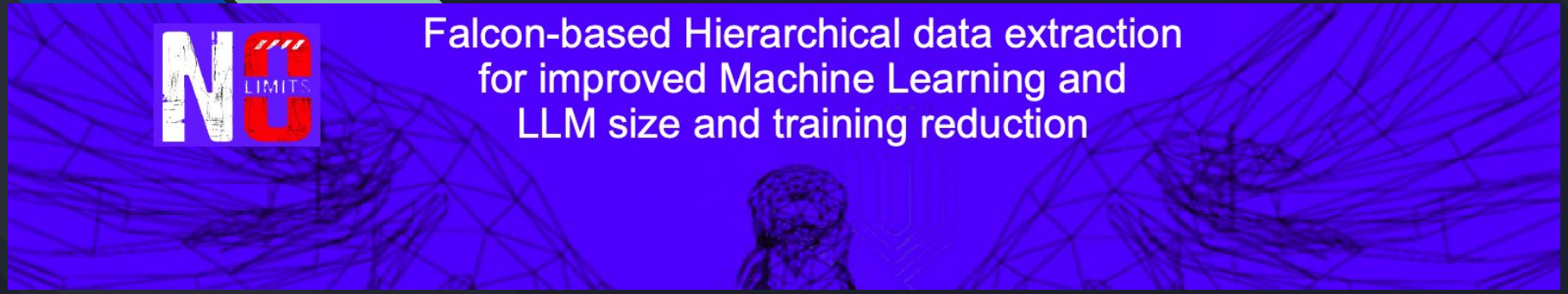


By team NoLimits

Team

Jean-Pierre Bianchi
MS in Machine Learning
ML engineer (Omdena)
Ex founder & CTO
LinkedIn

Ramnath Vaidyanathan
Head of Data Science
and Engineering
Theta
LinkedIn

Giriesh Jaiswal
Lean ML engineer, spec in LLM
Morgan Stanley
LinkedIn

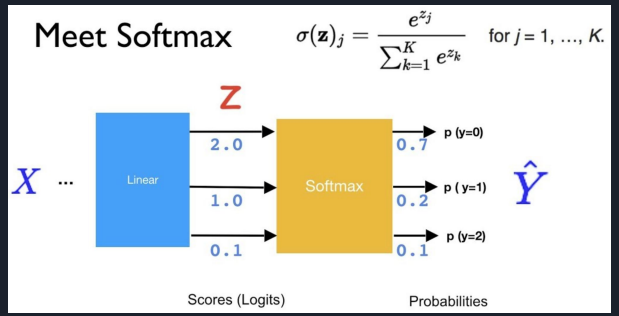
Information is hierarchical

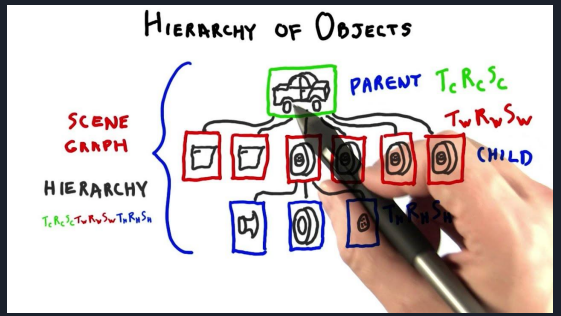
Human language describes reality, and reality is made of objects, which are made of objects. Atoms make cells, which make organs, which make bodies, which make families & groups etc

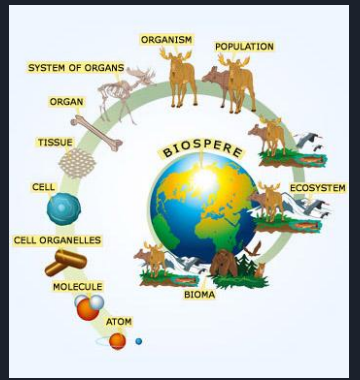
Why don't we already have all human knowledge represented in a graph with every object in a node, with edges to represent their relationships?

Because the concept of object, as clearly defined by dictionaries, that everyone uses in their language in every sentence is not used in Machine Learning and Data Science, which instead use statistical concepts such as Probably Approximately Correct, ie 'it's probably true, but it could be totally wrong'. It's not only LLM's that hallucinate.

