Memory Networks for Understanding Language

Yuji Akimoto

October 1, 2017

Abstract

Although recurrent neural networks (RNNs) - particularly long short-term memory (LSTM) networks - have proven very effective at NLP tasks, they are not able to interpret longer pieces of text and reason about their meaning. This is the next step in the advancement of artificial intelligence, and can be thought of as the creation of a truly intelligent chatbot. This is an active area of research and has clear applications for virtual assistants such as Siri or Alexa, but presents challenges that are very different from, say, image classification.

1 Research Context

Current state-of-the-art work in text recognition relies on long short-term memory (hereon referred to as LSTM) networks, a particular architecture of a recurrent neural network (hereon referred to as RNN). Unlike feed-forward neural networks, such as convolutional neural networks which are commonly used in image classification, RNNs have inputs and outputs that are dependent on time, and are designed to work with data where ordering is very significant (such as videos or text). RNNs are able to deal with sequential data by storing information in state vectors: these are updated with each new input, and are used to compute each output - in short, they represent the "knowledge" that the network has accumulated so far. However, because the states are updated at each time step, it becomes very difficult to store long-term information. LSTMs improve on this architecture by adding mechanisms to forget and store information in "memory cells" - yet even these are extremely limited in capacity.

Recently, new algorithms called Memory Networks have been proposed. These include a much larger memory component and are therefore able to store longer-term dependencies within data. A memory network is composed of a memory m (an array of vectors) and four component networks (which may share parameters):

- **I (Input feature map):** convert incoming data to internal feature representation.
- **G (Generalization):** update memories given new input.
- **O (Output feature map):** produce new output (in internal feature representation) given memories.

- **R (Response):** convert output into desired format (most likely the same format as input)

Although memory networks have produced promising results on synthetic datasets such as bAbI, there are still many architectural decisions that have not yet been fully justified an require empirical testing. These include the way we decide what to write/not write to memory, the data structure we use for memory, how memories should be represented, how to build hierarchical memories and reasoning, etc.

2 Data/Design

Yelp data contains millions of reviews, each of which are associated with a star rating from 1-5. To begin, we hope to be able to answer the question of "What did this user think of this business?" where the label is the star rating. Here, we assume that the same rating means the same thing to different users, although this may not be the case. If we are able to achieve some level of success at this task, we hope to be able to move on to more general queries that a user of Yelp may have, such as "is the service good?" or "are the bathrooms clean?" that cannot currently be easily searched. The challenge in this context is that the answer may not be contained in the text at all, and our algorithm must learn to respond accordingly.

The design for our research is to apply existing architectures of memory networks to the Yelp dataset, before modifying to suit our problem. Since the entire network relies on sub-networks (such as one to generate a lower-dimensional representation of a sentence, for example), we will experiment with different combinations of sub-networks.

3 Methods

Previous methods have often used a bag-of-words model (with a temporal component) to generate the internal feature representation. Since not all Yelp users will use the same vocabulary and different people will talk about the same topic using different phrasing, we hope to incorporate a topic-modeling model (perhaps from past Yelp papers) which can be trained using the "categories" tag of business. We can then use a RNN to generate sparse representations of sentences that store the relations between topics in sentences. Research by Sukhbaatar et. al has shown that using adaptive memories (finding a variable number of supporting "memories" for the question), N-grams (count the occurrence of specific n-word combinations as opposed to individual words), and non-linearities (using a fully connected network to transform into the feature space) improve on the original memory network by Weston et. al and so this approach appears feasible.

The primary evaluation metric of our model will simply be its classification accuracy into the five different star ratings, or to simplify the problem, either a positive or negative review. There are currently various different methods being explored to solve language comprehension, so if this method is infeasible we may consider alternatives such as the Neural Turing Machine.

4 Significance

Teaching an algorithm to truly "understand" the content of text is one of the fundamental goals of artificial intelligence. The creation of such an algorithm would represent a huge step towards the generalizable intelligence found in humans, moving beyond the learning of very specific tasks.

A Q&A system on Yelp would allow users to find information about a business that may be very specific and would otherwise require them to read a very large number of reviews. Since the overall star rating only represents the business as a whole, it would also be possible to generate sub-category ratings that would allow both businesses and users to see what aspects of the business are popular and unpopular.

References

[1] J. Weston, S. Chopra, A. Bordes. *Memory Networks*. ICLR, 2015.

[2] S. Sukhbaatar, A. Szlam, J. Weston, R. Fergus. *End-To-End Memory Networks*. NIPS, 2015.