On the Expressivity of Language Models (Ling 190 Final Project)

Franklyn Wang
franklyn_wang@college.harvard.edu

Spring 2020

1 Abstract

In machine learning (ML) many times we attempt to model very complicated distributions, with astonishing success. However, a question which receives less attention is whether the models we have are expressive enough to fully capture the complexity of the target domain. By surveying a few approaches, we summarize limits to their expressivity.

2 Introduction

In recent years, language models have made incredible progress. For the purpose of this paper, we define a language model to be an estimate for the function $p(w|c)$ which is probability distribution over a word conditioning on the previous words, which we call the *context*.

For example, a modelling problem we might face is to find the probability distribution of the word following the context

The 44th President of the United States is

An ideal language model would assign some probability to the word "Obama" and a large amount of probability to the word "Barack".

An understanding of language modelling is crucial not only due to its myriad applications but because a mastery of language modelling might very well translate into a mastery of other language tasks via transfer learning. Indeed, currently almost all of the state-of-the-art approaches for various language tasks are built on top of large language models [RWC+19].

3 N-grams

The first approach to this problem was given by n-grams, which we also covered in class. n-grams are a direct approach to this problem. With n-grams, the general approach is to model

$$p(w|c) = \frac{\#(c, w)}{\#c}$$

where essentially we just count the number of times the word w follows the context c in the text and divide it by the total number of times that the context c appears. N-grams were the first kind of language model. N-grams are fairly accurate for $N = 1, 2$. There are certain issues with rare contexts in these situations but there are many ways for fixing them, such as Kneser-Ney smoothing [NEK94].

3.1 Limitations

While enjoying powerful theoretical guarantees, such a model has one major deficiency. Since there are exponentially many possible contexts, in a corpus of any reasonable length (even all the words that have been written in human history) this estimator will have a small sampling of the possible contexts. Essentially, the n-gram model treats every context as completely different. However, this is not an accurate depiction of human learning. Essentially, contexts that are very similar to previous ones are still quite useful in deciding the subsequent word. Incorporating this modification in, to determine the probability distribution of the next word, we would compute a similarity function $k(c_1, c_2)$ between two contexts. Then we could calculate a probability distribution which represents a generalization of the n-grams

$$p(w|c) \propto \sum_{(c',w')\in\mathcal{D}} 1_{w=w'} k(c, c') \tag{1}$$

Note that the n-gram model is simply the case in which $k(c, c')$ is one if the contexts are equal and zero otherwise. Intuitively, it represents the case in which we don't learn any non-obvious similarities between similar contexts. That represents a huge waste of data, because many contexts in the English language, like

"Charles Dickens wrote" and "Charles Dickens is the author of "

are essentially identical ([KLJ+20]).

However, it is hard to find similarity functions between contexts, and even harder to sum over a training corpus (which is often large). The aforementioned method also has its own problems. When a child learns that his friend's name is Adam, he uses the word Adam very fluently, not requiring having seen it multiple times in the past, as the above algorithm would require. This suggests that there are limitations to the ability of an n-gram model, and even the generalized one.

4 Word Embeddings

A far superior method to the above involves compressing the context c into a representation $f(c)$ (often a fixed-length vector), and then finding which words are most similar to that given context. In the Adam example, it's possible that the context "Adam's friend took him, " could be mapped to a vector who is very similar to the word "Adam". Computing a representation $f(c)$, however, is quite difficult.

To compute a representation $f(c)$ for a context, note that the set of words is just the set of length-one contexts. Thus, we should try to calculate compact representations for words first, or else there is no hope of finding representations for arbitrary contexts. It will also turn out that calculating $f(c)$ is easier with representations for each of the individual words. In fact, it has been suggested that one can just calculate the word embeddings, and do random computations, to get fairly similar results to the state of the art in calculating context embeddings. ([WK19])

To think about what information we would like a word embedding to contain, consider an analogy. Essentially, the English language has a large amount of redundancy and complicated relationships between words, like

$$\text{king} : \text{man} :: \text{queen} : \text{woman}$$

To learn these relationships, in essence we tell a computer to take words that often appear close to each other and make them have similar representations. Of course there's a concern that we'll simply put all the words at the same location, so we want words that appear far from each other to be spread apart on average. As it turns out, the word embeddings we produce satisfy the property that

$$\text{king} - \text{man} = \text{queen} - \text{woman}$$

We can measure the performance of these embeddings by looking at their performance on analogies, which serve as the most natural test for these. As it turns out, they perform amazingly well, so we can have confidence these word embeddings are capable of capturing many complexities of the English language.