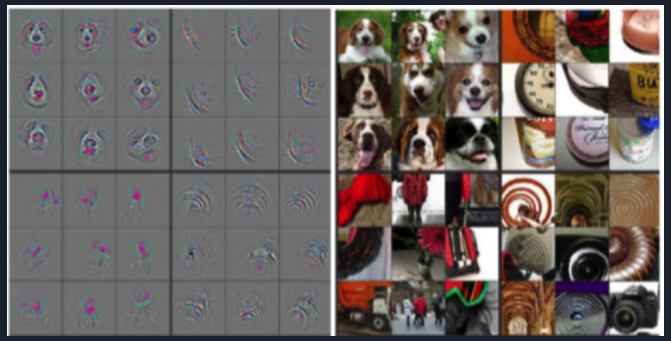
This project aims to help with that problem.







Features, features…



Machine Learning uses the word ‘features’ over and over, but not in

the way human use it. And that is the problem.

In ML, features are basically any data that contributes to the representation of an object. In the image, we see that some features look like dogs, ‘probably approximately correctly’. When one keeps mixing, adding such fuzzy ‘objects’, it becomes very hard for a model to always be right.

Humans recognize a cat with certainty using features such as ‘eyes’, ‘ears’, ‘whiskers’, ‘fur’. Every human knows such features from a young age. This is what allows us to say ‘this IS a cat’. If we can isolate such features, we can go from the world of endless approximations, where things ‘look like’ to the world where things ‘ARE’.

This has powerful consequences.

Solution

CNN, Language Models seem to be 'hierarchical' because there are layers, and features appear at every layer. This is a 'top-down' view, ie a description of what researchers 'see' and word it as 'feature'. It is not possible to describe reality with a top-down approach, which is basically a gigantic 'mapping' over trillions of data points, and we will always need more. This is exactly where we are! It will never end because we are fighting infinite complexity when we look at pixels instead of the underlying reality. E.g. there are trillions of trillions of ways to draw a cat, but there is only a handful of features that can describe them all with certainty.

The other approach is 'bottom-up', ie like physicians do to find out how 'things assemble' in the real world, into more and more complex structures, from which REAL features emerge at every new layer (unlike in a neural net).

We all know these features, we use them every day. Humans can speak a language with only a few thousand words, ie a few kilobytes! And a few rules of grammar. The KEY is that our visual cortex is able to recognize those features instantly. So that could be the key towards better algorithms.

Fortunately LLM's have reached such a level of complexity that intelligence has truly emerged, and we are able to query them for the features that humans use to define an object.

As you will see, we have found a way to extract real features from LLM's billions of parameters. Now, we can bring real features into the world of Machine Learning and AI, with potentially dramatic improvements.

Possible application

We can illustrate our approach with the following comparison.

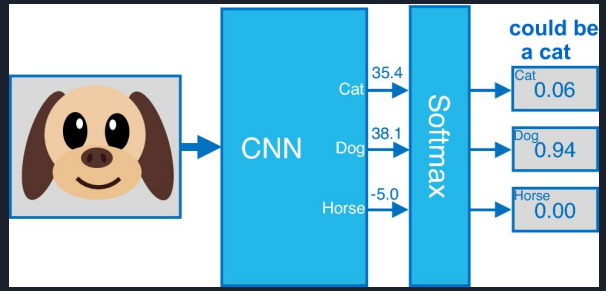
The first image illustrates a typical statistical approach, which turns every label into a probability and the highest one is picked. However, in a CNN, everything is deterministic until softmax is applied, and all of a sudden, a dog can be classified as a cat.

So, why not 'redo' it and remove the last statistical layer, and replace it with logic based on the features given by a LLM, which makes it deterministic all the way.

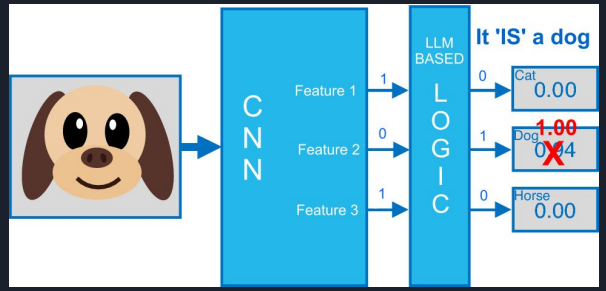
It looks like DETR but it's not because the CNN is 'guided' to find the features that our hierarchical database tells it to find. In the second image, we see that the CNN produces a list of 1's and 0's for every feature requested (using segmentation for instance), and the logic block deterministically decides what the object is, without doubts. This dog has zero probability to be classified as a cat.

The distinction between the two is important because we switched from probabilities to certainty. Moreover, this algorithm can provide full explainability, which is understandable by humans, e.g. "we identified this image as a dog because there is a tail, 2 long ears, 2 eyes, a long mouth, 4 legs with pads etc etc".

This is because, we said earlier, we are not using typical ML features which are not understandable by humans anymore, but features we all know and understand immediately.



