Patent

CHAU | IP

2022

Legal departments need to be able to identify quality AI patents to obtain the broadest possible protection. Here's how to do it.

Artificial intelligence (AI) represents an enormous business opportunity, with industry observers expecting the global AI economy to approach $16 trillion by 2030 . In industries from agriculture to finance, businesses are rapidly developing AI solutions that will give them an edge over their competitors. However, because the field is so young, attorneys often fail to write effective patents that properly protect these inventions.

Many mistakes that attorneys make when drafting AI patents are simply bad habits ported over from their experience drafting traditional software patents. However, there are also differences between AI and traditional programming paradigms that require a different approach. Ignoring these differences can result in additional risks and lost opportunities. The result? Weak, ineffectual patents filled with redundant claims. Poorly drafted patents are often difficult to enforce, and easy for competitors to design around. In short, they do a poor job of helping companies to protect the long-term value of their AI inventions.

At F. Chau & Associates, a boutique intellectual property law firm, we've worked with some of the world's largest and most sophisticated firms to develop best practices for drafting patents. These best practices will result in patents that are more likely to be granted and that will provide more comprehensive protection of AI inventions well into the future. The most important of these best practices focus on drafting quality claims for AI inventions.

https://www.forbes.com/sites/andrewcave/2019/06/24/-can-the-ai-economy-really-be-worth-150-trillion-by-2025/#5006222a3bf4

Attorneys drafting AI claims often make two big mistakes. First, they craft their first independent claim by laying out a list of steps (i.e., an “Algorithm Claim”), and then they repeat the same claim three times as though it were a magical incantation. Each time the claim is repeated they append a new preamble, ostensibly to protect a different “statutory class,” but the scope of protection is essentially the same.

While this approach is easier for attorneys to execute, it fails to provide the broadest possible scope of protection for an invention, and makes it difficult to focus on some of the unique aspects of AI inventions in particular. But without an alternative framework for evaluating claim quality, it is difficult to avoid falling into the bad habits of conventional software claims.

At first glance, it may appear that drafting quality AI claims might be even harder than drafting traditional software claims. This is because important parts of the underlying AI algorithms are obscured by the existence of a large number of parameters set during the training process, which may not even be understood by the inventors themselves.

However, this ignores the fact that neural networks are typically characterized by unique, but easily characterized, architectural components. Furthermore, the inventive element often involves unique training techniques to establish the parameters of these architectural elements.

Thus, rather than writing down an algorithm and repeating it three times, F. Chau & Associates drafts independent claims that focus on three distinct aspects of an AI invention: the application, the architecture and the training. In an “Application Claim,” we tell the story of what the invention is used for, while highlighting the inventive concept in the most concise manner possible. In an “Architecture Claim,” we outline the components of the machine learning model. In a “Training Claim,” we specify how the parameters of the machine learning model are established.

The first independent claim is often the most important part of a patent application. It sets out the broadest description of the inventive concept, and is often the first thing one reads when determining the scope of a patent. Thus, at F. Chau & Associates, we believe that the first independent claim should also tell a story, so that a reader can immediately determine what the invention does, and how it does it.

In conventional software patents, the first independent claim frequently focuses on a list of steps, i.e., the algorithm. However, an Algorithm Claim has two problems. First, when reading a list of steps it is easy to lose sight of what is being accomplished. Second, when the inventive concept is split among multiple steps, it is easier for an examiner to find prior art references related to each individual step and ignore important relations among the steps.

Therefore, a quality Application Claim should be written in a way that tells a story. The simplest model for writing a story claim is to use the tried and true “three-act structure.” Put simply, a story claim should have three identifiable parts: a beginning, middle and end. No matter how sophisticated they are, the readers of a patent are human beings, and naturally seek out this kind of structure.

The first “act” of the Application Claim should set the context and establish the inputs for the AI invention. While the preamble may play some role in setting the context, the first act of the Application Claim typically describes the input in a separate limitation.

