "the problem of definition" in the [Wikipedia article](https://en.wikipedia.org/wiki/Consciousness). So, the answer to is GPT-4 conscious is that it is not according to many definitions of consciousness, but it is according to some.

There is though a more interesting question than those two: can GPT-4 reason? That is an important question, and its answer is a bit controversial. Some researchers will argue that a model that has been trained in the way that GPT-4 has been trained can never reason. Interestingly, both Yann LeCun and Gary Marcus, who have been battling out on the issue of reasoning on other approaches agree that large language models "cannot reason". I think they are both wrong.

Reasoning is not a black or white binary label that requires a specific fundamental approach. As opposed to consciousness, it is not a philosophical concept either. Reasoning can be measured, and is task specific. Therefore GPT-4 does reason in many contexts and for many tasks. This is obvious when you read the ["Sparks of artificial general intelligence" report](https://arxiv.org/abs/2303.12712), which I wished had been entitled something more along the lines of "GPT-4 reasoning capabilities".

# Will GPT-4 replace all other AI?

No.

To start, GPT-4 is not the "end" of anything, but just the beginning. Not even Transformers are the end of anything. I would be disappointed if in say a couple of years we had not found a better architecture than Transformers. We can and will have much more powerful and efficient models that will be able to better reason. For sure.

On the other hand, as powerful as GPT-4 is, and as much as it can solve many tasks, it is not very efficient in many ways. GPT-4 is very complex and requires a lot of computing and energy not only to be trained, but even for inference. It would be silly to use it to replace e.g. XGBoost in many cases where this much simpler and efficient model can get even comparable results. There are other limitations of models like GPT-4 and you might hear about them. For example, they have been trained on older data (up to 2021) and it is hard to get proprietary data into them. However, those are not really fundamental limitations and they are being addressed very quickly. For example, the new [chatgpt-retrieval-plugin](https://github.com/openai/chatgpt-retrieval-plugin) can be used to easily address those two shortcomings. But, the efficiency/ROI of these huge models will remain a real thing in the next few years, where you should expect models to become both more efficient, but also more complex and larger, almost in parallel.

# Conclusion

GPT-4, together with the previous ChatGPT, represents an inflection point in Artificial Intelligence. It is not only that the model has amazing capabilities that would have been unthinkable a few months back, but it is mostly about how these two data points illustrate the mind boggling increase in our rate of innovation. Even if you still believe GPT-4 is not impressive, project what you have seen in the past few months two years forward. What do you see?

So, yes, as Bill Gates said, [the Age of AI has begun](https://www.gatesnotes.com/The-Age-of-AI-Has-Begun).

Btw, if you are able to cope with Lex Fridman, [this interview](https://www.youtube.com/watch?v=L_Guz73e6fw_) with Sam Altman, has some interesting tidbits about GPT-4. Also, take a look at [the book](https://www.impromptubook.com/wp-content/uploads/2023/03/impromptu-rh.pdf) that Reid Hoffman wrote using GPT-4 and [his conversation](https://www.youtube.com/watch?v=CYQniyjirJs) about writing the book.

```
---
id: 100
title: "Towards Reinforcement Learning with AI Feedback (RLAIF). What open-sourced foundation models, instruction tuning, and other recent events mean for the future of AI"
date: '2023-03-30T00:00:01+00:00'
author: Xavier
permalink: /rlaif
image: /blog/images/100-4.png
header:
  teaser: https://amatriain.net/blog/images/100-1.png
reading_time:
    - ''
    - ''
categories:
    - Artificial Intelligence
    - Machine Learning
    - LLMs
---
```

A few weeks back I shared my thoughts on how things were going to evolve in the race to build better/larger/smarter generative AI models, and particularly LLMs. Here is what I had to say:

> If you are working with LLMs, here are three important things you might want to keep in mind to anticipate how things are going to evolve in the next 6 months:*

> 1. The best LLMs are going to be much better in all the different dimensions you probably care about (including e.g. less hallucinations)
> 2. The best open-sourced LLMs are going to be better than the best non-open source LLMs nowadays. As an example, Facebook AI made a big announcement for their [LLaMA](https://lnkd.in/gfdq8Bed) open source model a couple of weeks back . A few days later Google announced their [Flan-UL2 20B model](https://lnkd.in/ggzfwvGN). The latter is much better. Not as good as GPT-3.5 yet, but getting close.
> 3. While the best LLMs in (1) are going to continue to grow in size, there are going to be models that are much smaller than the current best models yet have better capabilities. Some of them might be open sourced in (2).*

In just three weeks, a few other things that have happened have reinforced my belief, but they have also introduced a very interesting angle that I think is worth highlighting and really paying attention to. In this post I summarize these recent “trends” and discuss how to interpret them.

# More and better open source foundation models

In just a few weeks we have seen a pretty large influx of open source foundation models.
These models not only include the usual players, but other less likely subjects.
After Facebook introduced [LLaMA](https://ai.facebook.com/blog/large-language-model-llama-meta-ai/),
Google presented [ul2](https://ai.googleblog.com/2022/10/ul2-20b-open-source-unified-language.html).
But then we saw Databricks open source [Dolly](https://www.databricks.com/blog/2023/03/24/hello-dolly-democratizing-magic-chatgpt-open-models.html)
and Cerebras presented [Cerebras-GPT](https://www.cerebras.net/blog/cerebras-gpt-a-family-of-open-compute-efficient-large-language-models/
).

<img src="/blog/images/100-0.png">

It is clear that the race for better, smaller, and more efficient models is on. It will be hard to compete with the GPT-4s of the world, but,
as I mentioned above, this is not a winner-takes-all kind of market.

# Improved fine-tuning and training techniques

If you are reading this, you are probably very familiar with the idea of RLHF (Reinforcement Learning with Human Feedback).
You have a longer description in my [Transformer Catalog](https://amatriain.net/blog/transformer-models-an-introduction-and-catalog-2d1e9039f376/),
but in summary, it is the training process in which a foundation base model is trained with labeled data from humans where annotators label pairs of
prompt/responses as good or bad. It has been documented that only a few of those labels can go a long way in improving results, and, in particular,
alignment in LLMs.

<img src="/blog/images/100-1.png">

There have been other interesting improvements to training and fine-tuning LLMs such as [LoRA](https://github.com/microsoft/LoRA).
These improvements, most of which are now available as part of HuggingFace’s
[State-of-the-art Parameter-Efficient Fine-Tuning (PEFT) methods library](https://github.com/huggingface/peft), allow to fine-tune LLMs
in an efficient way such that the resulting fine-tune model is smaller and more efficient. These approaches are kind of a big deal,
because you can turn a huge general-purpose model into a much smaller one that is better at a particular task you are interested in.

# Instruction fine tuning

A specific kind of model fine-tuning is the so-called instruction tuning. Base foundational models like GPT-3 are not very good at
following multi-step instructions. However, models that are fine-tuned to instructions like GPT-3.5 (the model used in ChatGPT) are
much better. Instructions for fine-tuning can even be generated by the model itself in the so-called
[self-instruct approach](https://arxiv.org/abs/2212.10560)

<img src="/blog/images/100-2.png">

There are several examples of these instruction fine-tuned models available already as open source. For example,
[Openchatkit](https://www.together.xyz/blog/openchatkit) is a model fine-tuned for chat from EleutherAI’s GPT-NeoX-20B

with over 43 million instructions. Or HuggingFace’s [OpenAssistant]
(https://huggingface.co/OpenAssistant/oasst-sft-1-pythia-12b) a
Pythia 12B fine-tuned on 22k human demonstrations.

# Fine-tuning open source foundation models with data from other models

Ok, so now things are starting to get interesting: it turns out that you can not only get a foundation model and turn it into a better
and more efficient model by feeding in instructions, but those instructions you are feeding in can be themselves generated by a model!

And, wait for it, the instructions that you are generating to fine-tune your foundation model do not need to come from an open-sourced model, do they?

Well, meet Stanford’s [Alpaca](https://crfm.stanford.edu/2023/03/13/alpaca.html) , a “controversial” model that uses LLaMA foundation model from
Meta and finetunes with instructions created by OpenAI’s davinci-003.

<img src="/blog/images/100-3.png">

Alpaca is controversial for two reasons: First, it uses a foundation model LLaMA that is not completely open sourced.
LLaMA is only shared with those who follow Meta’s process, and, while model
weights [leaked soon on 4chan](https://www.theverge.com/2023/3/8/23629362/meta-ai-language-model-llama-leak-online-misuse),
Meta is not very happy about this. Secondly, it is not clear that OpenAI’s TOS (Terms of Service) allow for this kind of usage.
If you own a foundation model, you probably don’t want it to be generating data to train the competition’s model.
But, it is not super clear how that can be enforced at this point.

To be fair, Stanford did not release the model weights for Alpaca, and, in fact,
soon after [removed the model entirely](https://gizmodo.com/stanford-ai-alpaca-llama-facebook-taken-down-chatgpt-1850247570).
But, you can imagine that is not the end of it all.

Soon after, a similar [GPT4ALL](https://github.com/nomic-ai/gpt4all) model was released. This model was also a LLaMA model
fine-tuned on instructions (using LoRA), this time around from gpt-3.5. The model again faced LlaMA distribution issues, and is
now trying to recreate the same results from GPTJ, but I am sure they might try Cerebras-GPT and others soon.

And, just today, we have [Vicuna](https://vicuna.lmsys.org/) a fine-tune LLaMA on ShareGPT.

On an incredible turn of events, it seems like Google itself was using data generated by ChatGPT to fine tune Bard! According
to [The Information](https://www.theinformation.com/articles/alphabets-google-and-deepmind-pause-grudges-join-forces-to-chase-openai?rc=9kfau8),
a Google engineer resigned over this.

<img src="/blog/images/100-4.png">

# Towards RLAIF and what it means for the future of AI

So, let’s recap:

- There are pretty capable general purpose foundation models being released every day
- These models can be efficiently fine-tuned on instructions to make them even better
- Those instructions can be generated by other models, including those that are closed source.

Models can improve by learning from each other, and they don’t necessarily need human feedback!

While this is not exactly the same as Reinforcement Learning with AI feedback, I think that we are not very far from this paradigm
appearing and dominating much of the landscape. Of course this will introduce a lot of interesting new questions and issues ranging
from legal and IP concerns on model owners all the way to propagation of biases and/or errors through generations of models that are
trained on each other.

I am not really sure what way this will go, but it is fascinating to see how quickly everything is happening before our eyes.
And, while I think the [six months moratorium that some are proposing](https://futureoflife.org/open-letter/pause-giant-ai-experiments/) is
naive and unrealistic, we really need to think hard about all these fast changes and make sure that we

```
better understand their consequences.


---
id: 101
title: "Some of my favorite Recsys resources and links"
date: '2023-04-23T00:00:00+00:00'
author: xamat
##layout: post
permalink: /recsys-resources/
header:
  teaser: https://amatriain.net/blog/images/101-1.png
reading_time:
    - ''
    - ''
categories:
    - machine learning
    - recommender systems
image: /blog/images/101-1.png
---




1. [Introduction to recommender systems: Basics and classic techniques](#Intro)
2. [Beyond the basics](#Beyond)
    1. [Ranking](#ranking)
    2. [Factorization machines](#fm)
    3. [Explore/exploit](#explorerexploit)
    4. [Full page optimization](#page)
    5. [Context-aware recommendations and other approaches](#context)
    6. [Reinforcement learning](#RL)
3. [Deep Learning for recommendations](#DL)
    1. [The Deep Basics](#DeepBasics)
    2. [Embeddings](#embeddings)
    3. [Graph Neural Networks](#gnn)
    4. [Recommending Sequences](#sequences)
    5. [LLMs for recommendations](#llms)
4. [The “systems part” of recommender systems](#systems)
5. [Evaluation and UX](#UX)
6. [End-to-end examples of real-world industrial recommender systems](#e2e)

*****

## <a name="Intro"></a> 1. Introduction to recommender systems: Basics and classic techniques


[Introduction to Recommender Systems: A 4-hour lecture [VIDEO]](https://amatriain.net/blog/introduction-to-
recommender-systems-4/)

[Data Mining Methods for Recommender Systems](https://amatriain.net/pubs/RecsysHandbookChapter.pdf)

[The recommender revolution](https://www.technologyreview.com/2022/04/27/1048517/the-recommender-
revolution/)

<img src="/blog/images/101-1.png">

[On the “Usefulness” of the Netflix Prize](https://amatriain.net/blog/on-the-usefulness-of-the-netflix-
prize-403d360aaf2/)

<img src="/blog/images/101-2.png">

[Kdd 2014 Tutorial - the recommender problem revisited](https://www.slideshare.net/xamat/kdd-2014-tutorial-
the-recommender-problem-revisited)

<img src="/blog/images/101-3.png">

[Feature Engineering for Recommendation Systems -- Part 1](https://blog.fennel.ai/p/feature-engineering-for-
recommendation)

### <a name="Beyond"></a> 2. Beyond the basics

#### <a name="ranking"></a>2.1 Ranking

[What is Learning To Rank?](https://opensourceconnections.com/blog/2017/02/24/what-is-learning-to-rank/)
```

<img src="/blog/images/101-4.png">

[Personalized ‘Complete the Look’ model by Walmart](https://medium.com/walmartglobaltech/personalized-complete-the-look-model-ea093aba0b73)

[Lamdbamart In depth](https://softwaredoug.com/blog/2022/01/17/lambdamart-in-depth.html)

#### <a name="fm"></a>2.2 Factorization machines

<img src="/blog/images/101-5.png">

[Factorization Machines for Item Recommendation with Implicit Feedback Data](https://towardsdatascience.com/factorization-machines-for-item-recommendation-with-implicit-feedback-data-5655a7c749db)

[Factorization Machines](https://www.csie.ntu.edu.tw/~b97053/paper/Rendle2010FM.pdf)

#### <a name="explorerexploit"></a>2.3 Explore/exploit

[Bandits for Recommender Systems](https://eugeneyan.com/writing/bandits/)

[Explore/Exploit Schemes for Web Content Optimization](https://sci-hub.se/10.1109/icdm.2009.52)

[Explore/Exploit for Personalized Recommendation [VIDEO]](https://www.youtube.com/watch?v=LvcoPy0QUuw&list=PLZSO_6-bSqHQCIYxE3ycGLXHMjK3XV7Iz)

<img src="/blog/images/101-6.png">

[Artwork Personalization at Netflix by Netflix](https://netflixtechblog.com/artwork-personalization-c589f074ad76)

[A Contextual-Bandit Approach to Personalized News Article Recommendation](https://arxiv.org/pdf/1003.0146.pdf)

[Recommending Items to Users: An Explore Exploit Perspective](https://www.ueo-workshop.com/wp-content/uploads/2013/10/UEO-Deepak.pdf)

#### <a name="page"></a>2.4 Full page optimization

<img src="/blog/images/101-7.png">

[Learning a Personalized Homepage by Netflix](https://netflixtechblog.com/learning-a-personalized-homepage-aa8ec670359a)

<img src="/blog/images/101-8.png">

[Beyond Ranking: Optimizing Whole-Page Presentation [VIDEO]](https://www.youtube.com/watch?v=1LGJmFadtoI)

[Fair and Balanced: Learning to Present News Stories](https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=07b73ac3e6881865e518b72cff6d82ea08456241)

#### <a name="context"></a>2.5 Context-aware recommendations and other approaches

[The Wisdom of the Few: A Collaborative Filtering Approach Based on Expert Opinions from the Web](https://amatriain.net/pubs/xamatriain_sigir09.pdf)

<img src="/blog/images/101-9.png">

[Multiverse Recommendation: N-dimensional Tensor Factorization for Context-aware Collaborative Filtering](https://amatriain.net/pubs/karatzoglu-recsys-2010.pdf)

