A Reverse Mad Libs Game

Natural Language Processing

Scott Bing

BSSD 6000 System Software Design

Jonathan Daniel Lee

September 13, 2020

Contents

Abstract

Typically a game of Mad Libs involves the system asking the user to fill in a list of arbitrary words that represent a different part of speech, nouns, verbs adjectives, adverbs, prepositions, etc. The system substitutes these words using the appropriate parts of speech designation into a story. The story is then shown to the user. Everyone has a laugh when the completed story is read out loud.

What if you gave the game to a computer and asked it to not only substitute the correct parts of speech for each missing word but also to come up with a suitable word that matched the context of the story. So when the final story was read, for the most part it would make sense. This represents the theme of this project. Given an arbitrary story with a series of missing words, the Reverse Mad Libs application uses the principles of Natural Language Processing to fill in each missing word with the 'best' candidate word.

Introduction

Do you commonly use that autocomplete or grammar checking facilities on your favorite word processor? Have you ever used a dynamic language translator? Have you ever used a chatbot on a web page? This is a picture of a human, typically clipart, that pops up and asks you if you would like to chat about your issue. I don’t need to ask whether you have ever pleaded desperately “representative” over the phone when a machine has answered. I believe just about the entire population has dealt with this situation. If you have had any of these experiences, you have been exposed to some form of Natural Language Processing.

Natural Language Processing (NLP) is a branch of Artificial Intelligence that allows electronic devices to interpret human languages. It combines the fields of Linguistics, Computer Science, and Mathematics. NLP allows these devices to communicate with people, using a human language. NLP can read and interpret text and hear and interpret human speech. The discipline has been around for nearly 50 years. We are surrounded by language. We use it to speak, read talk, and even think. We see words about us every day in signs, Signs, email, text, books, web pages, etc.

Normally a computer processes structured data that it can easily read and parse. Human language is unstructured data. It is ambiguous and chaotic. How does a machine deal with homonyms – too and two, bank the financial institutions and bank the river, rose in a garden and stock prices suddenly rose. This is just one of the issues that NLP must address accurately to be of any value.

What is Natural Language Processing

Natural Langage Processing (NLP) is an attempt to meld human language with computer language. From this point on in this discussion for the sake of brevity, NLP will refer to Natural Language Processing. When you use the auto-completed facility on a phone or computer application, you are using NLP. When you converse with a non-human voice on the phone, you are using NLP. It is not a new concept. It was thought about long before the inception of the modern-day computer. NLP attempts to take noisy unstructured textual data and combine it with the contextual complexities of human language into structured vectorized data that a machine learning application will understand.[1]

Since the early days of the electronic computer, computer scientists have tossed around the idea of a computer understanding and communicating with the human being using their local language. Computers speak in terms of '0's and '1's. A leap had to be made to translate the complexity of the human language to binary ones and zeroes and the reverse. Alan Turing wrote a paper in the 1950s entitled, *Computing Machinery and Intelligence,* where he describes such a computer model.

This paper is truly the cornerstone of AI (Artificial Intelligence, because it contains not only an operational definition of intelligence for a machine (being able to imitate humans), but it also suggests how to design a machine to imitate human reasoning, in a section titled "Learning Machines". The idea of Turing is to build a machine which simulates not a human adult mind, but a child mind, which he conceives as "rather little mechanism, and lot of blanks". This machine should learn from scratch and by examples, not by hardcoded general rules, by means of some built-in general learning model.2

[1] **Dipanjan Sarkar; "Understanding Feature Engineering: Deep Learning Methods for Text Data"; 2018**
https://www.kdnuggets.com/2018/03/understanding-feature-engineering-deep-learning-methods-text-data.html
[2] **Paolo Caressa, Artificial Intelligence: "the new electricity"; 2020**
https://www.codemotion.com/magazine/dev-hub/machine-learning-dev/artificial-intelligence-the-new-electricity/

Knowledge-based machine learning applications came into existence during the 1980s and 1990s. A Knowledge-based system consulted a database full of phases and contexts to simulate human language. Modern-day NLP systems use pattern matching, neural networks and mathematics to learn and utilize human language.

