Grammaticality in neural natural language generation

A THESIS PRESENTED
BY
JOHNNY TIAN-ZHENG WEI
TO
THE COMMONWEALTH HONORS COLLEGE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
BACHELOR OF ARTS
IN THE SUBJECT OF
COMPUTER SCIENCE

UNIVERSITY OF MASSACHUSETTS, AMHERST
AMHERST, MASSACHUSETTS
SPRING 2019

Primary advisor: Brendan O'Connor
Secondary advisor: Brian Dillon Johnny Tian-Zheng Wei

Grammaticality in neural natural language generation

Abstract

Sequence to sequence (seq2seq) models are often employed in settings where the target output is natural language. However, the syntactic properties of the language generated from these models are not well understood. We explore whether such output belongs to a formal and realistic grammar, by employing the English Resource Grammar (ERG), a broad coverage, linguistically precise HPSG-based grammar of English. From a French to English parallel corpus, we analyze the parseability and grammatical constructions occurring in output from a seq2seq translation model. Over 93% of the model translations are parseable, suggesting that it learns to generate conforming to a grammar.

To provide a foundation for these results, the thesis first introduces natural language generation (NLG), the neural approach to NLG, and some introductory syntax. After introducing the seq2seq model, we connect its application in NLG and present our results in machine translation as a case study of seq2seq in the NLG domain. We propose potential future work to observe grammaticality in diversity sampling language generation settings and syntactic style in low-resource settings.

Contents

Listing of figures

This thesis is dedicated to my friends and family.

Acknowledgments

I AM AMAZED by those who have chosen to help me, without return. I wish I could repay them, but I am likely to stay indebted. At the very least, I will believe in karma, as they have built good ones with me.

Thank you Brendan! You made this thesis possible. Your investment in me makes it possible for me to pursue a research career. I will be taking what I have learned here, from you, forward in graduate school. Best of luck to you and the family you have started! Thank you Brian! Your investment in me, too, makes it possible for me to pursue a research career. The attention you gave me from freshman year is one of the reasons I will pursue research at all. I saw a lot more success in my research, because I had your support. Best of luck!

I'd like to thank Andrew McCallum for giving me opportunity, in many more ways than one. He also has amazing students and postdocs who mentored me: Ari Kobren, Nicholas Monath, Haw-Shiuan Chang, and Jeffery Flanigan. Thank you guys! I learned a lot of the technical side of research from you all.

I'd like to thank Noah Smith for taking me as a student in my sophomore summer. Many of the ideas in the thesis began while I was an intern at Noah's ARK. I received mentorship from some of his amazing students and postdocs: Omer Levy, Roy Schwarz, Chenhao Tan, Jesse Dodge, Lingpeng Kong, and Nelson Liu. Thank you guys! Nelson - best of luck in graduate school! See you around!

I'd like to thank Chris Potts and Sharad Goel for my summer at Stanford. I'd also like to give a big thank you to Dan Flickinger, for building the tools that made my research possible, and giving me feedback. Thank you Dan! The Center for Study of Language and Information facilitated this research, and helped me meet Khiem Pham. We completed much of the research as roommates, which I promise not to forget. Khiem - best of luck in everything you pursue! You rock - and I know it!

There are the countless people who took time out to talk to me, to give me feedback or advice. Some of these people include: Graham Neubig, Stephen Oepen, Michael Goodman, Nicholas Tomlin, and Ioannis Konstas. Thank you guys!

I'd like to give a special thanks to the 4 other members of my family - Yuji, Sherry, Victoria, and Dylan. Thank you to all my friends who made my college time special. If I could do it again, I'd wish to spend more time with you all. Thank you all!

1 Introduction

This thesis focuses on evaluating the quality of task-based language generation systems. Before conducting experiments, I had a few goals in mind for evaluation (which I still, and likely will continue to, keep close to my research agenda). My evaluation methods should

1. apply generally to any language generation task.
2. reflect model performance in practice.
3. be automatically computed.

and, as with any form of evaluation, reliably correlate with human judgment. Regrettably, this is a less than perfect introduction.

There are assumptions I have made in outlining these goals alone. The first is in "task-based" language generation. While our models and techniques seem to converge across language generation tasks, we have not yet seen models of a general language generation facility cut across these tasks (this may change, see Radford et al. [47]), as it has in language understanding research [45]. The second is that evaluation should "apply generally," but past research has, in contrast, seen success in metrics developed for specialized applications (e.g. summarization [35]). The final assumption is in correlation with human judgment. Evaluations based on estimating theoretical quantities can also be informative [23].

What does this leave for the content in this thesis? Evaluations that satisfy my three criteria may still be exciting as they are practical and leave room for the application of linguistic knowledge and natural language understanding technologies. This chapter will lay the groundwork for an evaluation method of grammaticality in language generation [65]. To do this, we will introduce a probabilistic model of language generation (§1.1), some tasks (§1.1.1, §1.1.2), the neural approach to generation (§1.3), and finally introduce some syntax from linguistics (§1.4).

1.1 Natural language generation

A wide range of natural language processing applications such as machine translation [27], dialog, and summarization require capability in natural language generation [NLG; 5]. Probabilistically, language generation may be viewed as

$$p(y_1, \ldots, y_T) = \prod_{k=1}^{T} p(y_k | (y_1, \ldots, y_{k-1})) \tag{1.1}$$

where each y_i is in some finite vocabulary Σ, and $(y_1, \ldots y_T)$ is a sequence of output tokens in the set of all possible sequences Σ^*. Factorizing over timesteps allows us to assign probabilities over Σ^* through the products of a tractable

problem: at timestep k, assign a probability over Σ for the next token y_k, given the previous history. A probabilistic model over sentences, or a *language model*, is a foundational tool [58] - perhaps high probability samples from this model are sentences we would wish to speak.