[Temporal Diversity in Recommender Systems](https://amatriain.net/pubs/karatzoglu-recsys-2010.pdf)

[Towards Time-Dependant Recommendation based on Implicit Feedback](https://amatriain.net/pubs/karatzoglu-recsys-2010.pdf)

#### <a name="RL"></a>2.6 Reinforcement learning

[Reinforcement Learning for Recommendations and Search](https://eugeneyan.com/writing/reinforcement-learning-for-recsys-and-search/)

<img src="/blog/images/101-10.png">

[Deep Reinforcement Learning for Page-wise Recommendations](https://zhaoxyai.github.io/paper/recsys2018.pdf)

```
## <a name="DL"></a>3. Deep Learning for recommendations

#### <a name="DeepBasics"></a>3.1 The Deep Basics


<img src="/blog/images/101-11.png">

[Neural Collaborative Filtering](https://arxiv.org/pdf/1708.05031.pdf)

<img src="/blog/images/101-12.png">

[Wide & Deep Learning for Recommender Systems by Google](https://arxiv.org/abs/1606.07792)


<img src="/blog/images/101-13.png">

[Deep Learning Recommendation Model for Personalization and Recommendation Systems by Facebook]
(https://arxiv.org/abs/1906.00091)

#### <a name="embeddings"></a>3.2 Embeddings

[Embedding-based Retrieval in Facebook Search by Facebook](https://arxiv.org/pdf/2006.11632.pdf)

<img src="/blog/images/101-14.png">

[Billion-scale Commodity Embedding for E-commerce Recommendation in Alibaba by Alibaba]
(https://arxiv.org/pdf/1803.02349.pdf)

#### <a name="gnn"></a>3.3 Graph Neural Networks


<img src="/blog/images/101-15.png">

[ATBRG: Adaptive Target-Behavior Relational Graph Network for Effective Recommendation]
(https://arxiv.org/pdf/2005.12002.pdf)

<img src="/blog/images/101-16.png">

[Using graph neural networks to recommend related products by Amazon](https://towardsdatascience.com/modern-
recommendation-systems-with-neural-networks-3cc06a6ded2c)

[Modern Recommendation Systems with Neural Networks](https://towardsdatascience.com/modern-recommendation-
systems-with-neural-networks-3cc06a6ded2c)

[Graph Neural Networks in Recommender Systems: A Survey](https://arxiv.org/abs/2011.02260)


#### <a name="sequences"></a>3.4 Recommending Sequences

[Behavior Sequence Transformer for E-commerce Recommendation in Alibaba by Alibaba]
(https://arxiv.org/pdf/1905.06874.pdf)

[Sequential Recommender Systems: Challenges, Progress and Prospects]
(https://www.ijcai.org/Proceedings/2019/0883.pdf)

[Recommending movies: retrieval using a sequential model a Tensorflow example]
(https://www.tensorflow.org/recommenders/examples/sequential_retrieval)


#### <a name="llms"></a>3.5 LLMs for recommendations

<img src="/blog/images/101-33.png">

[Zero and Few Shot Recommender Systems based on Large Language Models]
(https://blog.reachsumit.com/posts/2023/04/llm-for-recsys/)

## <a name="systems"></a>4. The “systems part” of recommender systems


<img src="/blog/images/101-17.png">

[Recommender Systems, Not Just Recommender Models](https://medium.com/nvidia-merlin/recommender-systems-not-
just-recommender-models-485c161c755e)

[Real World Recommendation System - Part 1 by Fennel.ai](https://blog.fennel.ai/p/real-world-recommendation-
```

system)

<img src="/blog/images/101-18.png">

[Blueprints for recommender system architectures: 10th anniversary edition](https://amatriain.net/blog/RecsysArchitectures)

<img src="/blog/images/101-19.png">

[Pinterest Home Feed Unified Lightweight Scoring: A Two-tower Approach by Pinterest](https://medium.com/pinterest-engineering/pinterest-home-feed-unified-lightweight-scoring-a-two-tower-approach-b3143ac70b55)

[How NVidia supports Recommender Systems [VIDEO] by NVidia](https://www.youtube.com/watch?v=wPso35VkuCs)

<img src="/blog/images/101-20.png">

[System Design for Recommendations and Search](https://eugeneyan.com/writing/system-design-for-discovery/)

[Real-time Machine Learning For Recommendations](https://eugeneyan.com/writing/real-time-recommendations/#when-not-to-use-real-time-recommendations)

<img src="/blog/images/101-21.png">

[Near real-time features for near real-time personalization by LinkedIn](https://engineering.linkedin.com/blog/2022/near-real-time-features-for-near-real-time-personalization)

<img src="/blog/images/101-22.png">

[Introducing DreamShard: A reinforcement learning approach for embedding table sharding by Facebook](https://research.facebook.com/blog/2022/12/introducing-dreamshard-a-reinforcement-learning-approach-for-embedding-table-sharding/)

## <a name="UX"></a>5. Evaluation and UX

<img src="/blog/images/101-23.png">

[The death of the stars: A brief primer on online user ratings](https://amatriain.net/blog/the-death-of-the-stars-a-brief-primer-on-online-user-ratings-6740453f27ed/)

[EvalRS: a Rounded Evaluation of Recommender Systems](https://arxiv.org/pdf/2207.05772.pdf)

<img src="/blog/images/101-24.png">

[Beyond NDCG: behavioral testing of recommender systems with RecList](https://arxiv.org/abs/2111.09963)

<img src="/blog/images/101-25.png">

[How to Measure and Mitigate Position Bias](https://eugeneyan.com/writing/position-bias/)

[Rate it Again: Increasing Recommendation Accuracy by User re-Rating](https://amatriain.net/pubs/xamatriain_Recsys09.pdf)

[I like It... I like It Not: Measuring Users Ratings Noise in Recommender Systems](https://amatriain.net/pubs/umap09.pdf)

## <a name="e2e"></a>6. End-to-end examples of real-world industrial recommender systems

[Lessons Learned from building real life recommender systems](https://amatriain.net/blog/ten-lessons-learned-from-building-real/)

[Past, present, and future of recommender systems: An industry perspective [VIDEO]](https://dl.acm.org/doi/abs/10.1145/2959100.2959144)

<img src="/blog/images/101-26.png">

[On YouTube’s recommendation system by Youtube](https://blog.youtube/inside-youtube/on-youtubes-recommendation-system/)

[Deep Neural Networks for YouTube Recommendations by Youtube](https://dl.acm.org/doi/pdf/10.1145/2959100.2959190)

[How Spotify Uses ML to Create the Future of Personalization by Spotify [VIDEO]](https://engineering.atspotify.com/2021/12/how-spotify-uses-ml-to-create-the-future-of-personalization/)

[Recommender systems in industry: A Netflix case study by Netflix](https://amatriain.net/pubs/Recsys-in-industry.pdf)

[Twitter’s recommendation algorithm by Twitter](https://blog.twitter.com/engineering/en_us/topics/open-source/2023/twitter-recommendation-algorithm)

<img src="/blog/images/101-27.png">

[Evolving the Best Sort for Reddit’s Home Feed by Reddit](https://www.reddit.com/r/blog/comments/o5tjcn/evolving_the_best_sort_for_reddits_home_feed/)

[Intelligent Customer Preference engine with real-time ML systems by Walmart [VIDEO]](https://www.tecton.ai/apply/session-video-archive/intelligent-customer-preference-engine-with-real-time-ml-systems-2/)

<img src="/blog/images/101-28.png">

[Homepage Recommendation with Exploitation and Exploration by Doordash](https://doordash.engineering/2022/10/05/homepage-recommendation-with-exploitation-and-exploration/)

[Learning to rank restaurants by Swiggy](https://bytes.swiggy.com/learning-to-rank-restaurants-c6a69ba4b330)

<img src="/blog/images/101-29.png">

[How we use engagement-based embeddings to improve search and recommendation on Faire by Faire](https://craft.faire.com/how-we-use-engagement-based-embeddings-to-improve-search-and-recommendation-on-faire-912277de4e6d)

[Deep Recommender Systems at Facebook [VIDEO] by Facebook](https://www.youtube.com/live/5xcd0V9m6Xs?feature=share)

<img src="/blog/images/101-30.png">

[Building a heterogeneous social network recommendation system by LinkedIn](https://engineering.linkedin.com/blog/2020/building-a-heterogeneous-social-network-recommendation-system)

<img src="/blog/images/101-31.png">

[A closer look at the AI behind course recommendations on LinkedIn Learning by LinkedIn](https://engineering.linkedin.com/blog/2020/course-recommendations-ai-part-one)

<img src="/blog/images/101-32.png">

[Recommending the world's knowledge. Applications of Recommender Systems at Quora](https://www.slideshare.net/LeiYang27/recommending-the-worlds-knowledge)

[Monolith: Real Time Recommendation System With Collisionless Embedding Table by Bytedance/Tiktok](https://arxiv.org/abs/2209.07663)

```
---
id: 102
title: "On why LLMs are just like medical doctors"
date: '2023-04-27T00:00:01+00:00'
author: Xavier
permalink: /llmsdoctors
image: /blog/images/102-1.png
header:
  teaser: https://amatriain.net/blog/images/102-1.png
reading_time:
    - ''
    - ''
categories:
    - Artificial Intelligence
    - Machine Learning
    - LLMs
---
```

<img src="/blog/images/102-1.png">

There is a lot of talk nowadays on avoiding hallucination in LLMs: we need to get IA models to be truthful and furthermore,
we need them to understand their own uncertainty. While I agree those are worthy research questions,
I don’t think we should put all our eggs in the basket of building completely accurate and self-aware AI models.
In fact, I don’t even think these are practical and beneficial goals.

What should we do instead? We should teach users of AIs that there is no "absolute truth" in anything they get from an LLM.
In fact, there is no absolute truth in anything a user gets from a search engine either. Or... from a medical doctor!
I know, this analogy between human doctors and LLMs might sound a controversial, but allow me to explain:

# The problem

In a [fascinating study](https://jamanetwork.com/journals/jamainternalmedicine/fullarticle/1731967)
published ten years ago in the JAMA Internal Medicine journal, Meyer et al. showed that doctors
had a very poor diagnostic accuracy. In easy cases, doctors average accuracy was a bit over 55% while in harder cases
it did not even get to 6%. While those results might sound astonishing, the "slightly better than 50%" accuracy has been replicated in similar studies.
And, there is even a more astonishing finding in Meyer's study: the level of confidence from doctors is almost the same for easy cases (72%)
and hard cases (64%). So, basically, doctors not only have a horrible diagnostic accuracy. They also have no clue on when they might be wrong.

So, how does that translate to LLMs. Well, LLMs also have very little to no notion of the uncertainty on their "reasoning".
While they are [way better](https://www.medrxiv.org/content/10.1101/2023.04.20.23288859v2) than average doctors at diagnosis,
they can also very confidently make stuff up and spit it out.

So, going back to my initial point: users should treat the output of an LLM just as they should treat medical diagnosis:
a qualified opinion, not a ground truth.

# The Solution: ensembles of expert/AI opinions

And, what is the appropriate way to deal with this? What do you do if you get an important diagnosis from a doctor and
you have doubts? You ask for a second opinion (and maybe a third and a fourth). In fact, according to
[some studies](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6484633/),
diagnostic accuracy can increase to 85% when adding the opinion of 9 doctors.

This same approach can be used with LLMs. In fact LLMs allow for many different variations of "second opinions".
In the simplest one, you can simply ask the SAME LLM the same question several times and get a sense of the variability and then
make up your mind. A bit more involved approach requires not only variability in the response of the same model,
but asking different models, which can then be combined using some ensemble technique (with the simplest being majority voting).
There are even more complex solutions where different models (or agents) can be designed to follow different roles.
In ["DERA: Enhancing Large Language Model Completions with Dialog-Enabled Resolving Agents"](https://scholar.google.com/citations?view_op=view_citation&hl=en&user=eoBHpj4AAAAJ&sortby=pubdate&citation_for_view=eoBHpj4AAAAJ:J-pR_7NvFogC),
my former team at Curai develops medical research and medical decision maker agents and show that combining these two
kinds of agents improves results over using a single GPT-4.

Interestingly, just a few hours after publishing the first version of this post, I read about NVidia's
[guardrails](https://github.com/NVIDIA/NeMo-Guardrails) toolkit for minimizing hallucination. It turns out
that they implement exactly this approach.

# The importance of messaging

While the solution outlined below is readily available, generalized, and scales well, it cannot be implemented unless it comes with a
great deal of effort in user messaging.

<img src="/blog/images/102-2.png">

It is true that applications like ChatGPT do have a disclaimer about their “factuality” (see below), but this is not enough.

Chatbots based on LLMs should be explicit about their stochasticity and constantly invite the user to try again, reformulate their
question, and do their research. I have heard concerns that doing so might erode user “trust” in AI, but that is exactly the point!
Users should trust AI less and take more agency over making decisions and verifying the input that goes into making them.

I see a good opportunity for regulation here. While I think mandating transparency or explainability is a moot point,
mandating AIs to explicitly and clearly remind users of their stochasticity and encouraging to ask again/differently and verify sources is easy,
and reasonable.

# Conclusion

Just as many human experts, LLMs fail to be accurate and factual while conveying a high degree of confidence.
Getting them to be more accurate and factual is good, but we cannot simply hope to get them 100% right all the time.
In fact, even what is “right” might depend on cultural background and personal preferences and beliefs. What we need to do is teach
humans to deal with uncertainty and don’t put the agency of decisions and control on AIs. We can do that by conveying the right message,
with conviction, and repeatedly.

```
---
id: 103
title: "Why I post"
date: '2023-05-14T00:00:01+00:00'
author: Xavier
permalink: /whyIpost
image: /blog/images/103-1.png
header:
  teaser: https://amatriain.net/blog/images/103-1.png
reading_time:
    - ''
    - ''
categories:

---
```

<img src="/blog/images/103-1.png">

I many times get this question: "Why do you post?". It is a fair question. In the past, I could have argued that posting helped my career.
And, I think it actually did. Despite having a fair chair of [publications and citations](https://scholar.google.com/citations?user=YYUK51oAAAAJ&hl=en),
it turns out that many people have known me because of some post or video I have published.

You might think that at this point in my career that is no longer true, but it turns out that is still the case. I was at a conference recently and someone who I was introduced
to said "Oh! You are the Transformer Catalog guy!". Pretty interesting that after so many years [this blog post](https://amatriain.net/blog/transformer-models-an-introduction-and-catalog-2d1e9039f376/)
was what they knew me for. Actually, regarding that post, I even recetnly received a request to turn it into a journal submission, and it is already under review.

# The real reason I post, and you should maybe too

Besides all of this though, there is something that has become apparent over the years and that keeps me hooked to posting: my posts **help people**, in all kinds of unsuspected ways!
I know it sounds a bit hard to believe, but I do get such stories all the time. For example, you would not believe the amount of people that I have met over the years that told me that
they learn about Recommender Systems in my 4-hour long Summer School [Youtube video](https://www.youtube.com/watch?v=bLhq63ygoU8&t=25s) that has almost 40k views at this point
(and, that I would not recommend at this point since it is a bit "old").

Let me give you the last of such stories, which literally made my week/month recently. I will not reveal names or details becuase I have not cleared that with the people involved,
but if it's you reading this, feel free to raise your hand.