5 Understanding Context Models

Many of the state of the art algorithms work by computing a context vector $\mathbf{h}_c \in \mathbb{R}^{d\times 1}$ as functions of word embeddings and then assigning log-probabilities given by

$$\log P(w_i|c) \propto \mathbf{h}_c^\top \mathbf{w}_i. \tag{2}$$

where $\mathbf{w}_i \in \mathbb{R}^{d\times 1}$ is the embedding vector corresponding to word w_i. Essentially, if the matrix $\mathbf{h}_c$ is more *aligned* with the word-embedding vector $\mathbf{w}_i$, it is more likely that the word w_i follows the context c. The word embeddings [MSC+13] themselves also capture valuable semantic information between the words. Recall that the word embeddings capture useful information, like

$$\text{man} - \text{woman} = \text{king} - \text{queen}$$

We can find an analogy to the image classification setting here. In image classification, we compute a representation $\mathbf{h}_i$ for an image, and then we take a dot product with another "embedding vector" (although in image classification it is often referred to as the "template vector") $\mathbf{e}_j$ for each of the classes to obtain the representation $\mathbf{h}_i^\top \mathbf{e}_j$.

6 The Softmax Bottleneck

[YDSC18] is a recent work which formalizes a problem called the "softmax bottleneck". Before we explain the formal linear algebraic reason for this phenomenon, we will see the origin of the softmax bottleneck with a toy example.

6.1 An illustrative example

Assume that the representation of the word man is $(1,1)$, woman is $(1,2)$, king is $(2,1)$, and queen is $(2,2)$. Intuitively, the first variable corresponds to the regality of the word, whereas the second variable corresponds to the gender. Again, note that the

$$\text{man} - \text{woman} = \text{king} - \text{queen}$$

relationship is preserved. Now, a linear relationship emerges. Specifically, from Eq. (2) we have

$$\frac{\mathbb{P}(\text{man}|c)}{\mathbb{P}(\text{woman}|c)} = \frac{\mathbb{P}(\text{king}|c)}{\mathbb{P}(\text{queen}|c)}$$

regardless of context c. We can now ask, is this a reasonable model for the English language? For most cases, this captures most of the essential complexity. The regality of a person and their gender can be reasonably assumed to be generated independently from one another. Yet due to the structure of the model, this restriction is present *in all circumstances*. No matter what the function h_c is, the above relationship holds for all contexts c. In most cases, this is fine. Consider the following sentences:

She, the

The monarch, the

The

In all of these sentences, it is reasonable to view the regality and gender of the following word as independent random variables. However, there are some subtle issues with this algorithm. Consider the following example:

Kristina Esfandiari is a

An omniscient language model would assign high probabilities to the word "woman" and to the word "king" (as Kristina Estandiari is a member of the band King Woman), but would not assign any probability to the words "queen" or "man".

6.2 Consequences

The above seems like it might be a simple case, but in fact it has deeper ramifications. More generally, linear algebra teaches us that for any embeddings $\mathbf{v}_1, \mathbf{v}_2, \ldots \mathbf{v}_n \in \mathbb{R}^K$, there exists $\leq K$ of them who form a basis. In other words, if the dimension of the embeddings has size 100, there are 100 words of English which essentially capture all of the variation in the English language. That would be incredible. It would mean that all the words in the English language were simple combinations of the others. Such a result would be a huge breakthrough for linguistics, but as far as we know, there is no basis of the English language.

6.3 The solution

To defeat the softmax bottleneck, [YDSC18] uses a different approach. The formulation in Eq. (2) provides certain probability models $P_{\theta_1}, P_{\theta_2}$ and so on. Now while each of these models individually has rank limitations, we can specify a mixing probability p_c dependent on context. This creates a new probability model

$$p(c)P_{\theta_1}(w|c) + (1 - p(c))P_{\theta_2}(w|c)$$

This probability model immediately has superior results with only one additional probability model. This method is known as a Mixture of Softmaxes. When this change is added, there are no limits to the rank of the model anymore.