However, sentences are rarely spoken in vacuum. Our task at hand may require generating an English translation given some Chinese (§1.1.1), or a conversational response to a query (§1.1.2). In any case, our language generation problem can be cast to a conditional one as such

$$p(y_1, \ldots, y_T|c) = \prod_{k=1}^{T} p(y_k|c, (y_1, \ldots, y_{k-1})) \tag{1.2}$$

where c is the *context* of generation. In machine translation, $c = (x_1, \ldots, x_N)$ may be an input sequence of tokens in another language, but we should not limit our imaginations here - c may be a knowledge base [64], or even a molecular compound [69].

This probabilistic framework is adopted by many machine learning algorithms applied to language generation problems, including neural approaches described in §1.3. By far, this is not the only way of designating what we wish to speak, and §1.4 introduces some perspectives from the field of Linguistics.

1.1.1 Machine Translation

The act of translation is to transform a string in one language to a meaning (and possibly stylistically) equivalent string in another. An example for English to Chinese translation and vice versa from Google Translate[1] has been provided

[1] https://translate.google.com

below:

```
Reaching human-level fluency may be possible.
```
→ 达到人类流利可能是可能的。 (1.3)

达到人类的流利程度是有可能的。

→ `It is possible to reach human fluency.` (1.4)

besides the practical applications (e.g. resource accessibility [69] and national security [25]), translation is a difficult problem that merits our scientific study. Imagine, for a moment, the existence of a universal translator. For a model to achieve universal understanding, this model must pick up on properties intrinsic to human language [56]. To correctly translate from one language to another, this translator may have an intermediate representation that captures relevant aspects of meaning and style [29]. Refer to Koehn [31] for a short history and reference on statistical machine translation.

Now what makes a translation good? In the MT community, translations are often thought to have two dimensions: *adequacy* and *fluency* [9]. A translation is *adequate* if it conveys the necessary information. A translation is *fluent* if it sounds natural. These two dimensions may be relevant among other NLG tasks as well. For instance, summarization may take into account these two features in addition to a feature of summary length [36]. In practice, it makes sense to evaluate output translations on one dimension only - the question used to elicit intrinsic judgments from humans would be "How well does this translation reflect the meaning of the reference translation?" [8].

1.1.2 Dialog

The act of conversing is to respond thoughtfully (and possibly engagingly) given the context of the dialog. An example response from Siri is given below

Me: `Is fluency an issue for natural language generation?`

Siri: `Here's what I found on the web... [Search results]`

which, by my judgment, is unsatisfactory. There have been great philosophical arguments over the implications of responding satisfactorily. Turing [62] proposed a test (now known as the "Turing test"), of whether a machine could generate responses indistinguishable from a human. For a machine to pass this test provokes the question of whether this machine is intelligent. Turing argued that, yes, this is sufficient criteria to be deemed intelligent. Refer to Searle [53] for a (famous) counterargument about a room that translates Chinese.

Now what makes a response good? Perhaps we might first turn to our wine-drunk ancient Greek philosophers. Aristotle believed persuasion was tripartite - it could be achieved through the character of the speaker, the emotional state of the listener, or the argument (*logos*) itself. [48]. In addition, he taxonomized three forms of argumentative speech. There are the deliberative species, which advises or warns. Then there are the judicial species, which either defends or accuses. Finally, there is the third species, which does not aim to persuade, it tries to describe a deed in the orator's opinion.

We may want to turn to linguists for a more closely motivated perspective. To describe our choice of utterances, Grice [22] outlined five maxims. They are

- **Maxim of quantity.** Speakers give as much information as needed, and no more.
- **Maxim of quality.** Speakers are truthful, and do not give claims that are not supported by evidence.
- **Maxim of relation.** Speakers mentions only those that are pertinent to

the discussion.

- **Maxim of manner.** Speakers try to be as clear, orderly, and brief as possible.

However, these maxims are far from comprehensive and may overlap in their roles. These maxims are also hard to specify computationally. Note that there exists a Bayesian framework for pragmatic reasoning [21].

In practice, we may punt on our definitions and evaluate in a manner similar to MT. To determine whether a response was satisfactory, we can use a question such as "how appropriate was this response?" to elicit human judgments on a response's intrinsic quality.

1.2 Automatic evaluation

There are two main problems with human annotation of system output quality: 1) it is time-consuming and expensive - for certain research questions that involve hyper-parameter tuning or architecture searches, the amount of human annotation makes such studies infeasible [10, 39]. 2) It does not facilitate comparisons across research papers. Therefore, developing reliable automatic evaluation metrics for NLG is important.

There are at least two challenges in automatic metrics. First, the metric must measure many dimensions of the output that are relevant. Second, in evaluations of many NLG tasks, several outputs may be acceptable. In translation, we may be concerned with a translation's fluency (how natural does it sound?) and adequacy (is the meaning correct?), and several outputs may be valid translations.

In domains such as machine translation (MT), a system's extrinsic value is both hard to define and measure, and intrinsic human judgments of a system's output quality have been the main indicator of progress in the field. [9] For those domains best evaluated intrinsically, automatic metrics, which both are computed automatically and correlate highly with human judgment, are ideal. If sufficiently correlated, a metric can be used as a surrogate evaluation, which may

be useful in developmental cycles. Therefore, the application of such metrics are dependent on studies of their validity [49, how well does the metric correlate with human judgment?]. In MT, BLEU [43] has seen widespread use, and, consequently, its validity has been extensively studied. [11, inter alia]

1.2.1 BLEU

The most commonly used metric of evaluation for machine translation is BLEU. [43] Metrics based on n-gram overlaps such as BLEU [43] and ROUGE [36], originally designed for evaluating machine translation and summarization, are often widely adopted to broader NLG domains [59]. BLEU score attempts to correlate human judgment of translations using n-grams from the ouput and reference text aggregated over a test corpus. BLEU is calculated as

$$\min(1, 1 - |r|/|c|)) \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

where $|r|$ and $|c|$ are the length of the reference and output sentence, respectively, p_n is the weighted (modified) n-gram precision, and w_n is the weight for the n-gram. The metric was inspired from the word error rate by the speech recognition community, but uses n-grams for variants of translation. Papineni et al. [43] claim that the unigrams capture adequacy, and the higher order n-grams capture fluency. The right hand term is the sentence brevity penalty to control for recall, computed over the corpus.