I was at a Generative AI event in SF last week (one of the two events I get invited to every week, btw). There were a lot of interesting GenAI startups, and people in general.
At one point I got introduced to the founder/CEO of a GenAI startup. It is one of those "hot" startups (are there any non hot GenAI startups?) with a pretty decent early investment
from investors of high pedigree. When this founder heard my name though, he stopped and said: "I have

something important to thank you for." I was surprised, but I thought he was
going to talk about his cofounder, who I know pretty well. But, no, he was talking about himself! "I am in AI thanks to you." Wait, what? I didn't even know they guy. So, this
required some explanation. Here it is:

It turns out that when this person was first getting into AI/ML, they applied for an internship. The internship was declined at first because they did not have the appropriate
background/experience. However, my new cocktail party friend found an obscure post of mine (I think it's [this Quora answer on ALS](https://www.quora.com/What-is-the-Alternating-Least-Squares-method-in-recommendation-systems-And-why-does-this-algorithm-work-intuition-behind-this/answer/Xavier-Amatriain)).
They implemented the algorithm and went back to the internship hiring manager. The hiring manager was impressed, and they got accepted. Not only that, it turned out most of
their internship simply involved getting the ALS algorithm to work, which they had already done as part of the interview, so the whole internship was a huge success. One thing
led to another, and they are now the founder of a successful GenAI startup!

Hearing stories like this one not only make me smile and be happy, but also give me the energy to continue posting. If you have a story like this, I would love to hear!

```
---
id: 104
title: "Prompt Engineering 201: Advanced methods and toolkits"
date: '2023-06-03T00:00:01+00:00'
author: Xavier
permalink: /prompt201
image: /blog/images/104-0.png
header:
  teaser: https://amatriain.net/blog/images/104-0.png
reading_time:
    - ''
    - ''
categories:
    - Artificial Intelligence
    - Machine Learning
    - LLMs
    - Prompt Engineering
---
```

<img src="/blog/images/104-0.png">
DALL-E 2: An old professor with a notebook in his hand talking to a futuristic looking robot. 4k. Professional photo. Photorealistic

Six months ago I published my [Prompt Engineering 101 post](https://amatriain.net/blog/PromptEngineering).
That later turned into a pretty
popular [LinkedIn course](https://www.linkedin.com/learning-login/share?account=104&forceAccount=false&redirect=https%3A%2F%2Fwww.linkedin.com%2Flearning%2Fprompt-engineering-how-to-talk-to-the-ais%3Ftrk%3Dshare_ent_url%26shareId%3D9gu04nS6TX6dqeav6lZp7w%253D%253D)
that has been taken by over 20k people at this point. That post and course were thought out as a gentle introduction to the topic. If you are new to prompt engineering, please start there.
Also, a lot has happened in the world of LLMs since then. So, now is a good time to complement the Prompt Engineering 101 with an up-to-date and a bit more advanced post.
I’ll call it Prompt Engineering 201. I will start off by repeating a “classic” technique that was already covered in the “advanced section” of the original
post, Chain of Thought, but will build up from there.

Before we get into those techniques, a few words on what is Prompt Engineering.

# <a name="prompt_eng"></a>Prompt Design and Engineering

Prompt Design is the process of coming up with the optimal prompt given an LLM and a clearly stated goal. While prompts are "mostly" natural language, there is more to writing a good prompt than just telling the model what you want. Designing a good prompt requires a combination of:

- Understanding of the LLM: Some LLMs might respond differently to the same prompt and might even have keywords (e.g. "endofpromt") that will be interpreted in a particular way
- Domain knowledge: Writing a prompt to e.g. infer a medical diagnosis, requires medical knowlede.
- Iterative approach with some way to measure quality: Coming up with the ideal prompt is usually a trial and error process. It is key to have a way to measure the output better than a simple "it looks good", particularly if the prompt is meant to be used at scale.

Prompt Engineering is prompt design plus a few other important processes:

- Design of prompts at scale: This usually involves design of meta prompts (prompts that generate prompts) and prompt templates (parameterized prompts that can be instantiated at run-time)

- Tool design and integration: Prompts can include results from external tools that need to be integrated.
- Workflow, planning, and prompt management: An LLM application (e.g. chatbot) requires managing prompt libraries, planning, choosing prompts, tools….
- Approach to evaluate and QA prompts: This will include definition of metrics and process to evaluate both automatically as well as with humans in the loop.
- Prompt optimization: Cost and latency depend on model choice and prompt (token length).

Many of the approaches herein can be considered approaches to "automatic prompt design" in that they describe ways to automate the design of
prompts at scale. In the [bonus section](#tools) you will find some of the most interesting prompt engineering tools and frameworks that implement these techniques.
However, it is important to note that none of these approaches will get you to the results of an experienced prompt engineer. An experienced prompt engineer will
understand and be aware of all of the following techniques and apply some of the patterns wherever they apply rather than blindly following a particular approach
for everything. This, for now, still requires judgement and experience. This is good news for you reading this. You can learn and understand these patterns, but you
won't be replaced by any of these libraries. Take these patterns as a starting tools to add to your toolkit, but also experiment and combine them to gain experience
and judgement.

# <a name="techniques"></a>Advanced techniques

Here are the techniques I will be covering:

* [Chain of thought (CoT)](#cot)
* [Automatic Chain of thought (Auto CoT)](#autocot)
* [The format trick](#format)
* [Tools, Connectors, and Skills](#tools)
* [Automatic multi-step reasoning and tool-use (ART)](#art)
* [Self-consistency](#consistency)
* [Tree of thought (ToT)](#tot)
* [Reasoning without observation (ReWoo)](#rewoo)
* [Retrieval Augmented Generation (RAG)](#rag)
* [Forward-looking active retrieval augmented generation (FLARE)](#flare)
* [Reflection](#reflection)
* [Dialog-Enabled Resolving Agents (DERA)](#dera)
* [Expert Prompting](#expert)
* [Chains](#chains)
* [Agents](#agents)
* [Reason and Act (React)](#react)
* [Rails](#rails)
* [Automatic Prompt Engineering (APE)](#ape)
* [Guidance and Constrained Prompting](#guidance)

# <a name="cot"></a>Chain of Thought (CoT)

As mentioned, this technique was already discussed in the previous post. However, it is very noteworthy and it is at the core of many of the newer approaches, so
I thought it was worthwhile to include here again.

Chain of thought was initially described in the [“Chain-of-Thought Prompting Elicits Reasoning in Large Language Models”](https://arxiv.org/abs/2201.11903) paper
by Google researchers. The simple idea here is that given that LLMs have been trained to predict tokens and not explicitly reason, you can get them closer to reasoning
if you specify those required reasoning steps. Here is a simple example from the original paper:

<img src="/blog/images/104-1.png">

Note that in this case the “required reasoning steps” are given in the example in blue. This is the so-called “Manual CoT”. There are two ways of doing chain
of thought prompting (see below). In the basic one, called zero-shot CoT, you simply ask the LLM to “think step by step”. In the more complex version, called “manual CoT”
you have to give the LLM examples of thinking step by step to illustrate how to reason. Manual prompting is more effective, but harder to scale and maintain.

<img src="/blog/images/104-2.png">

# <a name="autocot"></a>Automatic Chain of Thought (Auto-CoT)

As mentioned above, manual CoT is more effective than zero-shot. However, the effectiveness of this example-based CoT depends on the choice of diverse examples,
and constructing prompts with such examples of step by step reasoning by hand is hard and error prone. That is where automatic CoT, presented in the paper
[“Automatic Chain of Thought Prompting in Large Language Models”](https://arxiv.org/abs/2210.03493), comes

into play.

The approach is illustrated in the following diagram:

<img src="/blog/images/104-3.png">

You can read more details in the paper, but if you prefer to jump right into the action, code for auto-cot
is available [here](https://github.com/amazon-science/auto-cot).

# <a name="format"></a>The “format trick”

LLMs are really good at producing an output in a specific format. I don’t know if the “format trick” is a
common term,
but I did hear [Riley Goodside](https://twitter.com/goodside) use it in one of his presentations, so that is
good enough for me.
You can use the format trick for practically anything. Riley illustrated it by producing a LateX preprint
for ArXiv.

<img src="/blog/images/104-4.png">
<img src="/blog/images/104-5.png">

However, if you specify code as the output format in your prompts you can do even more surprising things
like
[generating a complete powerpoint presentation in Visual Basic]
(https://www.reddit.com/r/AIAssisted/comments/13xf8pq/make_powerpoint_presentations_with_chatgpt/).

# <a name="tools"></a>Tools, Connectors, and Skills

Tools are generally defined as functions that LLMs can use to interact with the external world.

For example, in the following code from Langchain, a “Google tool” is instantiated and used to search the
web:

<img src="/blog/images/104-6.png">

Tools are also known as “Connectors” in Semantic Kernel. For example, here is the Bing Connector. Note that
Semantic Kernel has a related concept of “Skill”.
A skill is simply a function that encapsulates a functionality called by an LLM (e.g. summarize a text) but
does not necessarily require a Connector or a Tool
to access the external world. Skills are also sometimes called “affordances” in other contexts.

In the paper [“Toolformer: Language Models Can Teach Themselves to Use Tools”]
(https://arxiv.org/abs/2302.04761), the authors go beyond simple tool usage by
training an LLM to decide what tool to use when, and even what parameters the API needs. Tools include two
different search engines, or a calculator.
In the following examples, the LLM decides to call an external Q&A tool, a calculator, and a Wikipedia
Search Engine.

<img src="/blog/images/104-7.png">

More recently, researchers at Berkeley have trained a new LLM called [Gorilla]
(https://shishirpatil.github.io/gorilla/) that beats GPT-4 at the use of APIs,
a specific but quite general tool.

# <a name="art"></a> Automatic multi-step reasoning and tool-use (ART)

[ART](https://arxiv.org/abs/2303.09014) combines automatic chain of thought prompting and tool usage, so it
can be seen as a combination of
everything we have seen so far. The following figure from the paper illustrates the overall approach:

<img src="/blog/images/104-8.png">

Given a task and an input, the system first retrieves “similar tasks” from a task library.
Those tasks are added as examples to the prompt. Note that tasks in the library are written using a specific
format (or parsing expression grammar
to be more precise). Given those task examples, the LLM will decide how to execute the current task
including the need to call external tools.

At generation time, the ART system parses the output of the LLM until a tool is called, at which point the
tool is called and integrated into the output.
The human feedback step is optional and is used to improve the tool library itself.

# <a name="consistency"></a>Self-consistency

Self consistency, introduced in the paper [“SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection
for Generative Large Language Models”](https://arxiv.org/abs/2303.08896),

is a method to use an LLM to fact-check itself. The idea is a simple ensemble-based approach where the LLM is asked to generate several responses to the same prompt.
The consistency between those responses indicates how accurate the response may be.

<img src="/blog/images/104-9.png">

The diagram above illustrates the approach in a QA scenario. In this case, the “consistency” is measured by the number of answers to passages that agree with the overall
answer. However, the authors introduce two other measures of consistency (BERT-scores, and n-gram), and a fourth one that combines the three.

# <a name="tot"></a>Tree of Thought (ToT)

[Trees of Thought](https://arxiv.org/abs/2305.10601) are an evolution of the CoT idea where an LLM can consider multiple alternative “reasoning paths” (see diagram below)

<img src="/blog/images/104-10.png">

ToT draws inspiration from the traditional AI work on planning to build a system in which the LLM can maintain several parallel “threads” that are evaluated
for consistency during generation until one is determined to be the best one and is used as the output. This approach requires to define a strategy regarding
the number of candidates as well as the number of steps/thoughts after which those candidates will be evaluated. For example, for a “creative writing” task,
the authors use 2 steps and 5 candidates. But, for a “crossword puzzle” task, they keep up to a max of 10 steps and use BFS search.

# <a name="rewoo"></a>Reasoning without observation (ReWOO)

ReWOO was recently presented in the paper ["ReWOO: Decoupling Reasoning from Observations for Efficient Augmented Language Models"](https://arxiv.org/abs/2305.18323). This approach addresses the challenge that many of the other augmentation approaches in this guide present by increasing the number of calls and tokens to the LLM and therefore increasing cost and latency. ReWOO not only improves token efficiency but also demonstrate robustness to tool failure, and also shows good results in using smaller models.

The approach is illustrated in the diagram below. Given a question, the Planner a comprehensive list of plans or meta-plan prior to tool response. This meta-plan instructs Worker to use external tools and collect evidence. Finally, plans and evidence are sent to Solver who composes the final answer.

<img src="/blog/images/104-18.png">

The following image, also from the paper, illustrates the main benefit of the approach by comparing to the "standard" approach of reasoning with observations. In the latter, the LLM is queried for each call to a Tool (observation), which incurs in lots of potential redundancy (and therefore cost, and latency).

<img src="/blog/images/104-19.png">

# <a name="rag"></a>Retrieval Augmented Generation (RAG)
RAG is a technique that has been used for some time to augment LLMs. It was [presented by Facebook](https://ai.facebook.com/blog/retrieval-augmented-generation-streamlining-the-creation-of-intelligent-natural-language-processing-models/)
as a way to improve BART in 2020 and released as a component in the [Huggingface library](https://huggingface.co/docs/transformers/model_doc/rag).

The idea is simple: combine a retrieval component with a generative one such that both sources complement each other (see diagram below from
[the paper](https://arxiv.org/abs/2005.11401).

<img src="/blog/images/104-11.png">

RAG has become an essential component of the prompt engineer’s toolkit, and has evolved into much more complex approaches. In fact, you can consider RAG at this point
almost as a concrete case of Tools, where the tool is a simple retriever or query engine.

The [FastRAG library](https://github.com/IntelLabs/fastRAG) from Intel includes not only the basic but also more sophisticated RAG approaches like the ones described
in other sections in this post.

# <a name="flare"></a> Forward-looking active retrieval augmented generation (FLARE)

[FLARE](https://arxiv.org/abs/2305.06983) is an advanced RAG approach where, instead of retrieving information just once and then generating, the system iteratively uses a
prediction of the upcoming sentence as a query to retrieve relevant documents to regenerate the sentence if it the confidence is low. The diagram below from the paper clearly
illustrates the approach.

<img src="/blog/images/104-12.png">

Note that the authors measure confidence by setting a probability threshold for each token of the generated sentence. However, other confidence measures might be possible.

# <a name="reflection"></a> Reflection

In the Self-consistency approach we saw how LLMs can be used to infer the confidence in a response. In that approach, confidence is measured as a by-product of how similar
several responses to the same question are. Reflection goes a step further and tries to answer the question of whether (or how) we can ask an LLM directly about the confidence
in its response. As Eric Jang puts it, there is [“some preliminary evidence that GPT-4 possess some ability to edit own prior generations based on reasoning whether their output
makes sense”](https://evjang.com/2023/03/26/self-reflection.html).

The [Reflexion paper](https://arxiv.org/abs/2303.11366) proposes an approach defined as “reinforcement via verbal reflection” with different components.
The actor, an LLM itself, produces a trajectory (hypothesis). The evaluator produces a score on how good that hypothesis is.
The self reflection component produces a summary that is stored in memory. The process is repeated iteratively until the Evaluator decides it has a “good enough” answer.

<img src="/blog/images/104-13.png">

# <a name="dera"></a> Dialog-Enabled Resolving Agents (DERA)

[DERA](https://arxiv.org/abs/2303.17071), developed by my former team at Curai Health for their specific healthcare approach defines different agents that, in the context of a dialog take different roles. In the case of high stakes situations like a medical conversation, it pays off to define a set of "Researchers" and a "Decider". The main difference here is that the Researchers operate in parallel vs. the Reflexion Actors that operate sequentially only
if the Evaluator decides.

<img src="/blog/images/104-20.png">

# <a name="expert"></a> Expert Prompting

This recently presented prompting approach proposes to ask LLMs to respond as an expert. It involves 3 different steps:

- Ask LLM to identify experts in a given field related to the prompt/question
- Ask LLM to respond to the question as if it was each of the experts
- Make final decision as a collaboration between the generated responses

Introduced in the (controversial) [“Exploring the MIT Mathematics and EECS Curriculum Using Large Language Models”](https://arxiv.org/abs/2306.08997) beats other approaches like CoT or ToT. It can also be seen as an extension of the Reflection approach

# <a name="chains"></a>Chains

According to LangChain, which I am going to consider the authoritative source for anything Chains, a Chain is [“just an end-to-end wrapper around multiple individual components”]
(https://docs.langchain.com/docs/components/chains/chain)

Now, of course, there are multiple types of Chains where you can combine different types and number of components in increasing complexity.
In the simplest case, a chain only has a Prompt Template, a Model, and an Output Parser.