Natural Language Processing is just a branch of Artificial Intelligence that deals with computational linguistics. The goal of NLP is to develop a system that can respond realistically to human communication. The electronic device can easily respond to a human making a request using everyday speech fraught with all of the complexities of local phrases and idioms. The machine is trained by listening to millions of human conversations. It no longer has to parse keywords to put together context, it can glean the context from the adjacency of words in a normal human conversation. The machine still parses words into tokens, but it relates those words to the context of the dialog through a series of neural networks. The original NLP systems store context into a database referred to as a *Knowledge Base*. The Knowledge Base will be discussed in the next section.

Rules-based vs. Mathematical Based Learning

Natuaral Language Processing uses the principles of machine learning to synthesis human language. There are two design pardigms for machine language. There is a Rules-based and Mathematical based machine language systems.The early forms of NLP were primarily rules-based. The computer would consult a ‘Knowledge Base’ which was a database of grammar. As the machine learned, the database had to be continually updated with new rules. While there were some significant advances in NLP achieved by rules-base machines, they were not that efficient and they did not mimic the way that humans process language. There were some rules-bases systems that were capable of rules self-regeneration.

Infants learn language by recognizing and imitating observed patterns. What if machines could do the same thing. This is where mathematical based learning had its beginnings. With a mathematical base learning, machines are trained to recognize patterns. And just as humans learn and think, patterns are evaluated and either kept for future reference or eliminated. This continual process of evaluation and dismissal is referred to a machine training. It is base upon statistical analysis. The latest NLP systems use neural network models to facilitate language learning. The ideal NLP system would combine all of these systems. As the machine acquires knowledge, that knowledge could be stored in a database for quick retrieval later, rather than repeating the learning process. The knowledge base would no longer need to be externally updated

Natural Language Processing, Terminology

There is some terminology in the field of computational language processing that needs clarification. What is the difference between Natural Language Processing, Natural Language Understanding, and Natural Language Generation? These three concepts all relate the computational analysis of human language. They differ in how they interact with human language.

National Language Processing (NLP) refers to how a machine recognizes human language. The keyword here is *read*. How does it take in the phrases and words? How does it process speech?

Natural Language Understanding(NLU) refers to how a machine picks up and assimilates the context of human speech. This is typically considered a part of NLP, however, some professionals consider this a separate category. It breaks down the language into parts of speech, sentences, and morphemes.

Natural Language Generation(NLG) refers to how a machine responds to human speech. The keyword here is *write*. How does it formulate e response to an NLP query? As its name suggests, how does it generate language.

Bidirectional Encoder Representations from Transformers BERT

BERT (Bidirectional Encoder Representations from Transformers) is an NLP package supplied by a Google. What makes BERT so special? Why was it chosen for this project? Traditional NLP processors process text in a lateral direction as most human speakers. This is left to right or right to left. BERT combines the lateral directions to achieve a bidirectional scanning procedure. This accounts for contextual meaning from both directions. Google engineers felt that would give a deeper analysis of the language. It may find aspects of the language not uncovered by a simple unidirectional scan. BERT does not rely on any directional scanning, it consumes the entire section of text in one access. The scanning process does not have a directional bias it evaluates the context of words based on their complete contextual surroundings. This makes BERT is ideal for classification problems, questions, and answering applications, and parts of speech recognition.

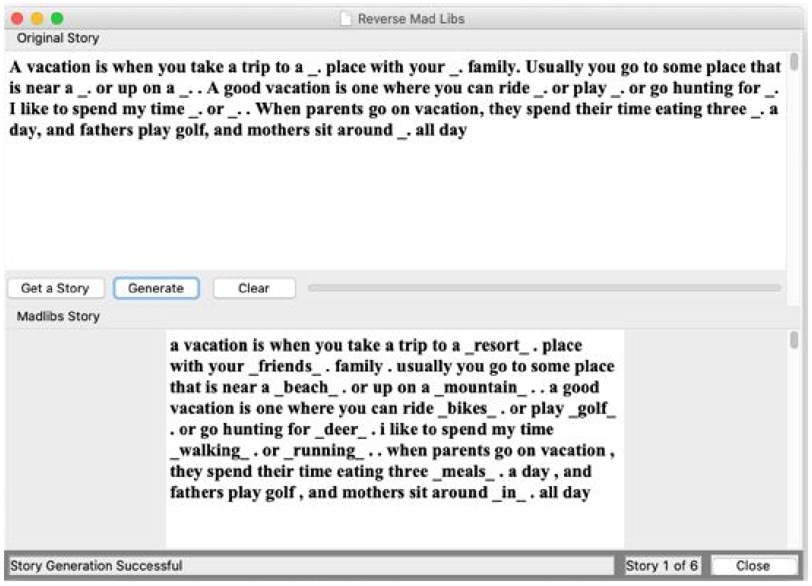
BERT is an open-source package provided by Google. The source is located at https://github.com/google-research/bert which also includes a thorough descriptive README file. BERT is also unique in the fact that it not only processes word tokens, it also processes complete sentences as a whole. First, It converts text to lowercase, if BERT's lowercase model is active. Next, it tokenizes the text into words. Words with prefixes and suffixes are broken down further into separate tokens. For example, the word calling would become 'cal; '##ing' and the word preview would become 'pre##', 'view'. The resulting words are indexed according to an internal BERT vocab file. Special BERT tags are inserted at the beginning of the text and the end of each sentence. The final tokens are converted to vectors that associate the current context with the word. These vectors are also referred to as tensors. A tensor is a mathematical object, like a scalar, a field, a group, a vector, etc.

