How Google "Translates" Pictures into Words

Chang Liu, *Student at University of California, Los Angeles*

Abstract

This paper concerns a mathematical approach to translating images taken by engineers at Google. The basic concept was to take their method of translating words, which already used vector space mathematics and apply it to images to give appropriate captions for them. The method is coined as Neural Image Captioning (NIC) and involves the use of Artificial Neural Networks, specifically Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN).

1. Introduction

The article I chose focused on a group of Google engineers who are developing a machine learning algorithm to write captions for pictures. The algorithm they used involved vector space mathematics that was nearly identical to the method used for translating words. To get a better grasp on this topic then, I backtracked and researched Google Translate and the process of translating words, rather than images.

The readings were very applicable to me personally, as I am pursuing a Linguistics and Computer Science degree. Previously, I had thought that the only career pathways that would utilize both Linguistics and Computer Science would be to work for Google Translate and the like. I wasn't wrong per se, but my idea of what Google Translate did was outdated.

1.1. Google Translate

I found that Google Translate originally conducted translations by comparing millions of texts in both the source language and the target language. It used what was known as Statistical Machine Translation to find patterns among these large amounts of text to get an idea of how the sentence structure works and where a word should occur. It assumes that words and phrases with the same statistical properties across languages are equivalent and uses this data to translate. Finally, Google Translate would reorder the words as necessary to have an appropriate syntax.

This method matched the idea in my head but I found that Google has implemented a better and more versatile way to translate. The conventional method requires bilingual text at the very least, which takes significant manual effort and time to compile.

1.3. Google Translate v2

Version two assumes that all languages must be able to describe a similar set of ideas, like numbers or common animals. The engineers then decided to construct an abstract vector space out of these words, each word being a vector that points to others word and consequently links to those words mathematically. With that, vector operations like "king - man + woman = queen" are acceptable and work. To me, this was an insanely creative way to approach it and it could even revolutionize the study of linguistics in my opinion. It's even crazier that the engineers called this approach "simple," but then again, it's the Google engineers.

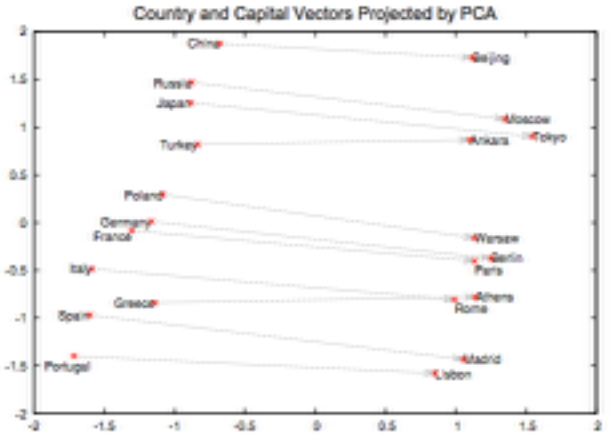

Figure 1. A program was trained on a massive amount of data without knowing the relationship between countries and capital cities.

1.3. Capital Cities Experiment

A better example to help understand this vector mathematics would be the capital cities experiment the engineers conducted. By "training" the program on massive amounts text, it places similar vectors (countries in this case) like China, Russia, Japan together and their respective capitals together in another area. Without even knowing the concept of capital cities, the

algorithm understands the relation between a country and its capital mathematically; it shows that the vector distance between China and Beijing for example is very close to the distance between Spain and Madrid. Like before, we can then perform an operation like “China - Beijing + Spain” which would yield Madrid.