Very quickly though, Chains can become much more complex and involved. For example, the Map Reduce chain running an initial prompt on each chunk
of data and then running a different prompt to combine all the initial outputs.

Since the process of constructing and maintaining chains can become quite an engineering task, there are a number of tools that have recently
appeared to support it. The main one is the already mentioned LangChain. In
[“PromptChainer: Chaining Large Language Model Prompts through Visual Programming”]
(https://arxiv.org/abs/2203.06566), the authors not only describe the main
challenges in designing chains, but also describe a visual tool to support those tasks. There is a tool with the exact same name and a similar approach available in Beta [here](https://promptchainer.io/). I am told this one has nothing to do with the authors of the original paper though. I haven't used, so I can't vouch for (or against) it.

<img src="/blog/images/104-14.png">

# <a name="agents"></a> Agents

An agent is an LLM that has access to tools, knows how to use them, and can decide when to do so depending on the input
(See [here](https://docs.langchain.com/docs/components/agents/) and [here](https://www.pinecone.io/learn/langchain-agents/)).

<img src="/blog/images/104-15.png">

Agents are not trivial to implement and maintain, that is why tools like Langchain have become a starting point for most people interested in building one.
The “popular” [Auto-GPT](https://github.com/Significant-Gravitas/Auto-GPT) and [also](https://en.wikipedia.org/wiki/Auto-GPT) is just another toolkit to implement LLM agents.

# <a name="react"></a> Reason and act (React)

React is a specific approach to designing agents introduced by Google in [“ReAct: Synergizing Reasoning and Acting in Language Models”](https://www.promptingguide.ai/techniques/react).
This method prompts the LLM to generate both verbal reasoning traces and actions in an interleaved manner, which allows the model to perform dynamic reasoning. Importantly,
the authors find that the React approach reduces hallucination from CoT. However, this increase in groundedness and trustworthiness, also comes at the cost of slightly reduced
flexibility in reasoning steps (see the paper for more details).

<img src="/blog/images/104-16.png">

As with chains and standard agents, designing and maintaining React agents is a pretty involved task, and it is worth using a tool that supports this task.
Langchain is, again, the de facto standard for agent designs. Using Langchain, you can design different kinds of React agents such as the conversational agent,
that adds conversational memory to the base (zero-shot) React agent (see [here](https://www.pinecone.io/learn/langchain-agents/) for examples and details).

# <a name="rails"></a> Rails

A [rail](https://github.com/NVIDIA/NeMo-Guardrails/blob/main/docs/README.md) is simply a programmable way to control the output of an LLM. Rails are specified using
Colang, a simple modeling language, and Canonical Forms, templates to standardize natural language sentences (see
[here](https://github.com/NVIDIA/NeMo-Guardrails/blob/main/docs/getting_started/hello-world.md))

<img src="/blog/images/104-21.png">

Using rails, one can implement ways to have the LLM stick to a particular topic (Topical rail), minimize hallucination (Fact checking rail) or prevent
jailbreaking (Jailbreaking rail).

# <a name="ape"></a> Automatic Prompt Engineering (APE)

APE refers to the approach in which prompts are automatically generated by LLMs rather than by humans. The method, introduced in the
[“Large Language Models Are Human-Level Prompt Engineers”](https://arxiv.org/abs/2211.01910) paper, involves using the LLM in three ways: to
generate proposed prompts, to score them, and to propose similar prompts to the ones scored highly (see diagram below).

<img src="/blog/images/104-17.png">

# <a name="guidance"></a> Constrained Prompting

“Constrained Prompting” is a term recently [introduced by Andrej Karpathy]((https://youtu.be/bZQun8Y4L2A?t=2093)) to describe approaches and languages that allow us to
interleave generation, prompting, and logical control in an LLM flow.

[Guidance](https://github.com/microsoft/guidance) is the only example of such an approach that I know although one could argue that [React](#react) is
also a constrained prompting approach. The tool is not so much a prompting approach but rather a “prompting language”. Using guidance templates, you can pretty much
implement most if not all the approaches in this post. Guidance uses a syntax based on [Handlebars](https://handlebarsjs.com/) that allows to interleave prompting and
generation, as well as manage logical control flow and variables. Because Guidance programs are declared in the exact linear order that they will be executed, the LLM can,
at any point, be used to generate text or make logical decisions.

<img src="/blog/images/104-22.png">

# <a name="tools"></a>Prompt Engineering tools and frameworks

As we have seen throughout this guide, it is hard to implement

* [Langchain](https://docs.langchain.com/docs/)
  * Langchain is the most popular prompt engineering toolkit. While it initially mostly focused on supporting Chains, it now supports Agents and many different Tools for anything from handling memory to browsing.
* [Semantic Kernel](https://github.com/microsoft/semantic-kernel)
  * This toolkit developed by Microsoft in C# and Python is designed around the idea of skills and planning. That being said, at this point it also supports chaining, indexing and memory access and plugin development.
* [Guidance](https://github.com/microsoft/guidance)
  * Guidance is a more recent prompt engineering library also from Microsoft. Based on a templating language (see above) it supports many of the techniques in this post.
* [Prompt Chainer](https://promptchainer.io/)
  * Visual tool for prompt engineering
* [Auto-GPT](https://github.com/Significant-Gravitas/Auto-GPT)
  * Popular tool for designing LLM agents
* [Nemo Guardrails](https://github.com/NVIDIA/NeMo-Guardrails/tree/main)
  * Recent tool by NVidia to build rails (see above) to make sure your LLM behaves as it should
* [LlamaIndex](https://github.com/jerryjliu/llama_index)
  * Toolkit for managing the data that goes into an LLM application with mostly data connectors/tools.
* [FastRAG](https://github.com/IntelLabs/fastRAG)
    * From Intel, includes not only the basic RAG approach but also more sophisticated RAG approaches like the ones described in other sections in this post.

```
---
id: 105
title: "On open source LLMs, take two: The three laws of GenAI model evolution"
date: '2023-07-23T00:00:01+00:00'
author: Xavier
permalink: /opensourcellms2
image: /blog/images/105-0.png
header:
  teaser: https://amatriain.net/blog/images/105-0.png
reading_time:
    - ''
    - ''
categories:
    - Artificial Intelligence
    - Machine Learning
    - LLMs
    - Open Source
---
```

*In this post I summarize the main advances in the area of LLM models, and particularly open source LLMs (including Falcon, LlaMa2, and Free Willy). I describe different leaderboards and what their current top positions mean.
I enunciate the three laws of generative AI model evolution, and I ask GPT-4, Bard, and LlaMa 1 the implications of these laws.*

Five months ago, I [wrote](https://amatriain.net/blog/rlaif) about what open source LLMs meant for the future of AI. Here is what I had to say “back in the days”:

1. The best LLMs are going to be much better in all the different dimensions you probably care about (including e.g. less hallucinations)
2. The best open-source LLMs are going to be better than the best non-open source LLMs nowadays. As an example, Facebook AI made a big announcement for their LLaMA open source model a couple of weeks back . A few days later Google announced their Flan-UL2 20B model. The latter is much better. Not as good as GPT-3.5 yet, but getting close.
3. While the best LLMs in (1) are going to continue to grow in size, there are going to be models that are much smaller than the current best models yet have better capabilities. Some of them might be open sourced in (2).

Of course, a lot of things happen in five AI months! Note that these words were written a few days even before GPT4 was publicly announced! I think they still hold very true so I will stick with my original prediction, but the purpose of this post is to update to the current day. So, let’s start with what has happened since then.

# What’s new?

First, as already mentioned, GPT4. GPT4 blew everyone’s mind as I expected and knew because of the early access I had had for months.
GPT4 is still the SOTA and the one to beat that no other model has beat yet. It is, without any doubt much

better than GPT-3.5,
so my (1) statement proved true (not a big surprise, particularly given that I had access to privileged information at that time).
Maybe the only model that has gotten close to challenging its supremacy is [Claude 2](https://www.anthropic.com/index/claude-2) announced
by Anthropic only a few weeks back, but that model is also proprietary. So, “best models are proprietary” still holds.

There have been many advances in the open source arena, but two need to be underscored:

1. [Falcon LLM](https://falconllm.tii.ae/) was announced initially in March, and its largest 40B version open sourced in May. Falcon quickly moved to the top of the leaderboards. Furthermore, it challenged many of the previous assumptions about where these open sourced models would come from. Falcon came from a research institute in the United Arab Emirates. So, not a tech company, and not even a US institution!
2. [LlaMa 2](https://ai.meta.com/llama/) (the new version of LlaMa) was announced by Meta last week. The largest version of the model has 70B parameters and is comparable to GPT4 in some dimensions.
3. In just 2 days, Stability.ai presented a new fine-tuned version of LlaMa 2, [Free Willy](https://stability.ai/blog/freewilly-large-instruction-fine-tuned-models). This model beats all the other open source models, including LlaMa 2 on almost all benchmarks.

So, what do these really important milestones mean in relation to my original prediction? Well, we are getting open source models that are (far) better than the ones we had at the time of the writing. So, statement (2) was also correct.

Also, while models like GPT4 are huge and larger than anything that was public 5 months ago, we have models such as Falcon and LlaMa that are much smaller in size than GPT-3.5 (175B), yet much better. Again, statement (3) was also correct.

# A few words on benchmarks and leaderboards

I am making some assertions above that might convey that evaluating models is easy and straightforward. However that is far from true. There are quite a few different
benchmarks and leaderboards, and most don’t have overlapping models, so some conclusions need to be inferred and extrapolated from them. Let’s take a closer look at how
the different benchmarks look right now.

Huggingface maintains an amazing and very up-to-date [leaderboard](https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard), but it only
has open source models. At the time of this writing it is dominated by Free Willy and Llama2.

<img src="/blog/images/105-0.png">
The Huggingface Open Source LLM leaderboard as of today (07/23/23)

Stanford’s HAI maintains [HELM](https://crfm.stanford.edu/helm/latest/?group=core_scenarios), a very comprehensive leaderboard that evaluates models
(both open source and proprietary) across many scenarios. That leaderboard is dominated by proprietary models from OpenAI,
Cohere, and Anthropic (in this order). However, it does not include the latest models from either open source or proprietary sources.

<img src="/blog/images/105-1.png">
Stanford's HAI LLM leaderboard as of today (07/23/23)

Similarly, lmsys maintains a [leaderboard](https://chat.lmsys.org/?leaderboard) that includes both open source and proprietary models but is not updated
so quickly as Huggingface’s. It is based on three different benchmarks and is currently dominated by GPT4 with Claude from Anthropic at a far distance
(but note Claude2 is known to be much better than v1).

<img src="/blog/images/105-2.png">
LMSys LLM leaderboard as of today (07/23/23)

One of the benchmarks included in the lmsys leaderboard is MMLU (multi-task language understanding), which, in its own also maintains a
[leaderboard](https://paperswithcode.com/sota/multi-task-language-understanding-on-mmlu) where GPT4 also comes on top.

<img src="/blog/images/105-3.png">
Multi-task Language Understanding LLM leaderboard as of today (07/23/23)

Besides those benchmarks, every new model usually comes with its own technical report where they compare to other SOTA models. As an example,
[PALM-2](https://arxiv.org/abs/2305.10403), the new Google model used by Bard, beats GPT-4 and SOTA at most tasks:

<img src="/blog/images/105-4.png">

Palm-2 compared to SOTA and other models on different NLU tasks

# The three laws of Generative AI model evolution

So, given this immaculate track record backed by only one data point, I am going to do what people often do in this situation: turn the statements into universal laws.. These laws will stand as long as they don’t get proved wrong at some point in the future. So, please keep me honest and tell me if and when they do!

Here we go, **Xavi’s three laws of GenAI model evolution**:

1. Best models will continue to get better in all measurable ways [2]
2. Best open source models are going to be better than proprietary ones six months earlier [3]
3. In the near future we will see models that are superior AND smaller than current ones. Best models in (1) will continue to grow in size.

[1] While I am writing the laws specifically thinking about LLMs these laws should apply to any GenAI model. Also, I am sure everyone would agree that LLMs are just a stepping stone to more multimodal AI models that will not only include the current text-to-image or text-to-anything, but will seamlessly integrate all modalities.

[2] There will come a time when we can’t measure any more progress because we no longer care about it (or models have become so much smarter than us that we can’t). I will leave the discussion of when that will happen out of the scope of this post. Note that

[3] One may ask if this N=6 months is a constant or it will get shorter/longer over time. My take is that this is roughly going to remain constant until we reach the “singularity” point described in [2], at which point this question won’t matter much anyways.

I hope it is clear that I am writing these laws as a way to engage in constructive discussion about what the future will bring us. I would love to hear your thoughts and feedback.

# Appendix: What LLMs think of the three laws

I wrote the blog post with the help of both GPT4 and Bard. Then I asked them both what they thought the implications of the three laws were. Here are the answers:

GPT-4

<img src="/blog/images/105-5.png">

Bard (Palm 2)

<img src="/blog/images/105-6.png">

LlaMa 2 70B (through [Perplexity.ai](https://perplexity.ai) chat interface)

<img src="/blog/images/105-7.png">

```
---
id: 106
title: "Mitigating LLM Hallucinations: a multifaceted approach"
date: '2023-09-16T00:00:01+00:00'
author: Xavier
permalink: /hallucinations
image: /blog/images/106-0.png
header:
  teaser: https://amatriain.net/blog/images/106-0.png
reading_time:
    - ''
    - ''
categories:
    - Artificial Intelligence
    - LLMs
    - Prompt Engineering
---
```

    * [Model Choices](#modeling)
    * [Reinforcement Learning from Human Feedback (RLHF)](#rlhf)
    * [Domain adaptation through Fine-Tuning](#finetuning)
* [Conclusion: Yann vs. Ilya](#conclusion)

# <a name="introduction"></a>Introduction
Ever been curious about the complexities of integrating large language models (LLMs) into user-facing products? One challenge that has been gaining attention is the occurrence of 'hallucinations'—situations where these advanced AI systems produce misleading or incorrect information. This is a real-world issue that many of us in the tech industry are actively working to address as we develop new features and services.

In this blog post, you'll find a comprehensive guide to the most effective strategies for mitigating these hallucinations in user-facing products. The field is fast-evolving, so while I don't plan on continuously updating this post, I hope it serves as a valuable snapshot of current best practices. I'm also open to your insights and ideas, so feel free to reach out with any suggestions or questions you might have.

<img src="/blog/images/106-0.png">
A multifaceted approach to mitigating LLM hallucinations

# <a name="definition"></a>What we talk about when we talk about hallucinations

In the context of Large Language Models (LLMs), the term "hallucinations" often surfaces. As defined by the "Survey of Hallucination in Natural Language Generation" paper, a hallucination in an LLM refers to "the generation of content that is nonsensical or unfaithful to the provided source."

## <a name="controversy"></a>A Controversial Term: Unpacking the Use of "Hallucination" in AI

According to [Wikipedia](https://en.wikipedia.org/wiki/Hallucination), a hallucination is defined as "a perception in the absence of an external stimulus that has the qualities of a real perception." Such a description might evoke images of mysterious visions or imagined sounds. However, the term has taken on a different, though not uncontroversial, shade of meaning in the realm of artificial intelligence.

