Posting on arXiv is good, flag planting notwithstanding.

This piece by Yoav Goldberg has been widely circulating over the Interwebz the last couple of days.

It mostly complains about the methodology used in a particular paper from MILA about text generation.

But it also complains about the habit of the deep learning community of posting papers quickly on arXiv. I vehemently disagree with that point.

I’m not going to defend the paper Yoav discusses. I haven’t read it. But a lot of Yoav’s argument sound awfully defensive to me, including the sub-title: “for fucks sake, DL people, leave language alone and stop saying you solve it” and the statement “I have a lot of respect for language. Deep-learning people seem not to”. This sounds to me a lot like what people in various communities have been saying just when neural nets/deep learning started to get good results in their field: character recognition in the early 90s, speech recognition until around 2010, computer vision until about 2014, and now NLP. I understand the reasons, but this sounds awfully like a read-guard battle, which is very surprising coming from Yoav who has been quite involved in applying deep learning to NLP.

To be fair, the piece has now been augmented by a significant amount of clarification (aka back-pedaling): https://medium.com/@yoav.goldberg/clarifications-re-adversarial-review-of-adversarial-learning-of-nat-lang-post-62acd39ebe0d

Nikos Paragios (someone who is "not that old", as he puts it) wrote a similarly defensive piece that laments the methodological shift in computer vision brought about by DL: https://www.linkedin.com/pulse/computer-vision-research-my-deep-depression-nikos-paragios

Any time one community collides with another, good things happen. At first community A views papers from community B as technically inferior and community B sees papers from A as methodologically flawed with sub-par results. It's happened over and over again with the ML community sometimes playing A (with B being speech, vision, graphics, NLP) and sometimes B (with A being statistics, optimization, and various theoretical branches of CS). Applying the criteria of one community to the works of another doesn't make sense until the communities have developed a common direction. If communities can be seen as unit vectors with different directions, the projection of A on B, is shorter than B and you may think A is inferior to B. But the converse is also true: the projection of B on A is shorter than A. It takes time for communities to develop a common language and adopt the best of each other's methodologies.

But back to arXiv. Yoav and others have been complaining that posting half-baked papers on arXiv is conducive to a kind "flag planting" behavior: authors get an idea, implement it quickly and sloppily, post so-so results on arXiv, and expect recognition for being first. Some people say that's unfair because there is much more to research than merely having the idea. That's totally true. In the evolution of an idea or a technology there is getting the idea, implementing it, showing it can work, making it work on a toy problem, making it work on a real problem as well as other methods, beating a record with it, coming up with a recipe to make it work, explaining theoretically why it works, optimizing and simplifying it, generalizing it, developing a technology around it, building a product, packaging and selling the product....

People at various points in the process should get some credit to various degrees. In
the history of science, it rarely goes exclusively to the person who had the original idea (though I've heard some people, who shall remain nameless, claim that it should).

The process that posting on arXiv gives us is simply much more efficient than the traditional model of publication. It certainly does not replace or displace traditional peer review. Yes, it changes how credit should be attributed, but that's fine.

Why is the fast arXiv posting practice more efficient? It follows the "bazaar" model of collaboration, instead of the "cathedral" model of traditional publication (read Eric Raymond's piece "The Cathedral and the Bazaar" http://www.catb.org/esr/writings/cathedral-bazaar/). In software, the bazaar model is the Linux model: release early, release often, while the cathedral model is the GNU/Hurd model: don't release until you have a perfect and bug-free piece of software. The bazaar model is incomparably more efficient.

Why? the arXiv/bazaar model is noisy and messy, but progress is faster because of more frequent feedback. This is very much like the difference between stochastic gradient and batch gradient: stochastic gradient is noisy and messy but way faster and more efficient.

I think that the fear that half-baked paper with sloppy methodology will get all the credit for an idea is misplaced. Communities will naturally recognize contributions and give credit when credit is due. It's always happen that way.