7 Non-Parametric Approaches

Of course, one natural problem in the above approach is that every context must essentially fit into a very small (K) number of dimensions. That's extremely problematic; can K dimensions really capture the full meaning of a sentence? Since we are essentially forced to use a linear layer afterwards, the number of meanings is far smaller. What if we didn't impose the implicit restriction of linear decision boundaries, and instead used nearest neighbors? Similar contexts would likely have similar representations, which inspires the following idea, inspired by [GJU17] and refined by [KLJ+20]

To calculate the most likely successors of a context c, we simply compute its representation $\mathbf{h}_c$, and look at the successors of contexts which are *similar* to c, where a context c' is similar to a context c if $\mathbf{h}_{c'}$ and $\mathbf{h}_c$ are close in L^2 distance. For example, in Eq. (1) we can let the similarity function between c and c' be e^{-d^2}, where d is the L^2 distance between the representations of c and c'. The fact that this immediately obtains state-of-the-art results (without any tuning of the metric between representations) suggests that this method has immense potential.

7.1 Applications of non-parametric language models: Question Answering

Non-parametric language models have found many practical applications. For example, when we train machine learning models to learn the English language, it's very unclear what the model has actually learned. To a computer, a standard machine learning model, even one which handles very complicated English, is just a bunch of multiplications that somehow go from words to other words. However, in some cases this lack of transparency is exceedingly harmful. Consider a company which would like to use an artificial intelligence to respond to customer support requests. Many such companies exist. One main challenge in this space is that it could be potentially hazardous for one of these chatbots to begin producing information which is highly sensitive, as it would bring up privacy concerns. Perhaps a good demonstration of what might happen can be seen from this xkcd comic, inspired from the real life Heartbleed bug.