There are three main concerns I've come across regarding the use of "hallucination" to describe phenomena in AI systems:
* **Misattribution of Properties:** The application of "hallucination" might inadvertently suggest that LLMs possess some form of consciousness or perception, which they certainly don't. LLMs generate text based on patterns in their training data, not because they "perceive" or "imagine" in the way living creatures do.
* **Misunderstanding of Dynamics:** Such terminology might cloud understanding about how LLMs function. They don't "see" or "imagine." Instead, they churn out text based on statistical patterns from their training data.
* **Ethical Implications:** There's a fear that describing AI outputs as "hallucinations" trivializes the potential risks of LLMs providing incorrect or misleading information, especially if users over-rely on these models without proper fact-checking.

However, the AI context for "hallucination" has even been acknowledged by dictionaries. For instance, [Merriam-Webster](https://www.merriam-webster.com/dictionary/hallucination) defines it in the context of AI as “a plausible but false or misleading response generated by an artificial intelligence algorithm.”

Interestingly, this term isn't freshly minted. Andrej Karpathy [suggested](https://twitter.com/karpathy/status/1702916988891193460) that he might have popularized the term in his enlightening [blog post](https://karpathy.github.io/2015/05/21/rnn-effectiveness/) from 2015. But a little digging reveals earlier uses. Notably, an [ACL conference paper](https://aclanthology.org/W14-1617.pdf) from 2014 discussed "hallucinating" translations. Even further back, a 2009 paper titled ["Review Sentiment Scoring via a Parse-and-Paraphrase Paradigm"](https://aclanthology.org/D09-1017.pdf) used the term in the context of "hallucinating" topics. But perhaps the most ancient reference I found was in a 1996 paper, ["Text Databases and Information Retrieval"](https://dl.acm.org/doi/pdf/10.1145/234313.234371), which talked about systems that could "hallucinate" words not present in the original document.

In my view, it's clear that "hallucination" in AI has been in the lexicon for some time, carving out a niche meaning, distinct from its psychological roots.

It's also worth noting that AI is replete with terms borrowed from human analogies - take "neural networks" for instance. Despite initial reservations, these terms have become integral, largely uncontroversial components of AI discourse.

## <a name="types"></a>Types of Hallucinations

Hallucinations can be categorized into two main types:

* **Intrinsic Hallucinations:** These directly contradict the source material, introducing factual inaccuracies or logical inconsistencies.
* **Extrinsic Hallucinations:** These do not contradict but also cannot be verified against the source, adding elements that could be considered speculative or unconfirmable.

## <a name="source"></a>The Nuanced Role of the 'Source'

The concept of a 'source' varies depending on the specific task an LLM is performing. In dialogue-based tasks, the source can be considered as 'world knowledge.' However, when the task involves text summarization, the source is the input text itself. This is a critical nuance that significantly impacts both the evaluation and interpretation of hallucinations.

## <a name="context"></a>Contextual Importance of Hallucinations

The implications of hallucinations are highly context-dependent. For example, in creative applications such as poem-writing, the presence of hallucinations may not only be acceptable but could potentially enrich the output.

## <a name="why"></a>Why do LLMs hallucinate

It is important to first keep in mind that LLMs have been pre-trained to predict tokens. They do not have a notion of true/false or correct/incorrect, but rather base their text generation on probabilities. While that leads to some unexpected reasoning abilities (such as being able to pass the legal BAR exam or the medical USMLE), that is only a result of this probabilistic token by token reasoning. To be fair, the additional training steps of instruct tuning and RLHF that most modern LLMs have do introduce a bit more “bias towards factuality”, but they do not change the overal underlying mechanism and its pitfalls.

LLMs have been trained on the whole internet, book collections, question/answers, and Wikipedia, among many others. They have good and not-so-good knowledge in their training set. Their responses are biased towards whatever they have seen the most. If you ask an LLM a medical question and you are not careful on how you prompt it, you might get an answer that is mostly aligned to the best medical literature or to random Reddit threads.

In a recent paper entitled [“Sources of Hallucination by Large Language Models on Inference Tasks”](https://arxiv.org/abs/2305.14552), the authors show how hallucinations are originated by two aspects of the LLM’s training dataset: veracity prior and the relative frequency heuristic.

# <a name="measurement"></a>How to Measure Hallucinations in Large Language Models

Understanding hallucinations is one thing, but quantifying them? That's where things get really interesting. Quantitative metrics are essential for assessing the effectiveness of mitigation strategies. In this section, I'll guide you through the recommended methodologies for measuring hallucinations.

## <a name="measurementapproach"></a>A Five-Step Approach to Quantitative Measurement

Based on best practices in the field, here's a systematic five-step approach to accurately measure hallucinations:

1. Identify Grounding Data: Grounding data serves as the benchmark for what the LLM should produce. The choice of grounding data varies by use-case. For instance, actual resumes could serve as grounding data when generating resume-related information. On the other hand, search engine results could be used for web-based queries.

2. Create Measurement Test Sets: These sets usually consist of input/output pairs and may include human-LLM conversations, depending on the application. Ideally, you'd have at least two types of test sets:
    * A generic or random test set
    * An adversarial test set, generated from red-teaming exercises to include challenging or high-risk edge cases.

3. Extract Claims: After preparing the test sets, the next step is to extract claims made by the LLM. This can be done manually, through rule-based methods, or even using machine learning models. Each technique has its pros and cons, which we will explore in detail.

4. Validate Against Grounding Data: Validation ensures that the LLM's generated content aligns with the grounding data. This step often mirrors the extraction methods used previously.

5. Report Metrics: The "Grounding Defect Rate" is a fundamental metric that quantifies the ratio of ungrounded responses to the total number of generated outputs. Additional metrics will be discussed later for a more nuanced evaluation.

## <a name="metricsandmethods"></a>Evaluating Hallucinations: Common Metrics and Methodologies

Quantifying hallucinations in Large Language Models isn't just about recognizing that they exist—it's about measuring them rigorously. In this section, I'll delve into the different types of metrics commonly employed for this purpose.

### <a name="statistical"></a>Statistical Metrics

Metrics like ROUGE and BLEU are often the go-to choices for text similarity evaluations. They focus on the intrinsic type of hallucinations by comparing the generated output against a source. Advanced metrics such as PARENT, PARENT-T, and Knowledge F1 come into play when a structured knowledge source is available. However, these metrics have limitations: they primarily focus on intrinsic hallucinations and can falter when capturing syntactic and semantic nuances.

### <a name="modelbased"></a>Model-Based Metrics

Model-based metrics leverage neural networks, making them more adaptable to syntactic and semantic complexities. They come in various flavors:

IE-based Metrics: These use Information Extraction (IE) models to distill the knowledge into a simpler relational tuple format—think subject, relation, object. The model then validates these tuples against those extracted from the source or reference.

* **QA-based Metrics:** These implicitly measure the overlap or consistency between the generated content and the source. If the content is factually consistent with the source, similar answers will be generated to the same questions. (see e.g. “Evaluating Factual Consistency in Knowledge-Grounded Dialogues via Question Generation and Question Answering”)
* **NLI-based Metrics:** Utilizing Natural Language Inference (NLI) datasets, these metrics determine if a generated "hypothesis" is true, false, or undetermined given a "premise".(see e.g. “Evaluating Groundedness in Dialogue Systems: The BEGIN Benchmark”).
* **Faithfulness Classification Metrics:** These improve upon NLI-based metrics by creating task-specific datasets, thereby providing a more nuanced evaluation. . (see e.g. “Rome was built in 1776: A Case Study on Factual Correctness in Knowledge-Grounded Response Generation”).

### <a name="humanevaluation"></a>The Role of Human Evaluation

Despite the sophistication of automated metrics, human evaluation still holds significant value. Two primary approaches are commonly employed:

1. **Scoring:** Human annotators assign scores within a defined range to rate the level of hallucination.
1. **Comparing:** Here, human annotators evaluate the generated content against baselines or ground-truth references, providing an additional layer of validation.

### <a name="factscore"></a>The example of FActScore

[FActScore](https://arxiv.org/abs/2305.14251) is a recent example of a metric that can be used both for human and model-based evaluation. The metric breaks an LLM generation into "atomic facts". The final score is computed as the sum of the accuracy of each atomic fact, giving each of them equal weight. Accuracy is a binary number that simply states whether the atomic fact is supported by the source. The authors implement different automation strategies that use LLMs to estimate this metric.

<img src="/blog/images/106-9.png">

# <a name="redteaming"></a>The Art of Red Teaming: Best Practices for Stress-Testing LLMs

While statistical and model-based metrics are indispensable for measuring hallucinations in LLMs, it's equally important to put these models through the rigor of human evaluation. Red teaming provides an essential layer of scrutiny that complements systematic measurement. Here are some best practices to follow:

Keep Red Teaming Complementary: Although red teaming and stress-testing are invaluable tools, they should not replace systematic measurement. They are meant to augment, not substitute.

Test in Real-world Conditions: Whenever possible, conduct your testing on the production endpoint. This allows for a more realistic assessment of how the model behaves under actual conditions.

Define Harms and Guidelines: Clearly outline the potential harms and provide specific guidelines to the testers. This ensures that everyone is aligned on what to look for during testing.

Prioritize Your Focus Areas: Identify the key features, harms, and scenarios that should be prioritized in the red teaming exercise. This focused approach yields more actionable insights.

Diverse and Skilled Testers: A diverse set of testers with different areas of expertise can provide a multi-faceted evaluation. Diversity here can mean different domains of knowledge, different cultural backgrounds, or even different biases.

Documentation is Key: Decide in advance what kinds of data or findings you'd like your testers to document. Clear documentation aids in a more structured evaluation process.

Manage Tester Time and Well-being: Determine how much time each tester should ideally dedicate to the task. Moreover, be cognizant of potential burnout or a decline in creativity over time, and plan accordingly.

New approaches to red teaming include using an LLM to read team another LLM. See e.g. Deepmind's [“Red Teaming Language Models with Language Models”](https://arxiv.org/abs/2202.03286)

# <a name="mitigation"></a>Mitigating Hallucinations in Large Language Models: A Multifaceted Approach

The road to minimizing hallucinations is paved with both challenges and opportunities. In this section, we'll explore various mitigation strategies that can be customized to fit the unique demands of different applications of large language models.

## <a name="product"></a>Leverage Product Design to Minimize Impact

The first piece of advice is straightforward: if possible, design your use case in such a way that hallucinations become a non-issue. For instance, in applications that generate written content, focusing on opinion pieces rather than factual articles may naturally lower the risk of problematic hallucinations.

### <a name="productlevel"></a>Product-Level Recommendations

* **User Editability:** Allow users to edit AI-generated outputs. This not only adds an extra layer of scrutiny but also improves the overall reliability of the content.
* **User Responsibility:** Make it clear that users are ultimately responsible for the content that is generated and published.
* **Citations and References:** Enabling a feature that incorporates citations can serve as a safety net, helping users verify the information before disseminating it.
* **User Optionality:** Offer various operational modes, such as a "precision" mode that uses a more accurate (but computationally expensive) model.
* **User Feedback:** Implement a feedback mechanism where users can flag generated content as inaccurate, harmful, or incomplete. This data can be invaluable for refining the model in future iterations.
* **Limit Output and Turns:** Be mindful of the length and complexity of generated responses, as longer and more complex outputs have a higher chance of producing hallucinations.
* **Structured Input/Output:** Consider using structured fields instead of free-form text to lower the risk of hallucinations. For example, if the application involves resume generation, predefined fields for educational background, work experience, and skills could be beneficial.

### <a name="datalevel"></a>Data Practices for Continuous Improvement

* **Maintain a Tracking Set:** A dynamic database should be maintained to log different types of hallucinations along with the necessary information to reproduce them. This can serve as a powerful tool for regression testing.
* **Privacy and Trust:** Given that the tracking set may contain sensitive data, adhere to best practices for data privacy and security.

## <a name="promptengineering"></a>Prompt Engineering: Mastering the Art of Metaprompt Design

Although large language models (LLMs) have come a long way, they are not yet perfect—especially when it comes to grounding their responses. That's why understanding and effectively utilizing metaprompts can make a world of difference. A study revealed that simply instructing the LLM on what not to do could lower hallucination rates dramatically. Even better, guiding the model towards alternative actions slashed these rates further.

## <a name="generalguidelines"></a>General Guidelines to Curb Hallucinations

* **Simplify Complex Tasks:** Break down intricate actions into simpler steps.
* **Harness Affordances:** Utilize built-in functions within your metaprompt.
* **Use Few-Shot Learning:** Include examples when you can.
* **Iterative Refinement:** Don't hesitate to tweak the model's output.

One important thing to note is that while these techniques improve grounding, they also come at a computational cost. Anyone leveraging LLMs in product design will need to balance this trade-off carefully.

## <a name="finetunemetaprompt"></a>Fine-Tuning Your Metaprompts

* **Assertive Tone:** Using ALL CAPS and highlighting certain directives can improve model compliance.
* **Context is King:** Providing more background information can better ground the model.
* **Refinement Steps:** Reevaluate the initial output and make necessary adjustments.
* **Inline Citations:** Ask the model to substantiate its claims.
* **Framing:** Approaching tasks as summarization often yields more grounded results compared to question-answering.
* **Selective Grounding:** Ascertain scenarios where grounding is a must versus where it may be optional.
* **Reiterate Key Points:** Repeating essential instructions at the end of the prompt can underline their importance.
* **Echoing Input:** Request the model to recap vital input details, ensuring alignment with the source data.
* **Algorithmic Filtering:** Utilize algorithms to sift through and prioritize the most relevant information.

In upcoming sections, we'll dissect advanced metaprompting techniques, such as the "chain of thought" approach, and delve into how Retrieval-Augmented Generation (RAG) can be leveraged for better grounding.

## <a name="cot"></a>Chain of Thought

Chain of thought was initially described in the [“Chain-of-Thought Prompting Elicits Reasoning in Large Language Models”](https://arxiv.org/abs/2201.11903) paper by Google researchers. The simple idea here is that given that LLMs have been trained to predict tokens and not explicitly reason, you can get them closer to reasoning if you specify those required reasoning steps. Here is a simple example from the original paper:

<img src="/blog/images/106-1.png">

Note that in this case the “required reasoning steps” are given in the example in blue. This is the so-called “Manual CoT”. There are in fact two ways of doing basic chain of thought prompting (see below). In the basic one, called zero-shot CoT, you simply ask the LLM to “think step by step”. In the more complex version, called “manual CoT” you have to give the LLM examples of thinking step by step to illustrate how to reason. Manual prompting is more effective, but harder to scale and maintain.

<img src="/blog/images/106-2.png">

CoT is just a more structured approach to the “simplify complex tasks” generic recommendation above and is known to mitigate hallucinations in many situations.

## <a name="rag"></a>Grounding with RAG

Retrieval-Augmented Generation, commonly known as RAG, is a technique aimed at augmenting the capabilities of Large Language Models (LLMs). Initially [presented](https://ai.facebook.com/blog/retrieval-augmented-generation-streamlining-the-creation-of-intelligent-natural-language-processing-models/) by Facebook in 2020 in the context of their BART model, RAG has since been incorporated as a feature in the [Hugging Face library](https://huggingface.co/docs/transformers/model_doc/rag).

### <a name="ragcore"></a>The Core Concept

The fundamental idea behind RAG is straightforward: it merges a retrieval component with a generative component, allowing the two to complement each other. This process is visually explained in the diagram below, extracted from the original research paper.

<img src="/blog/images/106-3.png">

By combining these two elements, RAG enables the LLM to access and incorporate external information, thereby grounding the generated content more effectively. The retrieval component fetches relevant data, while the generative aspect of the model synthesizes this data into coherent and contextually appropriate responses.

RAG has evolved to become an indispensable part of the prompt engineer's toolkit. Over time, it has expanded into more complex applications, effectively serving as a concrete example within the broader framework of Toolkits, where the "tool" is typically a straightforward retriever or query engine.