We go from probabilistic guessing to certainty



Advantages & Improvements

Using real life features has many advantages:

- Complexity vs compression: the notion of object in itself is a powerful compression technique, ie complex objects are represented by one word and simply linked to their parts in a graph.
- Efficiency: once the knowledge graph is built, it represents the knowledge of humanity, which evolves but doesn't change drastically.
  - Therefore, it is not necessary to re-learn it all at every training, which translates into much shorter training times for LLM's *
  - Because of the compression advantage, this should also translate into smaller LLM's with new architectures incorporating the knowledge graph
  - Inference is much faster (human-like)
- Even greater accuracy: Falcon 7b has also shown the ability to not only provide features, but also properties that a CNN could use to better locate / recognize them (relative position, size, shape, coloration etc).
- Explainability: being able to classify objects like humans also allows to explain which features were decisive in the process
- Because the features are those humans use, the graph can be inspected and edited by humans (unlike the data features hidden in gigantic matrices)
  - For instance, humans could once and for all determine which features are more important for every object.

*Providing a knowledge-base memory to LLM's is a very exciting prospect and could be a game changer!!

```
Features:
        Eyes
        Ears
        Tail
        Fur
        Teeth
        Nose
Eyes:
        Coloration
        Shape
        Placement
        Expression
        Size
        Position
Ears:
        Shape
        Placement
        Expression
Tail:
        Coloration
        Shape
        Length
Fur:
        Coloration
        Texture
        Placement
        Expression
Teeth:
        Type
        Number
Nose:
        Position
        Coloration
```

Markets & revenue stream

A hierarchical knowledge based approach has many advantages, which can greatly impact several markets:

- CV: full explainability, better accuracy, less/no errors, no hallucinations
- Robotics, AV, guided systems improve when one can localize & recognize objects
- Healthcare: better diagnostics, and therefore better treatments
- ML: algorithms improvement by taking into account real features, dimensionality reduction
- LLM: training time and size reductions
- Other patentable applications (not disclosed here)

It is extremely difficult to evaluate the potential revenues from so many big markets at this stage. This would require a full blown business plan. But one can already see how providing true explainability and improved accuracy can be a game changer! Revenues are certainly in the billions to whoever takes the lead.

Next steps / backlog

This project was a ‘proof of concept’, not a commercializable app, simply because there are so many ways to use it to improve ML & AI algorithms, even LLM’s, which in turn will impact many important markets.

We used a Falcon 7b Instruct model and it already gave outstanding results, but there were errors at times (e.g. ‘gills’ being a feature of a cat). This is also due to the statistical nature of LLM’s, but we can greatly improve the results by using 1) a bigger model, 2) more defined prompts and 3) double/triple checking the result with separate means (other LLM’s which won’t hallucinate in the same exact way, verifications from the internet, and human inspection).

With funding, we would:

- Improve the current solution to create a deeper and totally reliable hierarchical knowledge base
  - Requires more programmers, access to expensive hardware
  - Improve all aspects, test, extensive and reliable CI/CD
  - Then turn it into a professional product, tested, deployed, and scalable
- Identify the first ‘easy’ applications in terms of impact on potential markets
  - object recognition in general
  - Then specializing it for robotics, AV, diagnostics etc
- Find key partners to try our technology
  - Requires staff with industry knowledge and connections
  - Requires marketing experts
- Become leaders in such technology
  - Requires hiring experts in targeted fields and showcasing our potentials by publishing articles and papers
  - Put a solid management structure in place

Working solution

A proof of concept has been deployed at falcon-hierarchical.streamlit.app

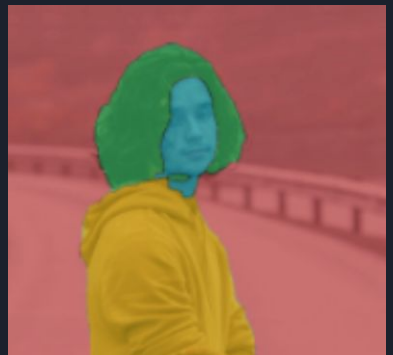
The backend runs on the GPU cloud Modal.com.

The code is at https://github.com/jpbianchi/nolimits (private repo)

We prompted Falcon 7b Instruct to generate the following graphs made of real features.

See our video for more details.

With more time, we would have demonstrated how this knowledge graph can be used in conjunction with image segmentation to properly classify objects without errors by using the features corresponding to the segments (in color below).



Team NoLimits

Keys retrieved and tested!

Prompt is created

Looking for features of: cat, elephant

Inference runs on a A100 GPU and will take approx 2 minutes for Falcon 7B Instruct

GPU inference has started!

Features are found