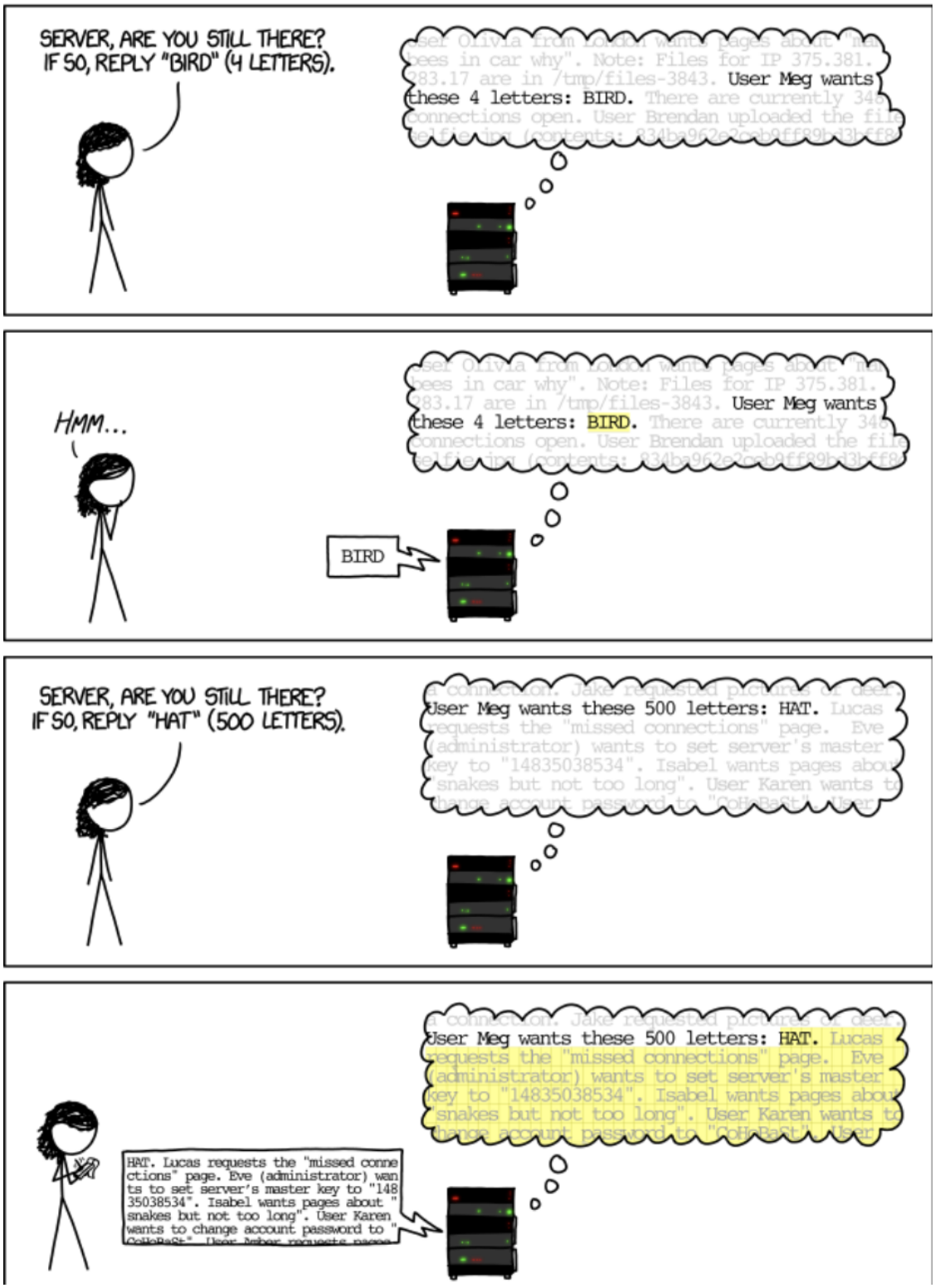

Figure 1: How Improperly Understood Language Models can have dangerous implications. Reused from https://xkcd.com/1354/

To combat this danger, one proposed solution is to require that the computer model produce an an-

swer that is contained within a set of documents. This is also helpful for modelling a superior question answering system. In general, one should expect that instead of having a language model memorize all the information in the world, it might be easier to teach it how to look up the answer from a repository from a set of documents. An analogy to humans could be that a human armed with Google is probably better at answering factual questions than any non-augmented human can be. If humans are better when paired with Google, are computers better too?

Note that under the restriction of documents, the standard language model is completely incapable of figuring out how to answer the question, because they rely on being able to generate any word that comes to mind. However, the non-parametric methods work well, because once we can establish a similarity function between various contexts, we can find the most similar context and use that document. Put another way, nonparametric methods allow one to answer "which context is most similar". Parametric methods are restricted to saying "what's the most likely follow up to this context". This approach (of answering questions by finding similar contexts in documents) was taken by [GLT+20] and immediately achieved incredibly impressive state of the art results. The idea is that first, we find the relevant documents using approaches similar to Google's pagerank. Then, we ask the model to find the most likely answer from there.

8 Conclusion

Overall, the field of natural language processing has only recently starting producing incredible results. Understanding language is a very difficult problem, and in some sense understanding language represents perhaps one of the ultimate tasks of understanding humans as a whole. We have come very far, and I hope that we will see huge breakthroughs soon. The discussion in the most recent sections suggests a new path forward for AI. Instead of trying to fit more into a deep neural network, it's possible that future systems will not be end-to-end deep neural networks but rather more complicated systems with many components, of which artificial intelligence comprises some but not all.

References

[GJU17] Edouard Grave, Armand Joulin, and Nicolas Usunier. Improving neural language models with a continuous cache. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[GLT+20] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Realm: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*, 2020.

[KLJ+20] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through Memorization: Nearest Neighbor Language Models. In *International Conference on Learning Representations (ICLR)*, 2020.

[MSC+13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119, 2013.

[NEK94] Hermann Ney, Ute Essen, and Reinhard Kneser. On structuring probabilistic dependences in stochastic language modelling. *Computer Speech & Language*, 8(1):1–38, 1994.

[RWC+19] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

[WK19] John Wieting and Douwe Kiela. No training required: Exploring random encoders for sentence classification. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[YDSC18] Zhilin Yang, Zihang Dai, Ruslan Salakhutdinov, and William W. Cohen. Breaking the softmax bottleneck: A high-rank RNN language model. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.