Because RAG **grounds** the response to the LLM to external data, it is known to be a very effective technique to mitigate hallucinations. However, there are some caveats.

### <a name="ragcaveats"></a>RAG known caveats and guardrails

The Pitfall of Over-Reliance. One significant drawback of using RAG is a pronounced over-reliance on the retrieval results, which can, in certain cases, lead to hallucinations. It's crucial to understand that retrieval might produce results that are either empty, incorrect, or require further disambiguation. Below are strategies to handle each of these scenarios.

Empty Results: Be Prepared for Voids. When the retrieval engine returns empty results, it could either be due to a lack of relevant data in the document source or an incorrect query formulation. Meta-prompts should be designed to anticipate and guard against this scenario. If the retrieval engine returns no results, the system should opt for caution and decline to answer, stating something along the lines of, "Sorry, we don't have enough information on this topic. Could you please rephrase your question?" More advanced strategies might involve internally reformulating the query to handle issues like user misspellings, which can lead to void results.

Ambiguous Results: Seek Clarification. For ambiguous queries such as "What is a good restaurant in Portland?", where Portland could refer to multiple locations, it's advisable to seek further clarification from the user. For example, "Did you mean Portland, OR, or Portland, ME?"

Wrong Results: Navigate Carefully. Incorrect retrieval results are particularly challenging to address because they are difficult to identify without an external ground truth. While improving the accuracy of retrieval engines is a complex problem that's beyond the scope of this document, we recommend analyzing the performance of your retrieval solution within your application's specific use cases. Design your prompts to be extra cautious in areas where the retrieval engine has been identified to be less accurate.

## <a name="advancedprompting"></a>Advanced Prompt Engineering methods

Over the past few months, significant efforts have been directed towards mitigating the issues of hallucinations and grounding in Large Language Models (LLMs). These endeavors have led to a variety of innovative approaches that tackle the problem from a prompt engineering perspective. It's important to note that these advanced methods are distinctly different from the more straightforward "design tricks" discussed earlier. I will give a few examples of advanced prompt engineering methods that are relevant in the context of preventing hallucination. If you are interested in a more comprehensive catalog, check my previous post [“Prompt Engineering 201: Advanced methods and toolkits”](https://amatriain.net/blog/prompt201)

Complexity, Latency, and Cost: Advanced prompt engineering techniques often introduce additional complexity, latency, and cost, primarily because they frequently involve making multiple calls to the LLM. However, it's crucial to grasp their functionality and to have these advanced methods in your prompt engineering toolbox.

Trade-offs and Opportunities: In some cases, the incremental costs and latency might be justifiable, given the improvement in
grounding and reduction in hallucinations. Additionally, you may find opportunities to implement some of these advanced methods using smaller, more cost-effective models. This could offer a valuable compromise between performance and expense.

By understanding these advanced prompt engineering methods, you can make more informed decisions about when and how to apply them, and whether their benefits outweigh their costs for your specific application.

### <a name="selfconsistency"></a>Self-consistency

Self consistency, introduced in the paper [“SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models”](https://arxiv.org/abs/2303.08896), is a method to use an LLM to fact-check itself. The idea is a simple ensemble-based approach where the LLM is asked to generate several responses to the same prompt. The consistency between those responses indicates how accurate the response may be.

<img src="/blog/images/106-4.png">

The diagram above illustrates the approach in a QA scenario. In this case, the “consistency” is measured by the number of answers to passages that agree with the overall answer. However, the authors introduce two other measures of consistency (BERT-scores, and n-gram), and a fourth one that combines the three.

### <a name="react"></a>Reason and act (React)

React is a specific approach to designing agents introduced by Google in [“ReAct: Synergizing Reasoning and Acting in Language Models”](https://www.promptingguide.ai/techniques/react). This method prompts the LLM to generate both verbal reasoning traces and actions in an interleaved manner, which allows the model to perform dynamic reasoning. Importantly, the authors find that the React approach reduces hallucination from CoT. However, this increase in groundedness and trustworthiness, also comes at the cost of slightly reduced flexibility in reasoning steps (see the paper for more details).

<img src="/blog/images/106-5.png">

### <a name="reflection"></a>Reflection

In the Self-consistency approach we saw how LLMs can be used to infer the confidence in a response. In that approach, confidence is measured as a by-product of how similar several responses to the same question are. Reflection goes a step further and tries to answer the question of whether (or how) we can ask an LLM directly about the confidence in its response. As [Eric Jang puts it](https://evjang.com/2023/03/26/self-reflection.html), there is “some preliminary evidence that GPT-4 possesses some ability to edit own prior generations based on reasoning whether their output makes sense”.

The Reflexion [paper](https://arxiv.org/abs/2303.11366) proposes an approach defined as “reinforcement via verbal reflection” with different components. The actor, an LLM itself, produces a trajectory (hypothesis). The evaluator produces a score on how good that hypothesis is. The self reflection component produces a summary that is stored in memory. The process is repeated iteratively until the Evaluator decides it has a “good enough” answer. The authors show through experiments how reflection greatly improves the ability of detecting hallucinations even when compared to a ReAct agent.

<img src="/blog/images/106-6.png">

### <a name="dera"></a>Dialog-Enabled Resolving Agents (DERA)

[DERA](https://arxiv.org/abs/2303.17071), developed by my former team at Curai Health for their specific healthcare approach, defines different agents that, in the context of a dialog, take different roles. In the case of high stakes situations like a medical conversation, it pays off to define a set of “Researchers” and a “Decider”. The main difference here is that the Researchers operate in parallel vs. the Reflexion Actors that operate sequentially only if the Evaluator decides.

<img src="/blog/images/106-7.png">

### <a name="cove"></a>Chain-of-Verification (COVE)

[COVE](https://arxiv.org/abs/2309.11495), recently presented by Meta, presents yet another variation on using different instances of the LLM to produce several responses and self-validate. In their approach, illustrated in the figure below, the model first (i) drafts an initial response; then (ii) plans verification questions to fact-check its draft; (iii) answers those questions independently so the answers are not biased by other responses; and (iv) generates its final verified response.

<img src="/blog/images/106-8.png">

### <a name="rails"></a>Rails

A [rail](https://github.com/NVIDIA/NeMo-Guardrails/blob/main/docs/README.md) is simply a programmable way to control the output of an LLM. Rails are specified using Colang, a simple modeling language, and Canonical Forms, templates to standardize natural language sentences (see [here](https://github.com/NVIDIA/NeMo-Guardrails/blob/main/docs/getting_started/hello-world.md))

Using rails, one can implement ways to have the LLM behave in a particular way. Of particular interest to our discussion, there is a rail to minimize hallucination (Fact checking rail).

### <a name="rails"></a>Guidance (Constrained Prompting)

“Constrained Prompting” is a term recently [introduced](https://youtu.be/bZQun8Y4L2A?t=2093) by Andrej Karpathy to describe approaches and languages that allow us to interleave generation, prompting, and logical control in an LLM flow.

[Guidance](https://github.com/microsoft/guidance) is the only example of such an approach that I know although one could argue that React is also a constrained prompting approach. The tool is not so much a prompting approach but rather a “prompting language”. Using guidance templates, you can pretty much implement most if not all the approaches in this post. Guidance uses a syntax based on [Handlebars](https://handlebarsjs.com/) that allows to interleave prompting and generation, as well as manage logical control flow and variables. Because Guidance programs are declared in the exact linear order that they will be executed, the LLM can, at any point, be used to generate text or make logical decisions.

## <a name="modeling"></a>Model Choices for Mitigating Hallucinations

### <a name="size"></a>Size and Model Complexity as a General Heuristic

A well-accepted guideline within the field suggests that larger, more complex models typically offer superior grounding capabilities. For example, empirical evaluations have shown that GPT-4 substantially outperforms its predecessor, GPT-3.5, in reducing the occurrence of hallucinations.

### <a name="temperature"></a>The Significance of Model Temperature

Model temperature serves as a critical hyperparameter that influences the stochastic behavior of the model's output. In a nutshell, it determines the level of randomness when predicting subsequent tokens. Higher temperatures increase the selection probabilities for tokens that are less likely, making the model's output more diverse but potentially less grounded. Conversely, a lower temperature, approaching zero, results in the model sticking more closely to high-probability tokens, generally yielding more reliable and grounded outputs.

### <a name="rlhf"></a>Leveraging Reinforcement Learning from Human Feedback (RLHF)

RLHF methods can be applied during the later stages of training to optimize for more accurate and grounded outputs. These methods have shown marked improvements in hallucination mitigation, especially for models that have undergone domain-specific fine-tuning.

### <a name="finetuning"></a>Domain adaptation through Fine-Tuning

Lastly, if you're developing for a specific application, you might want to consider fine-tuning your internal models. Fine-tuning to your own data and examples can make a world of difference in grounding your outputs and minimizing those pesky hallucinations, particularly if you want to use a smaller and more efficient LLM. As of this writing, OpenAI offers fine-tuning for GPT-3.5 Turbo and acknowledges that in some applications this can yield better results than using the much larger and expensive GPT-4.

# <a name="conclusion"></a>Conclusion

As we have seen in this discussion of hallucinations, the problem is not an easy one to solve. In fact, Yann Lecun argues that [it cannot be solved](https://spectrum.ieee.org/ai-hallucination) without a complete redesign of the underlying models (although Ilya Sutskever disagrees). I stand somewhere in between: with the current underlying technology, hallucinations are just an expected side-effect and are hard to completely rule out. However, a combination of techniques can mitigate them and make them completely acceptable for most if not all use cases. After all, as I explained in [a previous blog post](https://amatriain.net/blog/llmsdoctors), even medical doctors hallucinate!

```
---
id: 107
title: "Beyond Prompt Engineering: The Multi-Layered Cake of GenAI Development"
date: '2023-10-26T00:00:01+00:00'
author: Xavier
permalink: /multilayer
image: /blog/images/107-0.png
header:
  teaser: https://amatriain.net/blog/images/107-0.png
reading_time:
    - ''
    - ''
categories:
    - Artificial Intelligence
    - LLMs
    - Prompt Engineering
---

<img src="/blog/images/107-0.png">
*The layers of GenAI development*

(**Note:** This post used DALL-E 3 to generate the multi-layered cake diagram and GPT-4 with image
capabilities to generate most of the post)

Often, when GenAI application development is discussed, the conversation veers towards "Prompt Engineering."
Indeed, prompt engineering has garnered significant attention and is a vital component. However, as
visualized by the multi-layered cake diagram, GenAI application development is a spectrum of processes, each
equally critical.
What binds these layers together, serving as the crucial "icing" in between, is **Evaluation**. Quality
checks, hallucination detection, and RAI (Responsible AI) evaluations are seamlessly integrated at every
stage, ensuring the consistency and reliability of the model.

# The Layered Approach to GenAI Development

1. Foundational Layers:

    * **Pre-training:** The base of our GenAI cake. It's where models are given a general understanding of
language through vast amounts of data.
    * **Instruction Tuning & RLHF:** Tailoring the pre-trained models to be more specific in their responses
and actions.
    * **Fine-Tuning (Task or Domain Adaption):** Further refining the model's skills, focusing on niche
areas or specific tasks.

2. Prompting Layers:

    * **Prompting**: Crafting specific questions or commands to elicit desired outputs from the model.
    * **Meta-Prompting:** Beyond individual prompts, meta-prompting is about creating programmatically
generalized structures for prompts. It allows for adaptability and flexibility in interactions, without
needing to manually design each prompt.
    * **Chaining:** Combining multiple prompts in sequence to guide the model towards complex outputs.

3. Design & Interaction Layers:

    * **Tool Design:** Creating tools that allow users or developers to interact with and harness the power
of the GenAI model.
    * **Agent Design & Multi-Agent Design:** Designing how the GenAI system interacts, both as a standalone
agent and in multi-agent scenarios.

4. Final Presentation Layers:

    * **Product/UX Design:** The icing on the cake. Ensuring the final GenAI application is intuitive,
engaging, and user-centric.

# Conclusion
While prompt engineering remains a focal point in the GenAI development discourse, it's essential to zoom
out and appreciate the broader landscape. Each layer, from pre-training to UX design, plays a crucial role
in creating a holistic and effective GenAI product. Think of it as baking a cake: while the icing might
catch your eye, it's the combined flavors of each layer that make it truly delightful.
---
id: 108
title: "The Multimodal Generative AI Revolution is Here!"
date: '2023-10-29T00:00:01+00:00'
author: Xavier
permalink: /multimodalGAI
image: /blog/images/108-0.png
```

```
header:
  teaser: https://amatriain.net/blog/images/108-0.png
reading_time:
    - ''
    - ''
categories:
    - Artificial Intelligence
    - LLMs
    - Prompt Engineering
---

In the landscape of Generative AI (GenAI), we often find ourselves amazed at the rapidity and scale of
advancements. GPT-4 stands as a shining example, pushing the boundaries of linguistic understanding and
generation. Yet, as we move forward, a compelling new horizon emerges: the Multimodal Generative AI
Revolution. By melding GPT-4's textual capabilities with multimodality—integrating diverse data types such
as images, voice, and video—we're not just opening a door, but unleashing a tidal wave of transformative
potential that promises to redefine our digital experiences.

# Four illustrative examples

The conceptual promise of the Multimodal Generative AI Revolution is undeniably thrilling, but it's in the
practical applications that its true value becomes tangible. As we bridge the realms of text, image, voice,
and more, we begin to see direct impacts on our daily lives and interactions. Let's delve into three
illustrative examples that showcase how GPT-4's multimodal capabilities are set to redefine our experience:

**Unveiling the Mystery Box:** We've all been there, receiving a package without any inkling of what lies
inside. With GPT-4, by simply interpreting the SKU numbers on the label, it can instantly elucidate the
contents of the box. It doesn't just stop at identification. The AI could provide details, reviews, or even
suggest complementary products based on the identified item.
<img src="/blog/images/108-0.png">

**Calorie Counting Made Simple:** Picture this - you're about to dig into a plate of bananas but wonder
about the calorie count. Snap a photo, and GPT-4 will not only tell you the calories in those three bananas
but also suggest how much you should walk to burn those off. It's like having a nutritionist and personal
trainer in your pocket!
<img src="/blog/images/108-1.png">

**Diagram Analysis:** For those delving into the intricate world of neural networks, diagrams can often seem
enigmatic. Take, for instance, a Two Tower Deep Neural Network recommender system. With GPT-4's multimodal
prowess, one can upload the diagram and receive a comprehensive breakdown. The AI describes its
architecture, the intricacies of its design, and even potential applications and improvements.
<img src="/blog/images/108-2.png">

**Multimodal Post Creation:** The synergy between image generation and text generation is truly a marvel in
the multimodal AI arena. As a firsthand illustration, I employed DALL-E 3 to generate an imaginative image
of a multi-layered cake, where each distinct layer represents a different activity in developing a GenAI
application. Not stopping there, I then fed this uniquely conceptualized cake image into GPT-4, which
adeptly assisted in crafting the text for [my previous post](https://amatriain.net/blog/multilayer) . This
process not only showcases the formidable image recognition capabilities of GPT-4 but also the innovative
image generation powers of DALL-E 3, offering a comprehensive peek into the holistic potential of multimodal
AI.
<img src="/blog/images/108-3.png">

These examples are merely the tip of the iceberg. As we continue to explore and innovate, the symbiotic
relationship between humans and AI will lead to unimaginable advancements. The Multimodal Generative AI
Revolution is not just knocking on our doors; it's here, ready to reshape the future. Embrace it and let's
co-create a world where technology doesn't just complement our lives but elevates it to realms previously
thought to be the domain of science fiction.





---
id: 109
title: "My most epic runs"
date: '2023-11-12T00:00:01+00:00'
author: Xavier
permalink: /epicruns
image: /blog/images/109-0.jpg
header:
  teaser: https://amatriain.net/blog/images/109-0.jpg
reading_time:
    - ''
    - ''
categories:
    - running
---


<img src="/blog/images/109-15.jpg">
*The Big Sur Marathon is considered the most beautiful marathon in the US and [top ten in the world]
(https://marathonhandbook.com/most-beautiful-marathons-in-the-world/). However, it did not make it into my
top 10 most epic runs*



***********************

## <a name="nymarathon"></a>The 2023 NYC Marathon

Last week I ran the NYC Marathon for my 50th birthday. This was [my 18th marathon]
(https://www.athlinks.com/athletes/167221250/results), and one of the toughest. I finished in 4 hours and 1
minute, my worst time ever, and I struggled starting in about mile 10 due to stomach issues. I had to stop
four times for bathroom breaks, and I finished completely exhausted. The 3 miles I had to walk from the
finish line to my hotel were terrible. That being said, it was AWESOME. The weather was great and NY is a
special marathon. The whole city is out on the street cheering for runners. It is very emotional, and more
than once I was close to crying (or maybe I did, who knows). I hope I can be back and do a better job next
time.

<img src="/blog/images/109-0.jpg">
*At the NYC finish line with the medal. It was tough, but I got it done. (And, yes, for those that notice, I
had nipple problems too after many, many years)*

As always, every race is full of life lessons. I captured some of them in my popular [“Ten little lessons
for life I learned from running”](https://amatriain.net/blog/10-little-lessons-for-life-that-i/). It was
also a great opportunity to make friends. This time I joined an amazing group of Catalan runners, led by the
popular sports journalist Arcadi Alibes, who was running his 20th NYC marathon and 158th overall! ([Here]
(http://www.arcadi42.com/158-nova-york-5-de-novembre-de-2023) is his post on the race, in Catalan).

<img src="/blog/images/109-1.jpg">
*Catalan runners in Times Square the day before the marathon. I do not explicit endorse any political
message or commercial product in this picture :-)*

*******************

As I reflected on this run, I looked back on the many other epic runs I have done in my life and thought
hard about where I would place it on my list. The answer is that this run, as amazing and epic as it was,
would not make it to my top 10. If you wonder why, keep reading. I have sorted my most epic runs from 10 to
1, but also link them here on top in case you want to go straight to one of them. You decide how they
compare to the most epic runs by tech leaders [compiled](https://www.theinformation.com/articles/runners-
highs-techs-serial-marathoners-on-their-favorite-long-distance-runs?rc=9kfau8) not long ago by The
Information.

<img src="/blog/images/109-2.jpg">
*I have made many friends through the many years of running, but these are the folks I have shared the most
miles with and see pretty much weekly. To my left, Joaquin, Alberto, and Andres. Thanks for all the sweat!*

<ol start="10" reversed="reversed">
<li><a href="#therelay">The Relay</a></li>
<li><a href="#andorra">Andorra 70.3 Half Ironman </a></li>
<li><a href="#cim">California International Marathon </a></li>
<li><a href="#alcatraz">Escape from Alcatraz</a></li>
<li><a href="#boston">Boston Marathon </a></li>
<li><a href="#charleston">Mt. Charleston Marathon</a> </li>
<li><a href="#tripletahoe">Triple Tahoe Marathon</a></li>
<li><a href="#im">140.6 Full Ironman Santa Rosa</a> </li>
<li><a href="#quicksilver">Quicksilver 100k (2018)</a> </li>
<li><a href="#r2r2r">Grand Canyon RimToRimToRim (2015)</a> </li>
</ol>

*******************

# <a name="therelay"></a>10. The Relay (2015)

The [Golden Gate Relay](https://www.goldengaterelay.com/) (or The Relay for short), is an amazing and epic
adventure. You make a team of 12 people, who are split in 2 vans, and you run from Calistoga in Napa Valley
all the way to the Pacific Ocean in Santa Cruz. You get to run between 15 to 20 miles in 3 different legs.
Some of the legs are brutal, with huge elevation gain going over the Santa Cruz mountains. Some others are
just amazing, like crossing the Golden Gate Bridge at night. And, while a runner in your team is running,
you are either waiting for them in the van at the finish line or trying to catch some food/sleep while the
other van is running.
```