1.4. Generating Distributed Representations

To actually generate the vector representation of a set of words, the algorithm must first construct a distributed representation. In this space, each word in the dataset would correspond to a specific point and each dimension to a grammatical or semantic feature. Words with similar functions that appear in the same contexts then would be grouped near each other. To determine how close words are grouped to one another, two methods are used: Continuous Bag of Words (CBOW) and Skip Gram. CBOW is the more popular method and it involves creating a vector that combines features from a large set of data and plots the points onto a vector space. For example, the phrases “Hi Fred how was the pizza” and "Hi Fred, Pizza offer: two for one" would yield a vector of { hi, Fred, how, was, the, pizza, offer, two, for, one }. The first statement then would be plotted as [1,1,1,1,1,1,0,0,0,0] because we count the occurrences of the words; the second would be [1,1,0,0,0,1,1,1,1,1]. In the case of plotting countries as shown above, the vector would count certain semantic features instead of word occurrences. The points, which lie on many dimensions, would then undergo a process called Principal Component Analysis, which reduces the dimensions and converts them into a 2D space for us to easily visualize.

1.5. Modifications to Allow Picture Translation

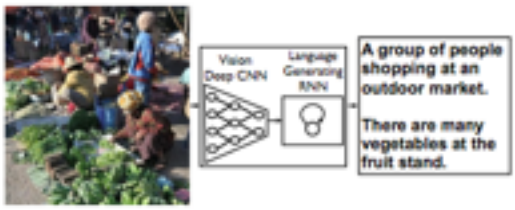

Figure 2. Instead of using RNNs for encoding as well as decoding, the engineers used a CNN for encoding the images and RNN to decode the vectors generated by the CNN.

Words and sentences in one language would be encoded by an artificial learning algorithm called Recurrent Neural Network into a vector representation. A complementing decoding RNN would project that vector representation onto the words of another language and output the vectors that are the closest match to “translate.” All the engineers had to do then to translate images was to switch out the encoding RNN for an encoding Convolutional Neural Network, which is a learning algorithm used for images and videos.

The decoding RNN would do the same job and effectively “translate” an image this time. RNN’s and CNN’s are both types of Artificial Neural Networks and these algorithms are something I would be very interested in doing further research on. These are the algorithms used for handling data that could potentially have large amounts of inputs that may also vary greatly. An example would be handwriting and the many different visual representations of a particular word; I’ve also wondered how computers could parse comprehensible words from handwriting and now I have a small grasp on how.

1.6. Test Results

Of course, I did not expect it to work perfectly. Using the BLEU scale, a standard metric for evaluating a machine translation, it turns out human translation still tops everything at a score of 69. However, it is very impressive that this Neural Image Caption algorithm, as the Google engineers call it, has surpassed the score of state-of-the-art technology by more than double at a BLEU score of 59, compared to the latter’s 25.

Figure 3. Captions generated for some PASCAL images sorted by accuracy.

1.7. Concluding Thoughts

I also liked how the source code for computing vector representations of words is open source. I can see some interesting projects in the near future that utilize this algorithm. As mentioned in the article, this would greatly help the visually impaired in navigating the internet and it would also be interesting to be able to search for an image by caption in the future.

References

[1] ArXiv, Emerging Technology From the. "How Google Converted Language Translation Into a Problem of Vector Space Mathematics." MIT Technology Review. MIT, 25 Sept. 2013. Web. 04 Mar. 2015.

[2] ArXiv, Emerging Technology From the. "How Google "Translates" Pictures into Words Using Vector Space Mathematics." MIT Technology Review. MIT, 21 Dec. 2014. Web. 04 Mar. 2015.

[3] Mikolov, Tomas, Quoc V. Le, and Ilya Sutskever. "Exploiting Similarities among Languages for Machine Translation." (n.d.): n. pag. 17 Sept. 2013. Web. 4 Mar. 2015.

[4] Vinyals, Oriol, Alexander Toshev, Samy Bengio, and Dumitru Erhan. "Show and Tell: A Neural Image Caption Generator." (n.d.): n. pag. 17 Nov. 2013. Web. 4 Mar. 2015.

[5] "Neural Net Language Models." - Scholarpedia. N.p., n.d. Web. 04 Mar. 2015.