1. A method of object recognition, the method comprising:
receiving an image and a natural language expression representing an object included in the image;

An example of the First Act of an Application Claim

The second act of the Application Claim describes the function of the AI system, and then includes a concise description of the inventive concept. This part of the claim is akin to the “climax” of the story, and patent practitioners should strive to describe the invention in a concise way that clarifies how the AI system accomplishes its task.

In many cases, the inventive concept involves elements of the architecture or training, so they are often mentioned in the middle act. But while architectural and training elements can be mentioned, they do not dictate the form of the claim.

generating an image mask indicating the location of the object in the image using a machine learning model that includes both a convolutional neural network (CNN) for encoding the image and an recurrent neural network (RNN) for encoding the natural language expression; and

An example of the Second Act of an Application Claim

The third and final act of the Application Claim describes how the output of the AI system is applied to solve a problem. The description of the application must be carefully calibrated to achieve a balance between achieving a concrete effect the user will understand, and maintaining the broadest possible scope of the invention.

creating an annotated image showing the location of the object based on the image mask.

An example of the Third Act of an Application Claim

Drafting the first independent claim as an Application Claim with a three-act structure takes advantage of the natural tendencies of the reader, and makes it easier to process the information described in the claim. By contrast, an Algorithm Claim of a conventional software-based application often lists a series of steps that makes the claim more difficult to understand, risks limiting the scope of the invention (i.e., by including too many limitations), and makes the claim more vulnerable to an obviousness rejection including references directed to each step.

While the “architecture” of code may be incidental for conventional software inventions, it is essential to an AI invention. Thus, while conventional software-related applications often include an apparatus claim, these claims typically include either purely generic elements (i.e., a processor and memory) or components described in purely functional terms.

By contrast, an AI system often includes architectural elements that can be described in a way that is broad, but not purely functional. This is because AI components are categorized into convenient and well-known, high-level categories such as a convolutional neural network (CNN) and a recurrent neural network (RNN). For example, a CNN is designed to operate in a way that makes it ideal for identifying features of an image, whereas an RNN is designed to operate in a way that makes it efficient at processing ordered sequences such as natural language expressions and time series data.

Similarly, one characteristic of many Deep Learning inventions is that they are organized according to an encoder/decoder structure in which an encoder generates a set of features representing the input data and a decoder makes a prediction or classification based on the output of the encoder. In many cases, the novelty of an invention may be found in how the encoder or decoder are structured or arranged.

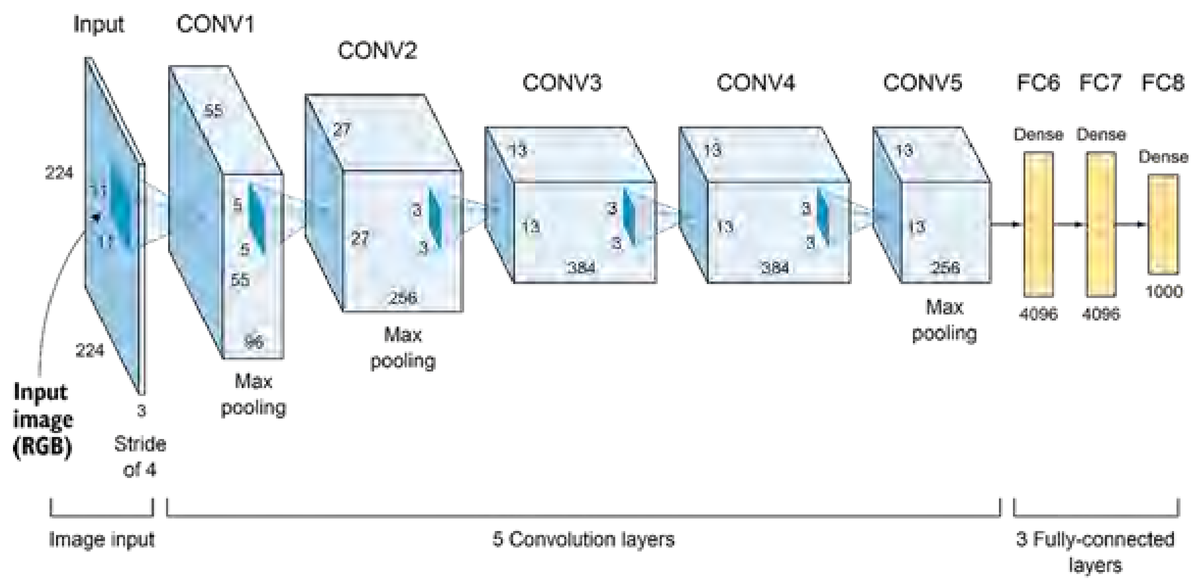
For example, different kinds of neural network layers may be used to encode different kinds of features (e.g., CNN for images, RNN for text or time series data), which may then be combined and fed to a decoder. In some cases, unique architectural elements such as an attention network may be used to combine features from different decoder elements.

