How NLP can give us more clues in deciphering cryptic messages

Rayan Wali
Cornell University
Ithaca, NY 14850
rsw244@cornell.edu

NLP, abbreviated for *Natural Language Processing*, is a sub-field within Artificial Intelligence that serves as a tool used in all of our speech-recognizing and translating devices, such as Google Home and Amazon Alexa. It can take two forms: text and speech recognition. We witness it daily, however, most of the times, we do not recognize the processes behind what we see. It is not simple as it seems for computer systems to comprehend and translate spoken human languages, such as English. Rather, it involves a plethora of data — from the use of millions of vectors to hidden Markov chains to store similarities or linkages within data sets and categorize such links to help decipher the same words or phrases appearing in other sentences. This speech and text recognition algorithm could be used to decipher cryptic messages that cannot be understood by humans. For instance, this algorithm is commonly used by government agencies, such as the FBI, to decode important messages and give them much needed clues for an ongoing investigation.

Keywords: NLP; Artificial Intelligence; Markov Chains.

1. Where we experience NLP on a daily basis

NLP is so commonly installed within almost all of our devices, whether it is smartphones or home appliances. Detecting a keyword from a phrase is a complex process for the computer, involving a large data set in which it can match words to. Essentially, a computer goes through a pattern matching algorithm, in which it has to compare the structure of current data with existing, already-stored data. We will explore this algorithm in more detail in our later sections of the paper.

2. The Pattern Matching Algorithm

Pattern matching involves a sequence of comparisons made with regular expressions, and may be part of speech or textual analysis. Both speech and textual analysis somewhat rely on machine learning, however, textual analysis is simpler as it does not have to make any approximations like speech pattern analysis has to, which will be discussed more in detail later in the text. The way textual analysis works is that *regular expression* templates are stored in a database, with words mapped to machine language, which computers can recognize. To achieve this, the computer translates sample text, for training purposes, to a sequence of integers by converting each character of a string to an integer value, which is commonly called the

"ASCII" code of the character — an universal standard for computers to interpret text. In ASCII, 'A' corresponds to 65, 'B' to 66, and so forth. Then, the ASCII code is converted to binary form, taking form of machine language. This completes the process, and now the original text has been made computer-recognizable. However, for larger texts, a lot of memory could be saved by matching the structure of the text to the trained regular expression templates. Note that the text suggestions we receive on our smartphones, tablets, or computers involve a more complicated process, which may be implemented several different ways, including machine learning, Markov implementation, word vector operations, and more.

3. The Hidden Markov Model of NLP

The Hidden Markov Model (HMM) is a type of a Markov Model that is an implementation of natural language processing that is based upon probability of occurrence for characters. In general, a Markov chain is a form of a finite automaton that consists of states, with transitions (also known as weights) connecting them, which in our context, represent the probabilities of transitioning from one character to another. A Markov chain could be stored as a stochastic matrix. However, to determine which data structure is better to use in a particular situation, we have to consider memory. If the stochastic matrix is sparse, it would probably be a better option than Markov Chains, and if dense, then it would be better for the computer to use Markov Chains. HMM is a special Markov chain that could both be used in speech, printed character, or handwritten character recognition. The components and such uses of HMM are described as follows:

3.1. *Components of the HMM Model*

- ☐ The number of states, represented by *N*. The individual states can be denoted by the set {a, b, c , d, e}.

- ☐ The state transition probability matrix, denoted by the set $\{\mathrm{P}_{ij}\}, with$

$$P_{ij} = P(Z_{n+1} = s_j | Z_n = s_i), \qquad 1 \leq \text{i, j} \leq \text{N} \tag{1}$$

- ☐ The observation probability distribution, denoted by the set $\{\mathrm{b}_j(k)\}, with$

$$b_j(k) = P(X_n = v_k | Z_n = s_j), \qquad 1 \leq \text{j} \leq \text{N}, 1 \leq \text{k} \leq \text{M} \tag{2}$$

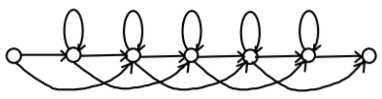
- ☐ The initial state distribution, $\pi = \{\pi_i\}, with$

$$\pi_i = P(Z_0 = s_i), \qquad 1 \leq \text{i} \leq \text{N} \tag{3}$$

3.2. *Applying HMM to Speech Recognition*

The Hidden Markov Model can recognize speech by representing it as a series of states. For instance, the phone model can be depicted as follows, with circles representing the states, self loops representing observation probabilities, and arrows to other states representing transitions.