1.3 Neural approach to language generation

Traditional approaches to language generation used pipelines of several stages, including stages such as task planning and surface realization [50]. Recent advancements in general sequence-to-sequence models provide new data-driven alternatives [seq2seq; 61]. Applying these neural seq2seq models to language generation has now become mainstream likely due to their ease of

implementation, flexibility, and expressivity. Refer to Goldberg [19] for a review of neural techniques in natural language processing. In some fields, neural models even achieve state of the art performance [8].

Among other lines of reasoning [38], neural models likely achieve their high performance in NLP due to end-to-end training. These models leverage large datasets and learn all their complex behavior completely from data, reducing the need for error-prone, low coverage, and hand-engineered features. If anything, this makes the models easier to implement. End-to-end training is made possible because neural models are fully differentiable functions of input data and parameters. Once a loss function is specified (and calculated with a given training instance), gradients of the trainable parameters can be calculated through backpropagation [51]. With the gradients of the loss w.r.t. the parameters, we can adjusted these parameters with stochastic gradient descent (SGD) or Adam [30] to minimize the loss.[2] For my research, the community generously provides a number of high-quality, open-source autodiff libraries [1, 42, 44] that I built my thesis research on.

1.3.1 Recurrent neural networks

A recurrent neural network [RNN; 15] is a parameterized function that can be recursively applied to a *sequence* of vector input. An RNN $g(\mathbf{h}^{(t-1)}, \mathbf{x}^{(t)})$ is specified with the equations given below (notation adopted from Goodfellow et al. [20])

$$\begin{aligned} \mathbf{h}^{(t)} &= \tanh(\mathbf{b} + \mathbf{W}\mathbf{h}^{(t-1)} + \mathbf{U}\mathbf{x}^{(t)}) \\ \hat{\mathbf{y}}^{(t)} &= \text{softmax}(\mathbf{c} + \mathbf{V}\mathbf{h}^{(t)}) \end{aligned} \tag{1.5}$$

[2] Training a model that achieves good generalization by minimizing the loss function is nearly art. There are important details, which are usually compacted into the "training details" section of an NLP paper. We will not discuss training in-depth, and refer the reader to Neubig [41] for a tutorial.

where $\{\mathbf{W}, \mathbf{U}, \mathbf{V}\}$ are trainable matrix parameters and $\{\mathbf{b}, \mathbf{c}\}$ are vector ones. The function returns $[\mathbf{h}^{(t)}, \mathbf{o}^{(t)}] = g(\mathbf{h}^{(t-1)}, \mathbf{x}^{(t)})$ and is recursively applied through inputs $\mathbf{x}^1, \ldots, \mathbf{x}^n$, passing hidden states $\mathbf{h}^{(t)}$ to the next call of g at each timestep t (with an initial trainable hidden state $\mathbf{h}^{(0)}$).

Intuitively, the hidden state at each timestep is a nonlinear combination of the previous hidden state and the current input. There are several outputs of the RNN that you may wish to use. The first is the last hidden state $\mathbf{h}^{(n)}$, obtained after the last input $\mathbf{x}_n$ is applied, which would, in theory, contain information from all of the inputs $\mathbf{x}_1, \ldots, \mathbf{x}_n$. In the case of sequence classification, you may directly wish to use $\mathbf{o}^n$, which utilizes the hidden state $\mathbf{h}^{(n)}$. It is also possible to use the sequence of hidden states $\{\mathbf{h}^{(i)} : 1 \leq i \leq n\}$. For part-of-speech tagging, you may directly use the sequence of outputs $\{\mathbf{o}^{(i)} : 1 \leq i \leq n\}$.

The reader may notice that the hidden states passes through a sequence of $\tanh(\mathbf{a}^{(t)})$ functions. There are several things to notice about this non-linearity. First, with some non-linearity, RNNs have been proven to have equivalent expressive power to Turing machines [57] - if the parameters are set correctly. In practice, backpropagation + gradient descent is one of the few known practical methods to train neural models, and backpropagating through a series of non-linearities may result in the *vanishing gradient* problem. However, even if you are sucessfully backpropagating gradients in your model, the correct setting of the parameters (those required to achieve the model's theoretical potential) may not be reached [66].

The vanishing gradient problem motivates augumenting RNNs with linear components, which are lossless in gradient calculation. The long short-term memory [LSTM; 26] is usually the variant of choice in NLP. The equations of an

LSTM function $g(\mathbf{h}^{(t-1)}, \mathbf{c}^{(t-1)}, \mathbf{x}^{(t)})$ are given below

$$\begin{aligned}
\mathbf{c}'^{(t)} &= \tanh(\mathbf{b} + \mathbf{W}_{\mathbf{c}'}\mathbf{h}^{(t-1)} + \mathbf{U}_{\mathbf{c}'}\mathbf{x}^{(t)}) \\
\mathbf{i}^{(t)} &= \sigma(\mathbf{b} + \mathbf{W}_{\mathbf{i}}\mathbf{h}^{(t-1)} + \mathbf{U}_{\mathbf{i}}\mathbf{x}^{(t)}) \\
\mathbf{f}^{(t)} &= \sigma(\mathbf{b} + \mathbf{W}_{\mathbf{f}}\mathbf{h}^{(t-1)} + \mathbf{U}_{\mathbf{f}}\mathbf{x}^{(t)}) \\
\mathbf{c}^{(t)} &= \mathbf{f}^{(t)} * \mathbf{c}^{(t-1)} + \mathbf{i}^{(t)} * \mathbf{c}'^{(t)} \\
\mathbf{o}^{(t)} &= \sigma(\mathbf{b} + \mathbf{W}_{\mathbf{o}}\mathbf{h}^{(t-1)} + \mathbf{U}_{\mathbf{o}}\mathbf{x}^{(t)}) \\
\mathbf{h}^{(t)} &= \tanh(\mathbf{c}^{(t)}) * \mathbf{o}^{(t)} \\
\hat{\mathbf{y}}^{(t)} &= \text{softmax}(\mathbf{h}^{(t)})
\end{aligned} \tag{1.6}$$

