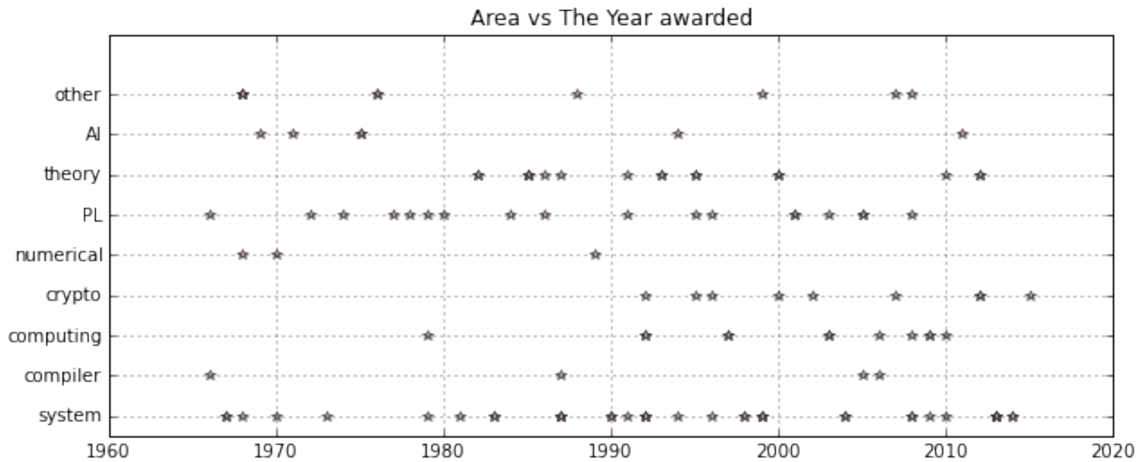
CS247r Final Project

Emma Wang

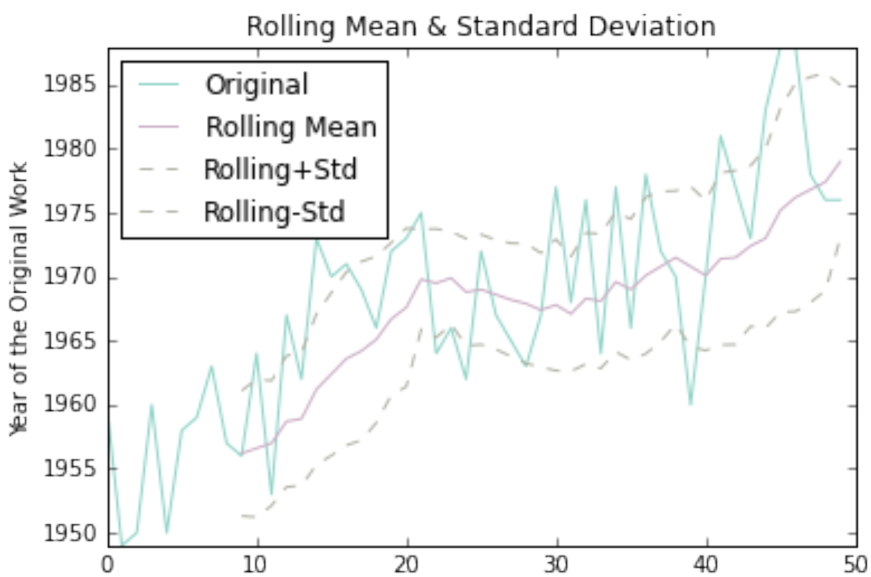
A study of Turing award

I categorize the works according to their areas. Here is a word cloud of the key words. It seems that Turing award are mostly awarded to programming language, system and cryptography researchers.

The next figure shows the works' awarded year and their areas. There are some interesting trends. Cryptography and computing are awarded more and more often. It indicates people are paying more attention to the areas. There is a decreasing trend on numerical analysis. People have always been interested in working on system related fields, including network, database, architecture, and OS.