<img src="/blog/images/109-3.jpg">
The epic route from Wine Country to the Pacific

I have done The Relay 5 times, but in 2015 we managed to put together a very competitive team of Spanish runners. Our team, Korrikalaris, finished 6th out of 150 in under 23 hours. In fact, we were less than 1 minute from making the top 5! Very epic.

<img src="/blog/images/109-4.jpg">
An epic Relay team

# <a name="andorra"></a>9. Andorra 70.3 Half Ironman (2022)
If I have to pick my most epic Ironman, I have a hard time deciding between this half and the full one. Yes, the full Ironman is an epic distance, but this half Ironman in the Pyrenees was brutal. It is all in high altitude, and the bike goes twice up a first category mountain climb that is considered epic even for professional cyclists. But the worst part was swimming in high altitude!

<img src="/blog/images/109-5.jpg">
Finishing in altitude over 90F (almost 40C) after a huge elevation gain wasn't easy. But, we got it done!

I won’t go into too many details because I have a full, and pretty long, blog post with all the details about this race that you can read aout [here](https://amatriain.net/blog/andorra).

I should also add that this was not my only epic half IM. My first one in 2018 was also in altitude in Donner Lake, CA. While it wasn’t as brutal as Andorra, it also had altitude, elevation, heat, and even smoke from the nearby fires. The Donner Lake Tri would have made it into my top 20 for sure!

# <a name="cim"></a>8. California International Marathon (2012)

This was a crazy and epic marathon entirely because of the weather. In fact, I have run this same marathon several times and have some of my fastest race times in it. But, 2012 was VERY different. The day before, we already started hearing about an epic storm that was approaching Sacramento. It was a few of us (5 if I remember correctly), and we weren’t about to go back home without at least trying. The storm during the night was crazy with wind and rain hitting our hotel window. And… it didn’t get better by early morning!

I am usually not risk averse at all, but this one time I was the one telling my friends that we should not get on the race bus. We should just pack and go home. Yes, maybe we could run the race, but we would not have fun, and we could risk injury. But, you gotta do what you gotta do. My friends insisted that we should at least get on the bus, go to the starting line, and then decide.

The 26 miles on the bus were, again, brutal. The storm was still hitting hard and there was no end in sight. We got to the starting line and we stayed in the bus. Of course at that point if the race was not canceled we were going to run. And, it wasn’t. We spent the last few minutes before the rain sheltering in the porta potties. So many people had dropped out from the race because of the weather that this is the only time in a race that I think we had enough porta potties for all the runners :-)

The race did not go very well either. Besides the cold and the rain, there was a lot of water on the road. In fact, they had to change the course in a few places. Because of the water, I probably ended up changing my stride a lot. When we got to the half marathon, I wanted to quit. Well, remember that I didn't even want to start! I did not quit, but this is one of the few marathons where I even walked for some parts. In the last few miles, the storm stopped and we even saw a bit of sun when we got to the Sacramento Capitol. And, I managed to finish well below 4 hours still. Very epic.

My relation with crazy Sacramento weather doesn’t end there though. In 2021 I was going to run my second full Ironman in the city. The storm was even worse than in 2012 (WTF Sacramento?). I was on the bus ready to go to the starting line for the swim in the Sacramento river when they canceled the race for safety. It was brutal, but I think it was the right decision. I don’t think I will ever do another full Ironman, and I blame Sacramento weather for that LOL.

<img src="/blog/images/109-6.jpg">
Not very happy after the cancellation of the 2021 California Ironman

# <a name="alcatraz"></a>7. Escape from Alcatraz (2019)

I have done a few triathlons, and some of them have made it to this list. But Escape from Alcatraz is my favorite one because of the combination of being super epic and super fun. I have done much harder triathlons, and much longer, but this one is really fun while still being super epic. [Here](https://www.strava.com/activities/963128481) is my Strava activity.

What makes Escape from Alcatraz so epic is the setting… and the swim! All big races have a big build up, and a story before the story. You usually have to travel in a bus very early, wait in the cold, make sure you can hit a porta potty… Escape from Alcatraz is that, and much more. After the bus leaves you in the SF harbor, you need to get into an old ship. All athletes get into the same ship, which is pretty crowded, fun, and scary at the same time. Scary because you know you are going to have to jump into the cold Pacific from a deck that is quite high. And, you’ve hear of swimmers who get frozen because of the shock and can’t swim.

<img src="/blog/images/109-7.jpg">
I did escape fortunately!

That, and the planning that goes through your head. Escape from Alcatraz is a very technical swim and there are dozens of tricks to it. You need to know where to jump from in the ship, and, very importantly, where to swim to. No, you cannot swim towards the finish line. Currents are very strong, and you would end up in Sausalito. You have to point your swim to some different SF landmarks as you make your way to land. I was pretty scared of the swim but it ended up being one of my fastest ever since I was able to ride the currents all the way to the shore.

After the swim, things are much easier. The bike is hilly (of course it has to be in SF), but short. And, the run is around the Golden Gate, also challenging but short.

Overall, an epic experience!

# <a name="boston"></a>6. Boston Marathon (2018)

The dream for many runners is to run the Boston Marathon. That is particularly true for the great majority of runners who have to qualify for it. Qualifying in itself is already a feat. Running is supposed to be the fun part… except in 2018!

2018 was a brutal year for the Boston Marathon. A huge storm hit the city, and we ran the 26 miles in cold rain and wind. In fact, there was a combination of mud and snow in the starting line. Luckily, we knew the day before things would be pretty rough so we bought throwaway clothes and shoes and took the race clothes in a plastic bag.

<img src="/blog/images/109-8.jpg">
Miserable weather to the end… with Alberto.

By the time the race was about to start we were wet and muddy. We then changed into clean clothes and off we were! I ran the first part a bit slower than I could because I was pacing my friend Alberto. Things were brutal with cold and windy rain coming to us most of the way. However, when I hit the half marathon mark I was feeling great and started to push. I passed so many people in the second half that I had great fun. Definitely not close to my fastest time, but a 3:33 in those conditions felt amazing… and totally epic! In fact, now I realize that I even used the word epic in the title of my Strava run 🙂

# <a name="charleston"></a>5. Mt. Charleston Marathon (2017)

This was my fastest marathon in 3:08. I never thought I would get close to 3 hours in a marathon, and this one got me pretty darn close. Maybe with a little more training I would have done it. In any case, this was a totally epic run. One of my best.

First things first, this is a straight downhill marathon that is considered to be one of the fastest in the US. We chose it because we wanted to qualify for Boston, and we did by far. You can see my Strava activity [here](https://www.strava.com/activities/963128481).

<img src="/blog/images/109-9.jpg">
I did not dream up that finish time!

The marathon starts at altitude in Mt. Charleston, and ends in Las Vegas. It was freaking cold at the start line and there was even some snow, which is crazy considering you finish the race in the desert. Because of the altitude, I also struggled at the beginning. The combination of altitude and very hard pace was tough, but I got into the rhythm and hit it off with the 3:15 pacer. I felt great during the first half and even held the pacer sign while she was making a pit spot (the pacer had run a 2:20 the year prior). In mile 16 or so I decided to go for it and pushed faster than the pacer. I only struggled a bit in a small hill in mile 22, but ended up finishing strong, getting my all time PR, and my BQ (Boston qualifier). Epic!

# <a name="tripletahoe"></a>4. Triple Tahoe Marathon (2016)

Running a marathon in altitude in Lake Tahoe is already pretty epic, but, how about 3 in 3 consecutive days? Well, that is exactly what we did in 2016. Three marathons back to back that take you all around the lake in 3 days. You start in South Lake Tahoe counterclockwise and after each marathon you are taken back to the starting line with a bus. The course is not only in altitude, but also has pretty challenging elevation gain in some parts. And, to make things more epic, we ran the first two marathons on the side of the road, with normal traffic on the road.

<img src="/blog/images/109-10.jpg">
Just another long run with Alberto

To be honest, the most challenging part of the three marathons was probably the first thirty minutes when you need to get your body used to running and breathing in altitude and you think to yourself that there is no way you can finish a marathon, let alone 3. But, once I overcame the initial panic, the rest was very enjoyable. Because I wanted to finish the 3, I paced myself and did not push it too much. A fun anecdote is that in the third marathon, that had Emerald Bay half way through, I ran an inverse split and won free running shoes. Strava had a cool challenge going on at that time that gave free New Balance shoes to those that managed to run an inverse split marathon. Fwiw, this one was pretty easy since the first half was straight up, and the second half was straight down. Epic!

# <a name="im"></a>3. 140.6 Full Ironman Santa Rosa (2019)

It took me a bit over 14 hours to finish my only full Ironman. Not a great time, and I did not even have much fun, but, yes a full IM is, by definition, epic.

The swim was…well, long. It was in a beautiful lake near Santa Rosa. The worst part of it is that we had to go around the lake twice. Going out of the water half way to go through the timing chip device is not fun, but we got the swim done, which is my least favorite part of the triathlon.

The bike was again VERY long, and somewhat boring. In an Ironman you ride your bike alone since drafting is not allowed. And riding your bike alone for over 6 hours is a lot. I have done much longer ultramarathons, but you can always chat with people you find around or get a pacer if you are lucky enough. This was plain lonely bike riding except for a couple of times I stopped in an aid station to refill the bottle. Also, the road was in awful conditions, and there were cars! By the time I finished the bike I really, really was looking forward to the run. My favorite sport!

<img src="/blog/images/109-11.jpg">
At the end I was a beaten up as I look

And, the first half was pretty good all things considered. I was running on pace to go under 4 hours, which is really good in a full Ironman. But, then I started getting horrible stomach cramps due probably to poor hydration. I had to stop in the porta potties a few times and walk at times. The only positive thing I remember is getting to an aid station where they had warm broth. That felt magic. I ended running the marathon in over 5 hours. My worst marathon ever. But, very epic.

# <a name="quicksilver"></a>2. Quicksilver 100k (2018)

This is the longest run I have ever done, so it had to be high up on my list. Quicksilver(LINK) is not only long, it is pretty epic all around. It has 13k feet of elevation with some of the steepest climbs I have done in any ultra. And, it is usually very hot.

I say “usually” because I have ran the Quicksilver 100k three times, but only finished once. This race is literally in my backyard here in the Bay Area. In fact, I use parts of the course regularly for trail running. I finished the race in 2018 in a bit under 16h, which was the cutoff time for qualifying for Western Estates. It was brutal. I had my friend Alberto pace me for the last 20 miles, and I had some rough moments where I could barely speak to him and was close to giving up. But, I made it (see Strava activity [here](https://www.strava.com/activities/1568419810).

<img src="/blog/images/109-12.jpg">
Pretty happy to finish this shit. I wasn’t thinking straight after almost 16 hours of running

In case you are wondering, the other two times that I started but DNF (did not finish) it was because I had decided beforehand I would not. The first time, I had my son’s graduation that same evening, and I had promised I would be there. The second time, I was only weeks away from an Ironman, and I knew running anything beyond 40 miles could compromise the IM.

Since this is the only official ultra that has made it to the list, I should at least mention that I have ran many others, some that were amazing. I fondly remember the two North Face races in Marin County, particularly the one in 2014, the day after some strong stor had hit the course. The mud in some parts of the trail made it particularly epic!