where $\mathbf{c}^{(t)}$ are claimed to be memory cells. The $\mathbf{i}^{(t)}$ and $\mathbf{f}^{(t)}$ are input and forget gates, respectively, which control how much information flows in and out of the memory cells based on the present input $\mathbf{x}^{(t)}$ and previous hidden state $\mathbf{h}^{(t-1)}$. As you may notice, each $\mathbf{c}^{(t)}$ is a weighted linear combination of $\mathbf{c}^{(t-1)}$ and $\mathbf{c}^{(t-1)}$, so the gradient for the memory cells should not easly vanish over long timesteps.

LSTMs have been found to be strong models on many different sequential modeling related to language. These models can be used for parse reranking [12], part-of-speech tagging, and named entity recognition. The hidden representations are also useful components in larger neural architectures e.g. for classifying entailment [13].

1.3.2 Sequence to sequence networks

Refer back to probabilistic language generation defined in Equation 1.2 - we now introduce sequence to sequence networks [seq2seq; 61] to model this probability. This neural approach builds on LSTMs introduced in §1.3.1. A *decoder* LSTM can parameterize the timestep factorized probability

$$p(y_k|c, (y_1, \ldots, y_{k-1}) \tag{1.7}$$

if $\mathbf{o}^{(t)}$ is a distribution over all possible output tokens in Σ. To condition on a context c, the decoder can be passed a representation of the input. This representation may be a vector representation, where

$$c = f(x_1, \ldots, x_N) \tag{1.8}$$

and is parameterized by another *encoder* LSTM. The vector taken from the encoder can be the last hidden state, or it can be a combination over all the encoder's hidden states.

Seq2seq models, like LSTMs, achieve good performance in a variety of NLP tasks. The most notable includes machine translation [33], where neural seq2seq models are often state of the art, and these models are also applied in large commercial settings. However, the applications do not end there - seq2seq models can also be used for style transfer [17] and story generation [70].

1.3.3 Syntactic capabilities of RNNs

Sequence to sequence models [seq2seq; 3, 61] have found use cases in tasks such as machine translation [68], dialogue agents [63], and summarization [52], where the target output is natural language. However, the decoder side in these models is usually parameterized by gated variants of recurrent neural networks [26], and are general models of sequential data not explicitly designed to generate conforming to the grammar of natural language.

In Linzen et al. [37], the ability of recurrent neural networks and RNN-based language models to predict subject-verb agreement was evaluated. This section focuses on the language modeling results. Data was generated by taking sentences from Wikipedia with present tense verbs, and a parser is then used to label the number of intervening nouns and attractors. There may be several intervening nouns, and a few of them which differ in number from the main verb making them attractors. Attractors are defined to be intervening nouns that differed in number with the subject. For the RNN the last intervening noun influenced the error rate if it was an attractor. Another observation was that the

The **keys** to the cabinet *are* on the table. (1.9)

Figure 1.3.1: An example of a grammatical illusion caused by agreement attraction. The bold "keys" is the main subject and should agree in number with the main verb "are". However, human tendencies have been shown to associate the number of "are" with the closest intervening noun (underlined) [7]. In the case of "the cabinet", this noun is not only intervening, but also an attractor - its number differs from the subject, increasing the acceptability of an ungrammatical sentence ("the keys to the cabinet *is*...").

error rate is larger for singular agreement attraction than plural agreement attraction, which is a result seen in humans as well. When there is no agreement attraction, prediction for plural verbs were more likely to be wrong (singular is the majority class, where there are about 66%/33% singular/plural cases.).

For language models, the author tests ability of subject-verb agreement by inspecting the conditional distribution of producing the verb with the correct number inflection, e.g. $p(sing) < p(pl)$ or $p(pl) < p(sing)$. In the case of the verb "to be", we inspect $p(is) > p(are)$ when a singular subject was mentioned. Two language models were tested, the author's own 50 dimension 1 layer LSTM, and Google's enormous character level billion word LM (pre-trained). For both of these language models, the first attractor causes a 30% drop in accuracy. Up to 4 attractors, both the Google LM and the small LM perform worse than the majority baseline. Unfortunately, a slightly different replication of the results using a much more powerful language model has different conclusions.

1.4 Introductory syntax

Language is often thought to be arbitrary. There may very well be no limit to the ideas that we may express. The quote below

(1) The ships hung in the sky, much the way that bricks don't.

from Douglas Adams's book, the Hitchhiker's Guide to the Galaxy, clearly

conveys an imagined scenario that has yet been witnessed by anyone in the past, present, or future. However, a slight rearrangement of (1) below

(2) The ships hung the in sky, much the way that bricks don't.

now no longer sounds like it could have been a sentence Douglas Adams had written - a typo perhaps. Even a small vocabulary provides a combinatorial explosion of possible sequences, yet English is only a small fraction of those. As you and I both can quickly identify the membership of sequences of words within the restricted subset, there must be rules governing what can and cannot be English. Syntacticians, linguists who study syntax, define syntax roughly as

> The laws that govern how words combine into sentences.

and presupposes what words and sentences are [28]. Speakers can agree on most cases of what these are, and we will begin our discussion with our shared notions of words and sentences.

1.4.1 Grammaticality

The three sentence below highlight the phenomenon that concerns syntax

(3) The cat in the hat is delighted.

(4) The cat understands that emotional logic floats.

