From Text to Information: Using Natural Language Processing Techniques to Glean Meaning

Gautam Mittal[1], Robert Cheung[2]
Gunn High School[1], The Fin Exploration Company[2]

BACKGROUND

Since the rise of commercially available computers, the ability to access the Internet and other computing resources has empowered educators and students alike. Educational platforms and applications have enriched the quality and delivery of education, especially for the younger generations of students. Early on in a student's formal education, the first challenge they must overcome is learning to read. Part of this challenge includes conceptualizing the meaning of written words and attaching a visual association with text. However, breakthroughs in modern computing can help students overcome these challenges. Machine learning, one of the most quickly evolving fields of computing, can aid in solving the challenges that students face when learning to read. More specifically, natural language processing (NLP), or the use of algorithms to glean semantic meaning from human-generated language, has the ability to help young students conceptualize the text they are learning to read. In Peter Norvig and Stuart Russell's Artificial Intelligence: A Modern Approach, they characterize natural language to be ambiguous, which is part of the reason why it is so difficult for both humans and machines to perceive a singular meaning from a piece of text. NLP enables computers to classify and evaluate text through the use of mathematical probabilistic models that determine the significance of words. Once a machine has a basic understanding of the meaning of the text, other techniques such as visual processing can be used to help the computer paint a graphical interpretation of the text. In the past, researchers have used NLP models to build search engines, automatically classify spam or non-spam email messages, and on the education front, automatically generate practice exercises and produce concise summaries. The study would build upon the widely unexplored educational applications of NLP, as well as use the fundamental algorithms that allow machines to efficiently classify and extract semantic meaning from text to generate a visual interpretation that can be used to allow young readers to conceptualize reading material through the help of these computer-generated animated scenes.

PROBLEM

Research Question: How can natural language processing techniques be utilized to construct educational storybook scenes from plaintext to allow those learning to read to better conceptualize their literature?

Issue: Extracting concrete information from an abstract piece of text to generate illustrations or images that act as visual aids for the reader.

SOLUTION

There are two primary possible solutions regarding the translation of plaintext to images. The first being reverse image captioning, which uses convolutional neural networks, a machine learning algorithm loosely modeled off of the human brain's neural interactions, to generate images based on a training set of text captions. The second possible solution is to first extract semantic information from text using natural language processing techniques, and then using that information generate an image that arranges objects in context with other nouns and verbs of the text. The first solution focuses more on generating an image while the second solution focuses more on extracting information from abstract text using natural language processing.

Solution: Develop software that favors an information extraction-based approach over generating images. This solution was much simpler and more straightforward to implement, and provides a basis for solving a lot of other challenges using information extraction other than generating images as reading aids. In addition, this solution does not require a large "training set", or set of examples for the computer to learn from, unlike the convolutional neural network approach.

Fig. 1. An example of a small subset of a large training set necessary for reverse image captioning.

IMPLEMENTATION

1. Pre-Processing Text

- Remove all contractions and other words that may "confuse" the chunker.
- Contrary to popular methods, the pre-processor keeps all stop words (the, it, that, etc.).

2. Summarizing Text Using TF-IDF

- To summarize text, the program uses term frequency-inverse document frequency (tf-idf), a method of numerically reflecting the importance of a word within a sentence. This method is commonly used for unsupervised summarization, as it does not require any "training examples".
- Each word is ranked (weighted) by finding the number of occurrences of that word in a particular sentence and comparing that to the number of occurrences of the word in the entire document.
- One can take the average score of each word's tf-idf score in a sentence, and subsequently create an "importance ranking" for that sentence.
- For simplicity, the top three most important sentences are used when generating a summary.
- Summarization allows the sentence bigram chunker to use more concise content to extract information about the "the important points" of the text.

"Once upon a time, there was a little girl named Goldilocks. She went for a walk in the forest. Pretty soon, she came upon a house. She knocked and, when no one answered, she walked right in. At the table in the kitchen, there were three bowls of porridge. Goldilocks was hungry. She tasted the porridge from the first bowl. "This porridge is too hot!" she exclaimed. So, she tasted the porridge from the second bowl. "This porridge is too cold," she said So, she tasted the last bowl of porridge. "Ahhh, this porridge is just right," she said … Goldilocks woke up and saw the three bears. She screamed, "Help!" And she jumped up and ran out of the room. Goldilocks ran down the stairs, opened the door, and ran away into the forest. And she never returned to the home of the three bears…"

`tf-idf` →

She whined. Goldilocks fell asleep. She screamed, "Help!"

3. Training and Building a Bigram Chunker

- A chunker is a simple natural language processing tool that uses given grammars, or part-of-speech patterns, to classify certain parts of a sentence.
- In this case, a specific grammar was not very useful, as the input data could be very diverse.
- Instead, the chunker was "trained" by reading many labeled examples of sentences with sections classified as noun, verb, or prepositional phrases. This allowed the chunker to formulate its own grammar for pattern-matching by looking at examples from Wall Street Journal articles.

We	saw	the	yellow	dog
PRP	VBD	DT	JJ	NN
NP		NP		

4. Extracting Information From Chunker Data

- Using the noun, verb, and prepositional phrases returned by the chunker, enough data about the sentence is available to start making assumptions about various entities within the sentence.
- With the knowledge about common trends about verb, prepositional, and noun phrase position within a sentence, and algorithm identifying the subject, verb, and object can be formulated.