Thus, Architecture Claims can often be drafted in a way that is more meaningful in AI inventions than an apparatus claim for a typical software invention. If a practitioner fails to draft a meaningful Architecture Claim with non-generic components, it is a sign that the claims might be of lesser overall quality because the drafter does not understand the specifics of drafting claims in an AI context.

Another sign of a low-quality AI claim is that an apparatus claim is drafted at a level that is too granular. For example, the image below represents a well-known neural network architecture. Obviously, a claim that specifies the number of dimensions in each of the layers would be too narrow. It may not be quite so obvious that specifying each of the layers would also be much too granular, and that some layers should be left out of an independent claim entirely. For example, it may be relevant that the network includes at least one convolutional layer, but the input layer, the max pooling layers and the fully connected layers should usually be left to dependent claims (or the specification).

An example of a Neural Network Architecture

Thus, the challenge in drafting a quality Architecture Claim is finding the “sweet spot” of abstraction that describes architectural elements in a way that provides an inherently meaningful description of the architectural components. Such a description will connect the structure of the component to the function of the component.



Training Claims are the third major category of AI claims. Although not relevant in every case, they are a unique category of method claims that should at least be considered in every AI application. The reason is that AI systems are often produced by first establishing the architecture, and then training parameters of the system to solve a specific problem. In some cases, the primary novelty of an AI claim centers around the training process.

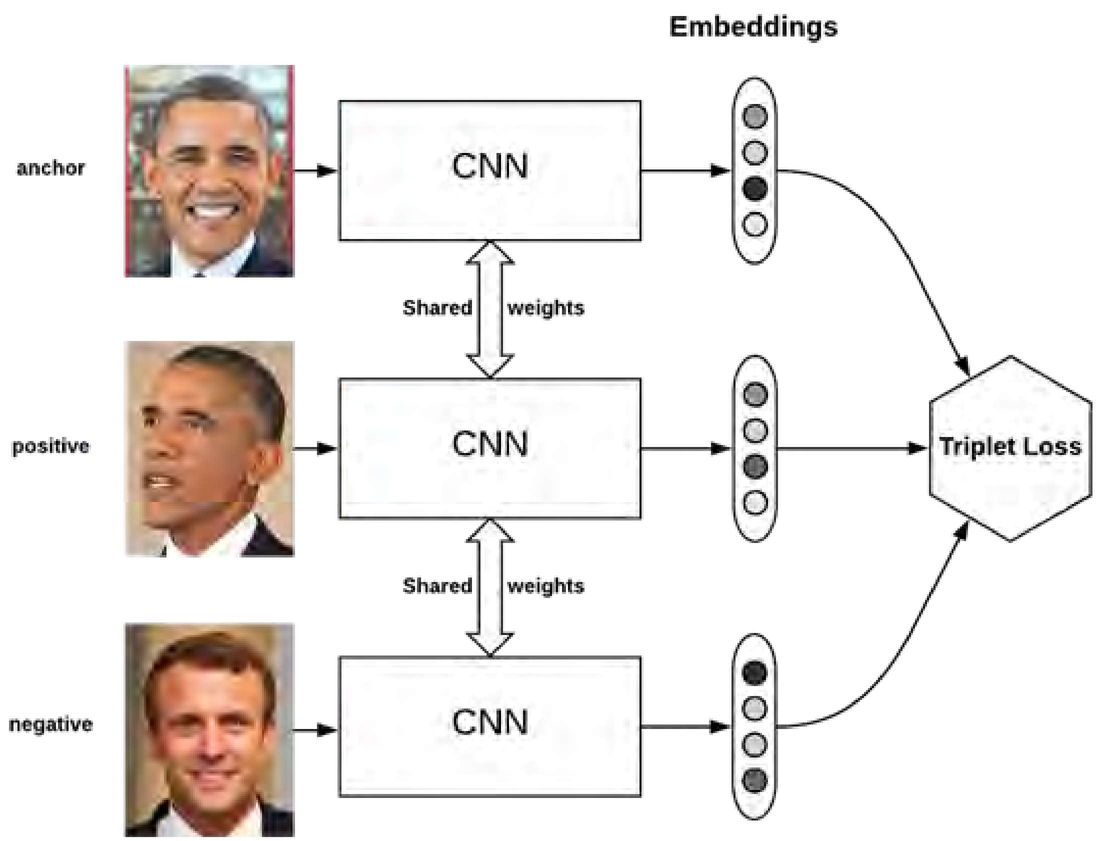
There are three overarching categories of AI training (supervised, unsupervised and reinforcement), and each of these should be treated somewhat differently. However, for now we'll focus on supervised training. Supervised training refers to a method of training a machine learning model with training data that includes the "ground truth" (i.e., the "answers" to the question being posed to the system). For example, training a neural network to recognize human faces might include collecting many images and manually indicating which ones include human faces. Then the trained network can be used to identify whether images outside the training set include a human face.