(5) * The in cat the hat is delighted.

where there is a salient difference between (5) and (3, 4), as the former example is a sentence that is inconceivable to be spoken by any English speaker. In the latter two examples, we can agree both are speakable sentences. The study of syntax is built on the assumption that this difference is real, and we are mainly interested in systems that account for this difference. We define (3) and (4) to be *grammatical*, but (5) to be ungrammatical, and use a preceding "*" to denote the difference. Note that there is also a salient difference in (4) and (3), where (4) is

non-sensical. Syntax is not interested in this distinction and we leave such details to semanticists.

In (3,4,5) I decided the grammaticality of sentences for illustration purposes, a methodology syntacticians term grammaticality judgments. There are two appealing properties of using grammaticality judgments like the ones above to study syntax. First, as an English speaker, I can reliably posit that since (5) is unspeakable to myself, other English speakers will share my intuition (which you will see throughout my tutorial!). Second, we may test sentences with phenomena that are the "edge cases" of our current knowledge of syntax. For instance, the judgments below

(6) The cat sings.

(7) The cat sings songs.

might lead us to hypothesize that when "songs" is appended to a grammatical sentence, it continues to be grammatical. A more capable syntactician could think of the counter-examples

(8) The cat sings songs.

(9) * The cat sings songs songs.

which shows, in fact, that our hypothesis need to be revised. This is the bitter sweet moment when we realize that an underlying mechanism is not what it seemed. Grammaticality judgements are the syntactician's main tool for refining our hypothesis of language syntax.

2

Grammaticality in neural machine translation

Sequence to sequence (seq2seq) models are often employed in settings where the target output is natural language. However, the syntactic properties of the language generated from these models are not well understood. We explore whether such output belongs to a formal and realistic grammar, by employing the English Resource Grammar (ERG), a broad coverage, linguistically precise HPSG-based grammar of English. From a French to English parallel corpus, we analyze the parseability and grammatical constructions occurring in output from a seq2seq translation model. Over 93% of the model translations are parseable, suggesting that it learns to generate conforming to a grammar. The model has trouble learning the distribution of rarer syntactic rules, and we pinpoint several

constructions that differentiate translations between the references and our model.

2.1 Related work

2.1.1 Understanding the language capability of neural models

Previous work in recurrent neural network based recognizers on artificial languages has studied the performance on context-free and limited context-sensitive languages [18]. More recent research in this setting provide methods to extract the exact deterministic finite automaton represented by the RNN based recognizers of regular languages [67]. These studies give exact analyses of RNN recognizers for simple artificial languages.

In the evaluation of language models in natural language settings, recent work analyzes the rescoring of grammatical and ungrammatical sentence pairs based on specific linguistic phenomenon such as agreement attraction [37]. These contrastive pairs have also found use in evaluating seq2seq models through rescoring with the decoder side of neural machine translation systems [54]. Both studies on contrastive pairs evaluate implicit grammatical knowledge of a language model.

2.1.2 Applications of HPSG-based grammars

HPSG-based grammars have found use in evaluating human produced language. To determine the degree of syntactic noisiness in social media text, parseability under the ERG was examined for newspaper and Twitter texts [4]. In predicting grammaticality of L2 language learners with linear models, the parseability of sentences with the ERG was found to be a useful feature [24]. These studies suggest parseability in the ERG has some degree of linguistic reality.

Our work combines analysis of neural seq2seq models with an HPSG-based grammar, which begins to let us understand the syntactic properties in the model output. Recent work most similar to ours is in evaluating multimodel deep

learning models with the ERG [34]. While their work uses the ERG for language generation to test language understanding, we evaluate language generation with the parsing capabilities of the ERG, and study the syntactic properties.

2.2 Head-driven phrase structure grammars

A head-phrase structure grammar [HPSG; 46] is a highly lexicalized constraint based linguistic formalism. Unlike statistical parsers, these grammars are hand-built from lexical entries and syntactic rules. The English Resource Grammar [16] is an HPSG-based grammar of English, with broad coverage of linguistic phenomena, around 35K unique lexical entries, and handling of unknown words with both generic part-of-speech conditioned lexical types [2] and a comprehensive set of class based generic lexical entries captured by regular expressions. The syntactic rules give fine-grained labels to the linguistic constructions present.[1] While the ERG produces both syntactic and semantic annotations, we focus only on syntactic derivations in this study.

Suitable to our task, the ERG was engineered to capture as many grammatical strings as possible, while correctly rejecting ungrammatical strings. Parseability under the ERG should have linguistic reality in grammaticality. Ideally, there will be no parses for any ungrammatical string, and at least one parse for all grammatical strings, which can be unpacked in order of scores assigned by the included maximum entropy model. We make a distinction between parseability and grammaticality. For our purposes of evaluating with a specified grammar, we consider the parseability of sentences under the ERG in §2.4.1, regardless of human grammaticality judgments. In §2.4.2, we manually annotate unparseable sentences for English grammaticality.

All experiments are conducted with the 1214 version of the ERG, and the LKB/PET was used for all parsing [14]. We use the default parsing configuration (command line option "`--erg+tnt`"), which uses a parsing timeout of 60

[1] A list of rules types and their descriptions can be found at `http://moin.delph-in.net/ErgRules`.