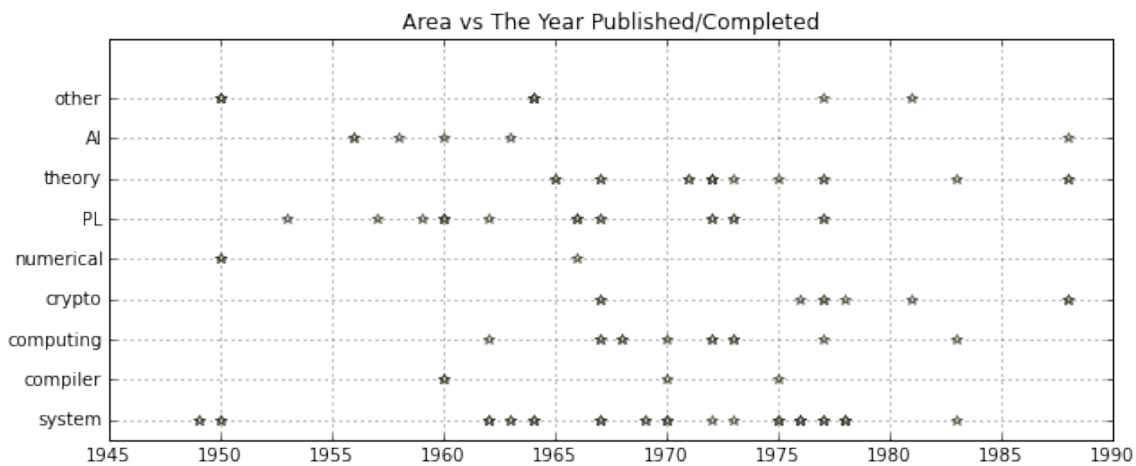


The following figure shows the years of the awarded works originally published/completed. The purple line is the rolling mean over a window of 10 years. The dashed lines are the mean plus or minus a standard derivation. The standard derivation is from a 10-year window, too. If we assume the year of the original work is a Gaussian distribution with a mean of the rolling mean and a standard derivation or the rolling standard derivation. We can predict that the next work being awarded should be completed or published between 1975 and 1985 with a confidence of 70% (1 sigma).



The next figure shows the works' areas and the year of the original works. It illustrates what areas people were interested in working on in that year. Most of the system works were completed between 1960 and 1980. Most of the Programming Language works were in 1950 and 1980. People were interested in AL mostly in 1955~1965. Since 1965, people started to look into theory, cryptography and computing.

From last figure, I deduct the next Turing award was published between 1975 and 1985. The following figure shows that between 1975 and 1985, people were likely working on system and cryptography.



As described above, within the range of this class, the next Turing award is possibly given to

Richard M. Russell for his Cray-1 Computer System or

Patterson, Gibson and Katz for their work on RAID.

A paper to add, before I was born

According to the reasons in last section, I select a paper published between 1975 and 1985 in the broad realm of system and architecture. When talking about 1980s, multiprocessors are important inventions during that period. I also notice that the parallel papers included in this course are all about vector machines, which exploit the data level parallelism. I argue that we should cover instruction level parallelism as well. Thus, I select Burton Smith's HEP paper [1] to add to the reading list.

It is the earliest paper documented multiprocessor paper. The architecture includes several processors, and they connect to each other and the memory system by sending packets through a network. The switches in the network forward the packets to the destinations. Within each processor, several processes share the computing resources by waiting in a queue and getting scheduled one after another. The memory and registers have states including full, empty and reserved, through which the memory consistency is maintained.

A paper to delete

I generally appreciate every paper I have read. If one paper should be deleted in order to add the new HEP paper, I would recommend one of the AI papers. The reason is they are too philosophical and many concepts do not have a concrete definition. I don't get much by reading them. They make me think what AI is, but reading one paper like that is probably enough. Since Turing test is a word people keep talk about these years, it may be useful to read the original paper. Thus I suggest deleting Searle's paper.

Another reason is, as discussed in the class, I don't like the Chinese understanding example. I argue that it is impossible to do answer all the questions correctly without actually understanding the language. It is exactly how the language tests are designed, e.g. GRE and TOEFL. People are asked to answer the questions after reading or listening to a short article.

A paper after I started college

I will include Sophia's Aladdin paper. I think it will be influential in the future after about 20 years.

The work which saves people a lot of time gets rewards. Through the entire computer science history, programming languages make it easier for people to give instructions to the hardware; CAD tools makes circuit designing and debugging more convenient. There are a lot of such examples. All these efforts lower down the prerequisites for people to do works. Now people need specialized hardware. Aladdin actually makes it much easier to simulate the accelerators, without going through all the circuit design, or even high-level synthesis processes.

Now is an important era for computer architecture. The perfect scaling is coming to an end. Software people can no longer program on the general-purpose machines and hope the performance will scale with the technology as it could scale before.

There is a wall between software (including algorithm) and hardware researchers. Software researchers need faster hardware for their workloads, while hardware researchers find it time consuming to fully understand a workload so that they can build most optimized hardware for it. If we are not getting the technology scaling back with novel devices, software and hardware researchers need to work closely together and provide better performance for cutting-edge applications.

With Aladdin, software researcher can run their code with Aladdin, sweep the parameters and get the performance. They then have a sense of whether it is worth to build hardware for the workload. They can talk to a hardware designer afterwards. For hardware designers, it’s okay that they are not the workload experts. They can run the workload through a workload characterization tool so that they will have a sense of how the workload behaves. The tool may also give the data that even the workload designer does not know. By the way, the tool is being developed by the same research group.

Just like history repeats itself, specialized hardware becomes important these days again as in 80s, when HT Kung published his systolic architecture papers. I believe there are more and more advanced circuit architectures being studied. One can modify Aladdin to simulate the new circuit architectures. We are having better specialized hardware in the market. The Aladdin work is actually a starting point of this important direction.