5. Generating Images

- Using the subject, verb, and object data, separate images for each entity are downloaded from the Internet.
- The images are then strung together in the order in which they appear in the original sentence when generating an animated image.
- These images can then be used as reading aids to further enforce conceptual and visual aspects of the text.

EVALUATION

- Solution works well for basic subject, verb, object information extraction.
- Fast — does not require a lot of computational power, mostly using a lot of basic math in some interesting ways.
- Is accurate most of the time -- there are exceptions where the solution fails due to lack of knowledge regarding certain words.
- Extracting "context" from words is **very** difficult, even with more complicated techniques (neural networks, training examples, etc.).

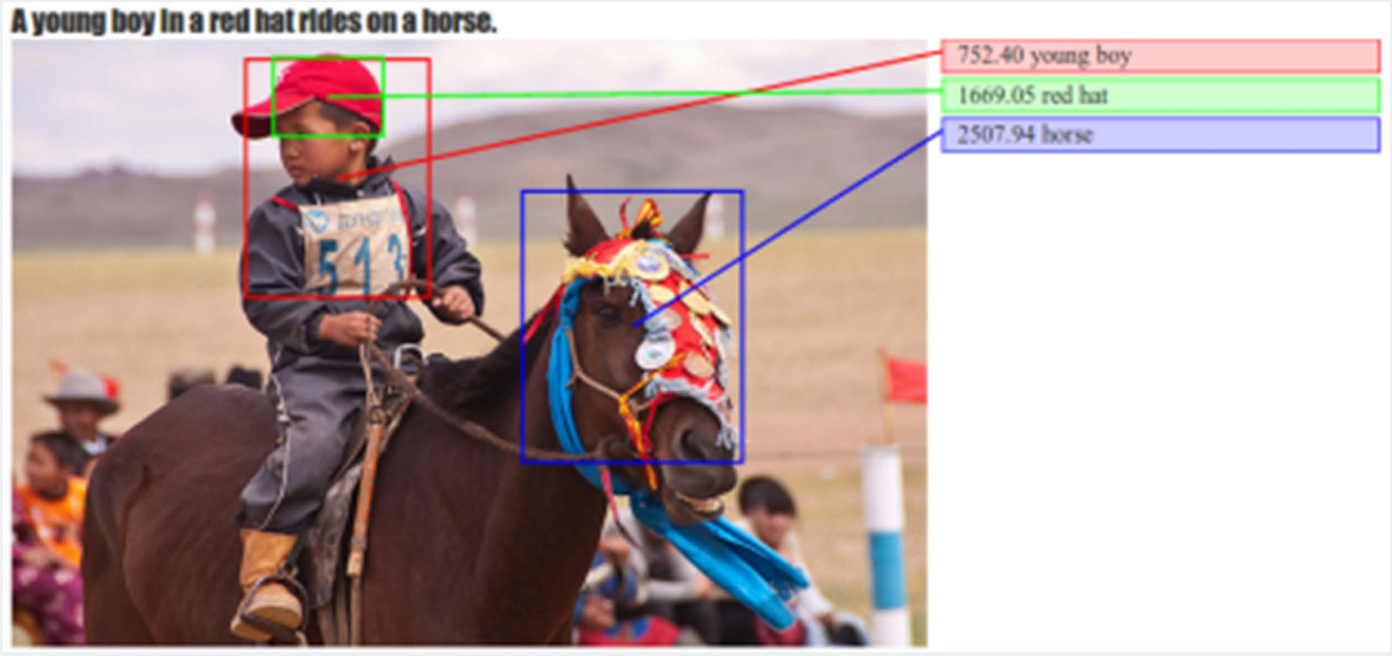

Fig. 2. Using hundreds of thousands of training samples, researchers at Stanford University are able to generate dense captions for images. Doing the reverse in a contextual manner is significantly more difficult.

ACKNOWLEDGEMENTS / REFERENCES

I would like to thank Mr. Robert Cheung of The Fin Exploration Company for mentoring and providing insight throughout the course of this project, the NLTK Project for providing natural language processing libraries and tools, and Amazon Web Services for hosting the code.

Bird, Steven, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. Beijing: O'Reilly, 2009. Print.

Introduction to Summarization By Daniel Jurafsky. Perf. Daniel Jurafsky. Coursera. Stanford University, n.d. Web.

Johnson, Justin, Andrej Karpathy, and Li Fei Fei. "DenseCap: Fully Convolutional Localization Networks for Dense Captioning." Diss. Stanford U, n.d. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016): n. pag. Print.

Norvig, Peter. "Natural Language Processing." *Artificial Intelligence: A Modern Approach*. By Stuart Russell. 3rd ed. N.p.: Pearson, 2009. N. pag. Print.

Supervised Relation Extraction By Daniel Jurafsky. Perf. Daniel Jurafsky. Coursera. Stanford University, n.d. Web.

TF-IDF Weighting By Christopher Manning. Perf. Christopher Manning. Coursera. Stanford University, n.d. Web.