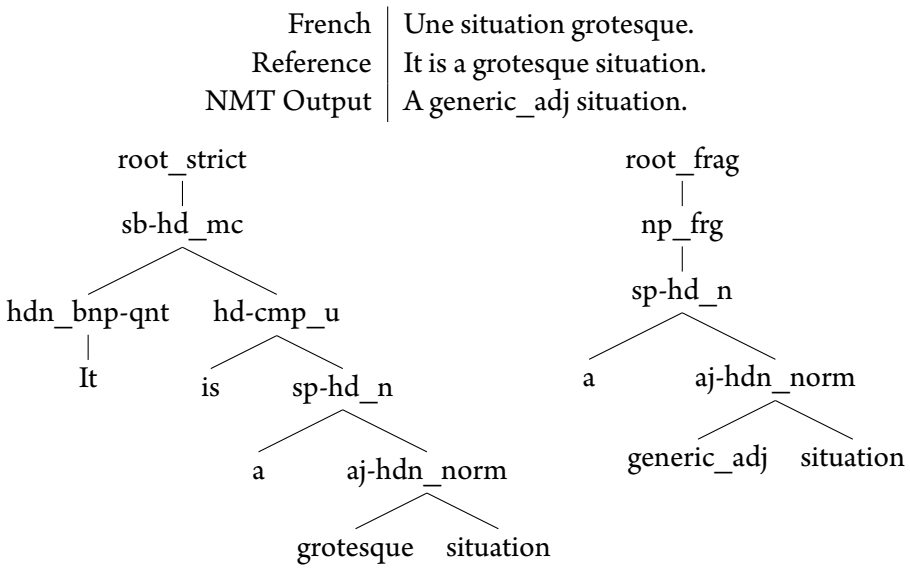

Figure 2.2.1: A test set source-reference pair and the NMT translation. Below are parser derivations in the ERG of both the reference and NMT translation. The ERG is described in §2.2. Non-syntactic rules have been omitted. The NMT model is trained and tested only on sentence pairs where the reference is parseable by the ERG. The NMT translation may not always be parseable. Analysis on model output parseability in §2.4.1.

seconds. A sentence is labeled unparseable either if the search space contains no derivations or if not a single derivation is found within the search space before the timeout. Figure 2.2.1 shows a simplified derivation tree.

2.3 Unparseability and ungrammaticality

This section establishes a connection between ERG (un-)parseability of model output and ungrammaticality, which will lay the foundation of future experiments in §3. To this end, we setup a French to English (FR $\rightarrow$ EN) neural machine translation system which we now refer to as NMT. Our goal was to test a baseline system for comparable results to machine translation and seq2seq models.

2.3.1 Dataset

From 2M French to English sentence pairs in the Europarl v7 parallel corpora [32], we subset 1.6M where the English/reference sentence was parseable by the ERG. For these 1.6M sentence pairs, we record the best tree of the English sentence as determined by the maximum entropy model included in the ERG. All sentence pairs we now consider have at least one English translation within our grammar, and we make no constraint on French. About 1.4M pairs were used for training, 5K for validation, and the remaining 200K reserved for analysis.

2.3.2 Out of vocabulary tokens

On the source-side French sentences, simple rare word handling was applied, where all tokens with a frequency rank over 40K were replaced with an "UNK" token. However, when handling rare words in the target-side English sentences, "UNK" will significantly degrade ERG parsing performance on model output. We replace our output tokens based on the lexical entries recognized by the ERG in our best parses (as in Figure 2.2.1's NMT output). This form of rare word handling is similar to the 10K PTB dataset [40], but with more detailed part-of-speech and regular expression conditioned "UNK" tokens. After preprocessing, we had a source vocabulary size of 40000, and a target vocabulary size of 36292.

2.3.3 Model

Our translation model is a word-level neural machine translation system with an attention mechanism [3]. We used an encoder and decoder with 512 dimensions and 2 layers each, and word embeddings of size 1024. Dropout rates of 0.3 on the source, target, and hidden layers were applied. A dropout of 0.4 was applied to the word embedding, which was tied for both input and output. The model was trained for about 20 hours with early stopping on validation perplexity with patience 10 on a single Nvidia GPU Titan X (Maxwell). We used the NEMATUS [55] implementation, a highly ranked system in WMT16.

	Strict		Informal		Unparseable
Source	**Full**	**Frag**	**Full**	**Frag**	
Ref	64.7	2.4	31.5	1.4	0.0
NMT	60.5	3.0	28.1	1.6	6.8
Δ	-4.2	+0.6	-3.4	+0.2	+6.8

Table 2.4.1: The distribution of root node conditions for the reference and NMT translations on the 200K analysis sentence pairs. Root node conditions are taken from the recorded best derivation. The best derivation is chosen by the maximum entropy model included in the ERG.

2.3.4 Translations

After training convergence on the 1M sentence pairs, the saved model is used for translation on the 200K sentences pairs left for analysis. A beam size of 5 is used to search for the best translation under our NMT model. We parse these translations with the ERG and record the best tree under the maximum entropy model. We have parallel data of the French sentence, the human/reference English translation, the NMT English translation, the parse of the reference translation, and the parse of NMT translation (if it was grammatical). Note that the NMT translation may have no parse.

2.4 Results

2.4.1 Parseability

The NMT translations for the 200K test split were parsed. Parsing a sentence with the ERG yields one of four cases:

- Parseable. A derivation is found and recorded by the parser before the timeout. The best derivation is chosen by the included maximum entropy in the ERG. About 93.2% of the sentences were parseable.
- Unparseable due to resource limitations. The parser reached its limit of

either memory or time before finding a derivation. This constitutes about 3.2% of all cases, and 47% of unparsable cases.

- Unparseable due to parser error. The parser encountered an error in retrieving lexical entries or instantiating the parsing chart. This constitutes about 0.5% of all cases, and 8% of unparsable cases.

- Unparseable due to exhaustation of search space. The parser exhausted the entire search space of derivations for a sentence, and concludes that it does not have a derivation in the ERG. This constitutes about 3.1% of all cases, and 45% of unparsable cases.

The distribution of the root node conditions for the reference and NMT translation derivations are listed in table 2.4.1, along with the parseability of the NMT translations. Root node conditions are used by the ERG to denote whether the parser had to relax punctuation and capitalization rules, with "strict" and "informal", and whether the derivation is of a full sentence or a fragment, with "full" and "frag". Fragments can be isolated noun, verb, or prepositional phrases. Both full sentence root node conditions saw a decrease in usage, with the strict full root condition having the largest drop out of all conditions. Both fragments have a small increase in usage.