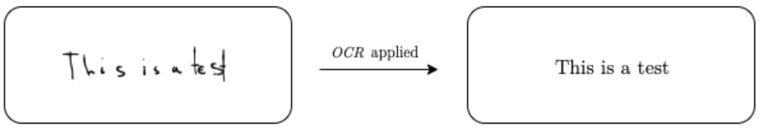
Figure 1: A Hidden Markov Model for phones



3.3. *Applying HMM to Handwritten Character Recognition*

The process of a computer translating handwritten characters to printed characters is known as *OCR*. To recognize characters, *OCR* uses a machine learning algorithm by matching the sequence of characters to pre-trained characters.

Figure 2: The OCR Process

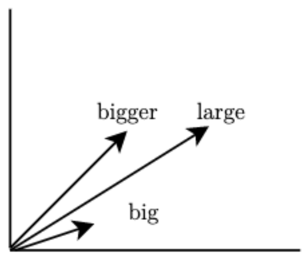


4. Predicting words with vector operations

A word vector is a vector with a weight of type string, representing the word. This approach works well for words with similar semantics, that is, meaning. For instance, the expression *"big" + "bigger"* = "large", and can be represented by the addition of word vectors with weights *"big"* and *"bigger."*[a] To construct such vectors from words, *Word2Vec* is a popular model that outputs word vectors from passing words as inputs. The following diagram illustrates a graphical representation of the word vector process in operation:

[a]Two words are said to be similar if the angle between their word vectors is small, by *Cosine Similarity.*

Figure 3: Operations on Word Vectors



5. The Big Picture of Speech Recognition Analysis

First, for a computer to interpret sounds, the words spoken are chopped down into several smaller pieces, with which, through pattern matching, the computer can understand with the help of previously stored patterns. Once it processes all characters of a word and all words of a phrase, it has successfully recognized speech. However, this type of pattern matching is not necessarily as accurate as it should be, since it could recognize an incorrect pattern. Accuracy is a priority for government agencies, for example, as they perform top-level work and need a highly accurate system of speech and text recognition. A more accurate processing algorithm would be one that involves probabilistic and mathematical models to make an estimate of what the character spoken actually is, for example, the implementation of a Hidden Markov Model we saw above. However, another emerging technology, Neural Networks, has the ability to perform speech recognition, similar to the Hidden Markov Model. Although it makes less assumptions regarding the structure of the language, it also requires more data to function accurately. In the past decade, tech companies considering the pros and cons decided to implement a combination of the two models in their electronics. We are likely and hopeful that will see such a similar advancement in the art of speech recognition in the coming years, one that will be accurate enough for robots to comprehend human instructions in a new, automated era.

References

1. Karlsson, Magnus. ”Hidden Markov Models.” http://www.math.chalmers.se/∼olleh/Markov_Karlsson.pdf. Accessed 16 Jan 2020.
2. Roche, Emmanuel, and Yves Schabes, eds. Finite-state language processing. MIT press, 1997.
3. Sedgewick, Bob, and Wayne, Kevin. “Markov Model of Natural Language.” Princeton University, The Trustees of *Princeton University*, 2004, www.cs.princeton.edu/courses/archive/spring05/cos126/assignments/markov.html.
4. Ward, Wayne. Hidden Markov Models in Speech Recognition. 17 Apr. 2012, www.cs.cmu.edu/∼roni/10601-slides/hmm-for-asr-whw.pdf.