Bert uses these vectors to predict the next or missing sentence, the next word or missing word. All of this takes place inside a BERT specific data structure known as a transformer. A transformer is a Google-developed facility used to learn the context between words and sentences. The transformer is composed of encoders and decoders. The encoder reads the text and breaks it down into tokens as described in the previous discussion. The decoder makes predictions about the contextual relationship of those words.

What is Reverse Madlibs?

Typically a mad libs game is played as follows. Player One is asked to supply a series of words broken down as parts of speech. The Player One chooses arbitrary words. Player Two substitutes these into a random story. Player Two reads the resulting story. The idea is that the story and the words have no connection with each other. In a reverse mad libs game, the machine sees the context of the story and uses NLP routines to substitute the best words into the story. My project uses BERT as the NLP mechanism to substitute the words. Bert is an NLP tool kit that provides functions that analyze language structures and simulate human conversation.

The application is fed a series of short stories where random words are missing. The missing words are indicated with a '_.' Symbol. BERT is used to make predictions for the missing words. The application's main screen is shown in Figure 1.



A vacation is when you take a trip to a _. place with your _. family. Usually you go to some place that is near a _. or up on a _. . A good vacation is one where you can ride _. or play _. or go hunting for _. I like to spend my time _. or _. . When parents go on vacation, they spend their time eating three _. a day, and fathers play golf, and mothers sit around _. all day

a vacation is when you take a trip to a _resort_ . place with your _friends_ . family . usually you go to some place that is near a _beach_ . or up on a _mountain_ . . a good vacation is one where you can ride _bikes_ . or play _golf_ . or go hunting for _deer_ . i like to spend my time _walking_ . or _running_ . . when parents go on vacation , they spend their time eating three _meals_ . a day , and fathers play golf , and mothers sit around _in_ . all day

Story Generation Successful

Story 1 of 6

Fig. 1Reverse MadLibs Application Screen

To operate the application, click the “Select a Story” button. The next story will be displayed in the upper frame. The number of the current story appears in the status bar at the bottom of the screen. Press the “Generate” button to process the story using BERT’s predictive algorithms. When the generation process is complete, the generated story will appear in the bottom frame and a confirmation message appears in the status bar at the bottom of the screen. The entire screen can be cleared off of its contents using the “Clear” button.

Project Resources

The application is written using the Python language. Python was chosen for its syntactical brevity, its eas of use, and its machine learning affinity. Python does not carry a built-in graphics capability. There is a built-in package called Tkinter. It creates simple graphic user interface screens and screen widgets, but the graphics are very limited. The Reverse Madlibs application was written using Tkinter. Many other Python packages provide graphical routines. Since this application primarily deals with text, Tkinter is sufficient. Here is a list of some of the additional resources used to complete this project

- Python – the primary implementation programming language. Python is terse, easy to use, and lends itself to machine learning applications.
- Pytorch – a Python package that is used for data analysis functions. Tensor functions library
- BERT – a Python package that implements NLP features
- Tkinter – a Python Package that provides a GUI wrapper for Python

Project Take-Aways

Any application is subject to improvements. Here are a few improvements to the application that would make it more robust. Rather than storing the stories as hard-coded text in the program, a file processing menu could be added to allow the user to read the stories stored in an external location. The user could simply select any story stored on a file system. This would provide more flexibility and allow for a theoretically unlimited number of stories, The application could also offer an “Edit” facility, so the user could alter the position of the substitute words. A new routine could be added predicated by a ‘Prepare’ button that would automatically edit the file for the user by randomly placing the missing word indicators into the file. The user would simply select a text story file from the local files system and the application would automatically prepare it for predictive processing. The substitution positions in the text are difficult to read. The substitution word could be written with bold typeface, surrounded by additional spaces, or utilize different font colors to distinguish them from the remainder of the text. Some of the resulting substitutions were still erroneous. Utilizing a different trained dataset might rectify these poor substitutions. Allowing the application to train itself to the process might also correct the inadequate substitutions.