2.4.2 Grammaticality

Out of the 14K unparseable NMT translations, there are 6.2K translations where the parser concluded unparseability after exhausting the search space for derivations. We will refer to these examples as "exhaustively unparseable." To understand the relation between English grammaticality and exhaustive unparseability under the ERG, two linguistics undergraduates (including the first author) labeled a random sample of 100 NMT translations from this subset. We sampled only those translations with less than 10 words to limit annotator confusion. Annotators were instructed to assign a binary grammatical judgment to each sentence, ignoring the coherence and meaning of the translation, to the

best of their abilities. Punctuation was ignored in all annotations, although the ERG is sensitive to punctuation. When the sentence was ungrammatical, subject-verb agreement and noun phrase agreement errors were annotated.

Within our random sample, 60 sentences were labeled as ungrammatical. Of these ungrammatical sentences, 5 could be made grammatical if a subject-verb agreement error was corrected, and 5 other translations could be made grammatical by correcting an article or determiner attachment to a noun. One translation exhibited both forms of agreement attachment errors. Agreement attachment errors are better studied phenomenon [37, 54]. However, correcting these errors only fixes 18.3% of ungrammaticality that we observed in our sample.

Out of the 100 sampled NMT translations that have no ERG derivations, we found 35 to be grammatical. 5 test examples were excluded. These include two cases where the source sentences were empty, and three cases where the sentence was parliament session information. Both annotators found annotating to be challenging, and possibly better annotated on an ordinal scale. Out of the exhaustively unparseable random sample, 37% was found to be grammatical. The ERG may have grammar gaps for near grammatical sentences.

2.5 Conclusion

We investigated the connection between ERG unparseability and ungrammaticality for this neural seq2seq model, and these two analyses were made possible by the English Resource Grammar. Neural sequence to sequence models do not have any explicit biases towards inducing underlying grammars, yet was able to generate sentences conforming to an English-like grammar at a high rate.

2.5.1 Wei et al. [65] leaves much to be desired

Our NMT model had a number of favorable conditions facilitating generation of grammar-conforming output. There are at least two conditions: 1) The training set included over a million sentences. 2) Techniques to increase the diversity of

output was not applied. These two conditions do not always hold in other language generation settings. For tasks such as dialog, paraphrasing, and style transfer, datasets are magnitudes smaller and techniques to improve output diversity are often applied. This motivates our exploration in §3, where we test the grammaticality of these systems under these settings encountered in practice, to generalize our results for the broader NLG domain.

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek Gordon Murray, Benoit Steiner, Paul A. Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: A system for large-scale machine learning. In Kimberly Keeton and Timothy Roscoe, editors, *12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016, Savannah, GA, USA, November 2-4, 2016.*, pages 265–283. USENIX Association, 2016. URL https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi.

[2] Peter Adolphs, Stephan Oepen, Ulrich Callmeier, Berthold Crysmann, Dan Flickinger, and Bernd Kiefer. Some fine points of hybrid natural language parsing. In *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008, 26 May - 1 June 2008, Marrakech, Morocco*, 2008. URL http://www.lrec-conf.org/proceedings/lrec2008/summaries/349.html.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Bengio and LeCun [6]. URL http://arxiv.org/abs/1409.0473.

[4] Timothy Baldwin, Paul Cook, Marco Lui, Andrew MacKinlay, and Li Wang. How noisy social media text, how diffrnt social media sources? In *Sixth International Joint Conference on Natural Language Processing, IJCNLP 2013, Nagoya, Japan, October 14-18, 2013*, pages 356–364, 2013. URL http://aclweb.org/anthology/I/I13/I13-1041.pdf.

[5] Anja Belz, Albert Gatt, François Portet, and Matthew Purver, editors. *ENLG 2015 - Proceedings of the 15th European Workshop on Natural*

Language Generation, 10-11 September 2015, University of Brighton, Brighton, UK, 2015. The Association for Computer Linguistics. ISBN 978-1-941643-78-5.

[6] Yoshua Bengio and Yann LeCun, editors. *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL https://iclr.cc/archive/www/doku.php%3Fid=iclr2015:accepted-main.html.

[7] Kathryn Bock and Carol Ann Miller. Broken agreement. *Cognitive Psychology*, 23:45–93, 1991.

[8] Ondrej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn, and Christof Monz. Findings of the 2018 conference on machine translation (WMT18). In Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno-Yepes, Philipp Koehn, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana L. Neves, Matt Post, Lucia Specia, Marco Turchi, and Karin Verspoor, editors, *Proceedings of the Third Conference on Machine Translation: Shared Task Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, pages 272–303. Association for Computational Linguistics, 2018. ISBN 978-1-948087-81-0. URL https://aclanthology.info/papers/W18-6401/w18-6401.

[9] Ondřej Bojar, Christian Federmann, Barry Haddow, Philipp Koehn, and Lucia Specia Matt Post and. Ten years of wmt evaluation campaigns: Lessons learnt. In *Workshop on Translation Evaluation: From Fragmented Tools and Data Sets to an Integrated Ecosystem*, pages 27–36, Portoroz, Slovenia, 2016. URL http://www.cracking-the-language-barrier.eu/wp-content/uploads/LREC-2016-MT-Eval-Workshop-Proceedings.pdf.

[10] Denny Britz, Anna Goldie, Minh-Thang Luong, and Quoc Le. Massive exploration of neural machine translation architectures. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1442–1451. Association for Computational Linguistics, 2017. doi: 10.18653/v1/D17-1151. URL http://aclweb.org/anthology/D17-1151.

[11] Chris Callison-Burch, Miles Osborne, and Philipp Koehn. Re-evaluation the role of bleu in machine translation research. In Diana McCarthy and Shuly Wintner, editors, *EACL 2006, 11st Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, April 3-7, 2006, Trento, Italy*. The Association for Computer Linguistics, 2006. ISBN 1-932432-59-0. URL http://aclweb.org/anthology/E/E06/E06-1032.pdf.

[12] Do Kook Choe and Eugene Charniak. Parsing as language modeling. In Su et al. [60], pages 2331–2336. ISBN 978-1-945626-25-8. URL http://aclweb.org/anthology/D/D16/D16-1257.pdf.