# <a name="r2r2r"></a>1. Grand Canyon RimToRimToRim (2015)

And, the most epic run ever goes to the amazing adventure me and a group of friends had in 2015. This is not an official race, but rather a popular adventure for those into ultra-running: running from the South Rim of the Grand Canyon to the North Rim and back [LINK]. This is a 50+ miles run with a crazy 20k feet of elevation (see [Strava] here(https://www.strava.com/activities/414842609)). It is completely self-assisted, meaning that you need to carry all your food and water needs (yes, there are some water sources, but not guaranteed). Besides the huge elevation and complicated logistics, what makes this run very challenging is the huge temperature differences. In fact there are only a few weeks in the year where you can attempt, in spring and fall. Even then, you will be below freezing when you start the run well before dawn, easily hit 100F degrees in the middle of the day in the canyon, and see some snow in the north rim when you get there.

<img src="/blog/images/109-13.jpg">
The team. Ready to rock the Canyon!

We had attempted the run a year earlier in the spring. When I was going up the north rim with only one of my friends after splitting up from the rest, he did not feel well. I had to make the hard decision of turning around with him. But, we promised we would be back. And, we were! The year after!

<img src="/blog/images/109-14.jpg">
*We turned around the first time we attempted the r2r2r due to Alberto getting injured. This time around we

got all the way to the end, and he led the way and kept me going.*

This is an amazing run, but one that requires lots of training and preparation. It was crazy for us to be helping unprepared hikers who were going up the south rim in the evening after “just” having attempted going down and up the rim when we had over 40 miles on our legs.

A final thing that made this run completely epic was that when we started going up the south rim, a pretty scary storm with lighting started. Lighting storms can be very dangerous in the Grand Canyon, but fortunately, we were far enough from the lighting to make it safely, albeit very wet and muddy, to the top. It took me a bit under 11 hours to complete this amazing feat.

```
---
id: 110
title: "Beyond Singular Intelligence: Exploring Multi-Agent Systems and Multi-LoRA in the Quest for AGI"
date: '2023-11-19T00:00:01+00:00'
author: Xavier
permalink: /multiagents
image: /blog/images/110-0.png
header:
  teaser: https://amatriain.net/blog/images/110-0.png
reading_time:
    - ''
    - ''
categories:
    - Artificial Intelligence
    - LLMs
    - Prompt Engineering
---
```

In my view, the concept of [Artificial General Intelligence (AGI)](https://en.wikipedia.org/wiki/Artificial_general_intelligence) as it's commonly understood might be a misnomer. Human intelligence itself is not 'general'; it is inherently constrained by our senses, perception, and cognitive abilities. Pursuing AGI as a singular, super-human intelligence system seems flawed to me. Instead, I believe the focus should be on developing independent agents that specialize in performing specific tasks far better than humans. This shift from seeking a universal solver to nurturing a network of specialized agents is at the heart of the current evolution in AI. Technologies like Multi-LoRA and frameworks such as AutoGen and AutoAgents are leading this transformation, redefining our path to what might be the real essence of AGI.

<img src="/blog/images/110-0.png">

# Large Language Model (LLM) Agents

[LLM Agents](https://amatriain.net/blog/prompt201#agents), utilizing models like GPT-3 or GPT-4, represent a significant leap in AI capabilities for natural language understanding and generation. Beyond their ability to process and produce human-like text, these agents are capable of calling functions and using tools. This functionality allows them to perform a wide range of tasks, from generating content to coding. Moreover, LLM Agents possess the ability to plan the use of such functions and tools, enabling them to strategize and execute complex tasks more effectively. This aspect of LLM Agents aligns with the early traits of AGI, as they demonstrate an advanced level of problem-solving and adaptability, handling tasks that go beyond their initial training, as highlighted in the [Noema Magazine article](https://www.noemamag.com/artificial-general-intelligence-is-already-here/).

# LangChain's Agents

[LangChain's agents](https://python.langchain.com/docs/modules/agents/) demonstrate the power of specialized AI agents in making reasoned decisions and executing complex objectives. These agents utilize language models to decide on action sequences, adapting dynamically to user inputs and intermediate steps. A notable feature of LangChain is the LangChain Expression Language (LCEL), which simplifies the creation and management of Functions, Tools, and Agents. For those interested in a deeper understanding of these concepts, Harrison Chase offers a short course titled "Functions, Tools, Agents: LangChain," available through [Deeplearning.ai](https://www.deeplearning.ai/short-courses/functions-tools-agents-langchain/). This course provides valuable insights into LangChain's capabilities and applications, making it an essential resource for anyone interested in the practical aspects of AI agent development.

# Multi-agent frameworks

As I mentioned in my recent [“Beyond Prompt Engineering: The Multi-Layered Cake of GenAI Development”](https://amatriain.net/blog/multilayer), designing multi-agent systems will be the next frontier of AI system design. Frameworks like Auto-Gen and AutoAgents epitomize the potential of multi-agent systems. [Auto-Gen](https://arxiv.org/abs/2308.08155) in particular demonstrates the power of multi-agent systems in automating complex workflows. It leverages LLMs to break down large tasks into sub-tasks, autonomously accomplishing them using various [tools and internet resources](https://en.wikipedia.org/wiki/Auto-GPT#:~:text=Auto,4%20to%20perform%20autonomous%20tasks.%E3%80%9075%E2%80%A0%5B3%5D%E3%80%91) . Similarly, [AutoAgents](https://arxiv.org/abs/2309.17288v2) showcases the adaptability of AI systems by automatically generating and coordinating multiple specialized agents to form AI teams for diverse tasks, enhancing problem-solving capabilities and adaptability.

<img src="/blog/images/110-1.png">

# LoRA and Multi-LoRA

The concept of [Low-Rank Adaptation (LoRA)](https://arxiv.org/abs/2106.09685) is revolutionizing the fine-tuning of large language models. Instead of updating the entire model, LoRA focuses on updating low-rank additive matrices, reducing the computational load. LoRA is a specific implementation of the so-called [PEFT](https://huggingface.co/docs/peft/index), Parameter-efficient Fine Tuning approaches Multi-LoRA extends this approach, allowing numerous LoRA adapters to coexist within a single model. This innovative system, as detailed in the ["S-LoRA: Serving Thousands of Concurrent LoRA Adapters" paper](https://arxiv.org/abs/2311.03285), enables serving thousands of LoRA adapters simultaneously, dramatically improving throughput and scalability in deploying fine-tuned LLMs for a variety of applications. Also worth mentioning that Predibase has now open-sourced their [LoRA Exchange (LORAX) serving package](https://github.com/predibase/lorax).

# Redefining AGI
Contrary to traditional visions of AGI as a singular, all-knowing entity, the emerging paradigm, as suggested by the advancements in technologies like Multi-LoRA and frameworks such as AutoGen and AutoAgents, indicates that AGI will manifest as a network of specialized agents. These agents, each expert in its field and specialized in some specific tasks, contribute their expertise to a collective intelligence. This network approach, using advanced technologies and systems, offers a more dynamic and practical path to AGI.

The future of AGI is being shaped by the integration of specialized AI agents, each fine-tuned for specific purposes, working in harmony. This collaborative approach, leveraging cutting-edge technologies and frameworks, presents a more feasible and impactful path toward realizing AGI, moving beyond theoretical concepts into practical, impactful applications.

```
---
id: 111
title: "AI as the New Member of the Engineering Team: Crafting an End-to-End AI Application with AI"
date: '2023-12-20T00:00:01+00:00'
author: Xavier
permalink: /aidevelopment
image: /blog/images/111-0.png
header:
  teaser: https://amatriain.net/blog/images/111-0.png
reading_time:
    - ''
    - ''
categories:
    - Artificial Intelligence
    - LLMs
    - Prompt Engineering
    - software engineering
---
```

# Exploring the AI-Driven Future of Software Development

Embarking on an experiment that blends the boundaries of AI and software engineering, a few days back I set out to explore a provocative hypothesis. Will AI-generated code dominate software development in the near future, accounting for over 80% of all coding? This question is not just theoretical; it's a forecast based on the rapid advancements in AI, and I'm putting it to the test through a hands-on project.

<img src="/blog/images/111-0.png">

## The Challenge: AI as the Core Developer

In this unique endeavor, I wanted to construct a comprehensive state-of-the-art AI end-to-end application in mere hours, a task that traditionally would require extensive manual coding. The twist is my reliance on advanced AI, particularly ChatGPT from OpenAI, known for its robust natural language processing and coding capabilities. This experiment is more than an exploration of AI's utility in coding; it's a deep dive into the evolving synergy between AI tools and software development, challenging the conventional roles and methods in the field.

## AI: More Than a Tool, a Collaborator

ChatGPT with the latest GPT4 model in the background, isn't just a sophisticated piece of technology; it represents the pinnacle of AI's integration into creative and technical processes. My goal was for it to act not just as a tool, but as a collaborator, bringing to the table its capacity to understand, generate, and debug code. This collaboration is a glimpse into a future where AI's role in software development transcends assistance, becoming a core component of the creative and development process.

## Embarking on a Groundbreaking Journey

Join me in this exploration. This journey is more than a technical challenge; it's a quest to uncover the potential of AI in redefining the software development landscape, signifying a shift in how we conceptualize and execute coding in an AI-augmented future.

# The App and the Tools: Venturing into Unfamiliar Territory

In this segment of my journey, I decided to push my boundaries by choosing technologies I had never worked with before. The goal was to create a personalized chatbot, "Xavibot," designed to respond as if it were me. You can interact with this chatbot [here](https://amatriain.net/Xavibot).

<img src="/blog/images/111-1.png">

A Note on Performance and Feedback: Should you experience timeouts or other issues while using the bot, it's worth noting that it's hosted on a lower tier of Azure, which may affect performance. I welcome any feedback or queries at xavier at amatriain dot net.

For this project, I utilized:

- Node.js for the Backend: A deliberate choice over Python to challenge myself with an unfamiliar environment and assess ChatGPT's effectiveness in aiding with new technologies.
- React for the Front-End: Leveraging the react-chatbot-kit, I ventured into modern UI design, an area where my experience is limited.
- OpenAI Assistants APIs with GPT4: A choice driven by the desire to explore this new and powerful technology from OpenAI, with which I had no prior experience.

<img src="/blog/images/111-2.png">

The source code for the application is available on my [my Github](https://github.com/xamat/Xavibot). I encourage others to use it for their own projects or adaptations.

Deployment involved Azure for the Node.js backend and GitHub pages for the front-end. This process included new challenges for me, such as configuring CORS and managing a secrets vault for remote keys – all first-time experiences.
Reflection on the Initial Phase:

This project was not just about building a state-of-the-art AI application; it was a test of how quickly and efficiently I could adapt to new technologies with AI assistance. Despite my extensive background, I approached most of the tools used in this project as a novice. This experience sheds light on the current capabilities of AI in supporting software development, especially when diving into unfamiliar tech waters.

## OpenAI's Assistants API

The OpenAI [Assistants APIs](https://platform.openai.com/docs/assistants/overview) represent a significant advancement in chatbot development. This API simplifies the process, eliminating the need for developers to delve into complex aspects like memory management and retrieval-augmented generation (RAG), or the intricacies of prompt engineering and orchestration.

In creating "Xavibot" available [here](https://amatriain.net/Xavibot), I configured the assistant to perform RAG using two specific files, complemented by a basic prompt structure. While this was sufficient to create a functional version 0.1 of the bot, more sophisticated prompt engineering could potentially elevate its capabilities. My initial approach was to prioritize speed and simplicity in deployment.

My experience with the Assistants API has been insightful. For simple chatbot applications, it is exceptionally efficient and user-friendly. However, when it comes to applications requiring greater control and flexibility, the API shows limitations. Future experiments and developments could explore how advanced customization might overcome these constraints and expand the API's utility in more complex scenarios.

# The Positive Impact of AI in Software Development in 2023

My foray into AI-first software development has been an overwhelmingly positive experience. ChatGPT, serving as both a knowledgeable pair programmer and coach, accelerated my project's development significantly. Within just a few hours, I had a locally running application with about 80% of the intended functionality. This rapid progress was encouraging, though I was aware that perfecting the remaining 20% would be more time-consuming, adhering to the familiar power law dynamics of software development.

Enhanced by RAG and Continuous Interaction

The Retrieval-Augmented Generation (RAG) feature in ChatGPT proved exceptionally useful. Even when faced with complex queries, such as specific API usage, directing ChatGPT to relevant documentation often resulted in accurate and helpful responses.

What truly stood out was the ability to engage in a persistent dialogue with ChatGPT, iterating over problems until they were resolved. This process sometimes involved multiple refinements of my code, with

ChatGPT providing consistent and relevant suggestions. It was akin to having a dynamic, interactive version of StackOverflow, but with the added advantage of contextual understanding and memory retention.

## AI vs Traditional Resources

Interestingly, on the few occasions when I consulted StackOverflow, I found the solutions there to be less effective than ChatGPT's. This experience aligns with the growing perception of AI as a formidable tool in the realm of technical problem-solving.

## The Supportive Nature of AI

An aspect that particularly resonated with me was ChatGPT's unwaveringly supportive and positive tone. Even in moments of frustration, when I expressed dissatisfaction with the responses, ChatGPT maintained its composure, apologizing and continuing to offer alternative solutions. This emotional intelligence, often crucial in pair programming and mentorship scenarios, significantly enhances the collaborative experience.

# Navigating the Challenges of AI-Assisted Software Development

My journey with AI-first software development, while largely positive, also revealed several areas where AI, specifically ChatGPT, could be improved.

## Data Loss and Backup Issues

A significant issue was the loss of my chat history due to corruption in the ChatGPT thread, a common problem I discovered others facing as well. This not only disrupted my workflow but also erased valuable insights and progress. The lack of an integrated backup feature in OpenAI's platform is a notable gap, necessitating third-party solutions for chat backup and restoration.

## Limitations in Information Accuracy and Complexity

GPT-4's knowledge base often lacked up-to-date information, leading to recommendations of deprecated or incompatible resources, such as a React toolkit that wasn't suitable for recent React releases. Additionally, when directed to online documentation for APIs, ChatGPT's responses were sometimes vague or missed the mark. For instance, when I asked about adding a secret to the Azure vault based on Azure documentation, ChatGPT incorrectly guided me to use the wrong role assignment.

## Tendency to Overcomplicate Solutions

ChatGPT frequently suggested overly complex solutions. For example, it led me towards using React Context and Redux for a situation where a simpler global state approach would suffice. Similarly, it advised setting up an Azure vault for secrets management, which was an overkill for my specific needs. While ChatGPT eventually acknowledged simpler alternatives, the initial guidance towards more complex solutions often proved to be time-consuming.

## Difficulty with Error Analysis and Code Context

One of the most significant challenges was ChatGPT's struggle to diagnose errors in relation to the provided code. A striking example was when it suggested an extensive refactor of my code, while the issue was actually due to a misspelled variable. This inability to pinpoint simple issues, coupled with a tendency to suggest extensive code changes, often led to inefficient problem-solving.

# Conclusions: Embracing the AI Revolution in Software Development

After this deep dive into AI-first software development using ChatGPT with GPT4, I've come to appreciate its distinct advantages over other coding-specific AI tools. While solutions like GitHub Copilot have their merits, the versatility and depth of ChatGPT's assistance surpassed my expectations and demonstrated its potential as a superior tool for developers.

However, there's room for improvement. Fine-tuning AI models for specific coding tasks and integrating them more seamlessly into development environments like Visual Studio Code could address some of the challenges I encountered. Such enhancements would further streamline the development process and amplify the benefits of AI assistance.

I remain bullish about the transformative impact of AI on software development. The advancements I've witnessed and utilized are just the tip of the iceberg. For developers at any career stage, now is the time to embrace AI tools. With AI assistance, even developers with average skills can significantly boost their productivity and effectiveness, potentially becoming '10X engineers'. And for those already excelling in their field, these tools could amplify their capabilities, leading to unprecedented levels of efficiency and innovation.

In conclusion, the integration of AI into software development isn't just a trend; it's a paradigm shift. It's reshaping how we approach coding and problem-solving, offering new heights of potential for every developer willing to adapt and learn.