The BERT transformer did not predict the masked adjacent adjectives or adverbs consistently. This might be an issue with adjacent word masks. BERT also encountered prediction difficulties when a masked word was adjacent to end of sentence punctuation. This was rectified by eliminating the punctuation. Perhaps the data needs to undergo a cleansing process before being fed to BERT. This process would eliminate punctuation. More research is needed into the placement of BERT word masks. Is there a special character needed to indicate that there are several adjacent words to be predicted?

Conclusion

With larger faster computers, the ability to store and process Big Data, and the growing sophistication of Neural networks, Natural Language Processing is entering a new era. Computers are getting closer and closer to achieving seamless human conversation Scientists are moving toward patterning the computer with the workings of the human brain. A computing device would have the capability to reason and feel emotions much like humans. The eventual goal of Natural Language Processing is

Computing devices will be able to reason for themselves. There will no longer be static literals and static data sources. The machine will develop its response based upon its surrounding environment. I might ask the machine, “Can I work into the garden today?” It may answer yes in one situation, but no in another situation, perhaps based upon local weather conditions. It will not run the same static internal program each time it answers a question. It will think for itself. The final goal of Natural Language processing is to achieve language abilities nearly identical to that of a human.

Bibliography

Alammar, Jay. “The Illustrated BERT, ELMo, and Co. (How NLP Cracked Transfer Learning).” *Github.Io*, 2018, jalammar.github.io/illustrated-bert/.

“BERT — Transformers 2.8.0 Documentation.” *Huggingface.Co*, huggingface.co/transformers/model_doc/bert.html.

Bird, Steven, et al. *Natural Language Processing with Python*. Winnipeg, Media Production, Manitoba Education And Advanced Learning, 2014.

Caressa, Paolo. “Artificial Intelligence: ‘The New Electricity.’” *Codemotion Magazine*, 10 Apr. 2020, www.codemotion.com/magazine/dev-hub/machine-learning-dev/artificial-intelligence-the-new-electricity/. Accessed 13 Sept. 2020.

Dai, Andrew M., and Quoc V. Le. “Semi-Supervised Sequence Learning.” *ArXiv:1511.01432 [Cs]*, 4 Nov. 2015, arxiv.org/abs/1511.01432. Accessed 4 May 2020.

Devlin, Jacob, et al. *BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding*. 24 May 2019, p. 16, arxiv.org/abs/1810.04805. Accessed 12 Sept. 2020.

Faria, Andreia V., et al. “Brain MRI Pattern Recognition Translated to Clinical Scenarios.” *Frontiers in Neuroscience*, vol. 11, 20 Oct. 2017, 10.3389/fnins.2017.00578. Accessed 13 Sept. 2020.

Manning, Christopher D., et al. “Emergent Linguistic Structure in Artificial Neural Networks Trained by Self-Supervision.” *Proceedings of the National Academy of Sciences*, 3 June 2020, www.pnas.org/content/early/2020/06/02/1907367117, 10.1073/pnas.1907367117. Accessed 13 Sept. 2020.

Mega, Laurie. “What’s the Difference Between NLP, NLU, and NLG?” *MarketMuse*, 10 Oct. 2019, blog.marketmuse.com/whats-the-difference-between-nlp-nlu-and-nlg/. Accessed 13 Sept. 2020.

Rani Horev. "BERT Explained: State of the Art Language Model for NLP." *Medium*, Towards Data Science, 10 Nov. 2018, towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270.

Vajpayee, Sarthak. "Understanding BERT — (Bidirectional Encoder Representations from Transformers)." *Medium*, 6 Aug. 2020, towardsdatascience.com/understanding-bert-bidirectional-encoder-representations-from-transformers-45ee6cd51eef. Accessed 13 Sept. 2020.