Although there are different ways to structure a Training Claim, a basic model includes three parts: collecting training data, predicting a result and then updating the parameters of the model using a loss function that compares the result to the ground truth. In many cases, the inventive concept of an AI system can be described in terms of how the loss function operates. For example, the loss function might take into account outputs from intermediate layers of the architecture, or it might be applied differently at different training stages.

As with any claim, consideration must be given to the appropriate level of abstraction when drafting a Training Claim. For example, an inventor may use a particular equation for a loss function, but that specific equation could be replaced by another equation that serves the same purpose. Thus, a patent practitioner unfamiliar with different ways to generalize the role of the loss function may draft a Training Claim that is either too narrow to be useful, or too broad to capture the novel element of the invention. Thus, as with other aspects of AI, it is critical that Training Claims are drafted by practitioners with experience and understanding of how AI training works at various levels of abstraction.

It is common for patent practitioners to either skimp on their descriptions of AI training, or to skip the category entirely. This is unfortunate, since a Training Claim can provide an opportunity to describe an inventive concept from another perspective, which helps broaden the scope of the overall set of claims.

One common question that comes up regarding Training Claims is whether they are detectable, i.e., for purposes of identifying infringers. The answer is yes. However, as in many software cases, detection may require access to the code. And in the case of AI, "the code" includes code used for training, and not just the code used at inference time. Thus, the discovery process for an AI invention may be somewhat different in the details, the detection challenges are largely the same.



Do not be satisfied with the conventional software approach when crafting AI patent claims. By repeating the same algorithm in the context of different statutory classes, patent practitioners can hit their claim counts without putting in the intellectual work of understanding an AI system from the perspective of the application, the architecture and the training. Taking an approach that is more specific to the AI context creates more long-term value.

Without a clear alternative framework for drafting independent claims, it can be tempting to accept a lower quality claim set. Given the time and budgetary constraints of most IP departments, it is important to have simple heuristics for determining whether an application has been drafted to the highest level of quality. The Application/Architecture/Training model for drafting independent claims may not work in every case, but it provides some clear guidelines for quickly identifying a quality AI patent.

Application Claim
1. Collect/describe inputs
2. Use AI to generate output (include description of novel element, such as architecture or training)
3. Apply output to solve problem

Architecture Claim
1. Pre-processing component (sometimes)
2. Encoder
3. Decoder
4. Post-processing component (sometimes)

Training Claim
1. Collect/described training data
2. Use AI to predict output
3. Compute loss function
4. Update the network