[13] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 670–680. Association for Computational Linguistics, 2017. ISBN 978-1-945626-83-8. URL https://aclanthology.info/papers/D17-1070/d17-1070.

[14] Ann A. Copestake and Dan Flickinger. An open source grammar development environment and broad-coverage English grammar using HPSG. In *Proceedings of the Second International Conference on Language Resources and Evaluation, LREC 2000, 31 May - June 2, 2000, Athens, Greece*, 2000. URL http://www.lrec-conf.org/proceedings/lrec2000/pdf/371.pdf.

[15] Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14(2): 179–211, 1990. doi: 10.1207/s15516709cog1402_1. URL https://doi.org/10.1207/s15516709cog1402_1.

[16] Dan Flickinger. On building a more efficient grammar by exploiting types. *Natural Language Engineering*, 6(1):15–28, 2000. URL http://journals.cambridge.org/action/displayAbstract?aid=58601.

[17] Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. Style transfer in text: Exploration and evaluation. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI*

Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, pages 663–670. AAAI Press, 2018. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17015.

[18] Felix A. Gers and Jürgen Schmidhuber. LSTM recurrent networks learn simple context-free and context-sensitive languages. *IEEE Trans. Neural Networks*, 12(6):1333–1340, 2001. doi: 10.1109/72.963769. URL https://doi.org/10.1109/72.963769.

[19] Yoav Goldberg. *Neural Network Methods for Natural Language Processing*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2017. doi: 10.2200/S00762ED1V01Y201703HLT037. URL https://doi.org/10.2200/S00762ED1V01Y201703HLT037.

[20] Ian J. Goodfellow, Yoshua Bengio, and Aaron C. Courville. *Deep Learning*. Adaptive computation and machine learning. MIT Press, 2016. ISBN 978-0-262-03561-3. URL http://www.deeplearningbook.org/.

[21] Noah D Goodman and Michael C Frank. Pragmatic language interpretation as probabilistic inference. *Trends in cognitive sciences*, 20(11): 818–829, 2016.

[22] H. Paul Grice. Logic and conversation. In Peter Cole and Jerry L. Morgan, editors, *Speech Acts*, volume 3 of *Syntax and Semantics*, pages 41–58. Academic Press, New York, 1975.

[23] T. Hashimoto, H. Zhang, and P. Liang. Unifying human and statistical evaluation for natural language generation. In *North American Association for Computational Linguistics (NAACL)*, 2019.

[24] Michael Heilman, Aoife Cahill, Nitin Madnani, Melissa Lopez, Matthew Mulholland, and Joel R. Tetreault. Predicting grammaticality on an ordinal scale. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 2: Short Papers*, pages 174–180, 2014. URL http://aclweb.org/anthology/P/P14/P14-2029.pdf.

[25] Ulf Hermjakob, Jonathan May, Michael Pust, and Kevin Knight. Translating a language you don't know in the chinese room. In Fei Liu and Thamar Solorio, editors, *Proceedings of ACL 2018, Melbourne, Australia, July 15-20, 2018, System Demonstrations*, pages 62–67. Association for Computational Linguistics, 2018. ISBN 978-1-948087-65-0. URL https://aclanthology.info/papers/P18-4011/p18-4011.

[26] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

[27] William J. Hutchins and Harold L. Somers. *An introduction to machine translation*. Academic Press, 1992. ISBN 978-0-12-362830-5.

[28] Kyle Johnson. University of Massachusetts, Amherst, LINGUIST 401: Introduction to Syntax, Lecture Notes: Phrase Structure Rules, 2016. URL: http://courses.umass.edu/kbj/ling401/content/401hand04_psrules.pdf. Last visited on 2018/04/29.

[29] Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda B. Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's multilingual neural machine translation system: Enabling zero-shot translation. *TACL*, 5:339–351, 2017. URL https://transacl.org/ojs/index.php/tacl/article/view/1081.

[30] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Bengio and LeCun [6]. URL http://arxiv.org/abs/1412.6980.

[31] Philip Koehn. *Statistical Machine Translation*. Cambridge University Press, 2010. ISBN 978-0-521-87415-1. URL http://www.statmt.org/book/.

[32] Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86, 2005.

[33] Philipp Koehn. Neural machine translation. *CoRR*, abs/1709.07809, 2017. URL http://arxiv.org/abs/1709.07809.

[34] Alexander Kuhnle and Ann A. Copestake. Deep learning evaluation using deep linguistic processing. *CoRR*, abs/1706.01322, 2017. URL http://arxiv.org/abs/1706.01322.

[35] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Proc. ACL workshop on Text Summarization Branches Out*, page 10, 2004. URL http://research.microsoft.com/~cyl/download/papers/WAS2004.pdf.

[36] Chin-Yew Lin. Rouge: a package for automatic evaluation of summaries. July 2004. URL https://www.microsoft.com/en-us/research/publication/rouge-a-package-for-automatic-evaluation-of-summaries/.

[37] Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. Assessing the ability of lstms to learn syntax-sensitive dependencies. *TACL*, 4:521–535, 2016. URL https://transacl.org/ojs/index.php/tacl/article/view/972.

[38] Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019.

[39] Gábor Melis, Chris Dyer, and Phil Blunsom. On the state of the art of evaluation in neural language models. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=ByJHuTgA-.

[40] Tomas Mikolov, Anoop Deoras, Stefan Kombrink, Lukás Burget, and Jan Cernocký. Empirical evaluation and combination of advanced language modeling techniques. In *INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association, Florence, Italy, August 27-31, 2011*, pages 605–608, 2011. URL http://www.isca-speech.org/archive/interspeech_2011/i11_0605.html.

[41] Graham Neubig. Neural machine translation and sequence-to-sequence models: A tutorial. *CoRR*, abs/1703.01619, 2017. URL http://arxiv.org/abs/1703.01619.

[42] Graham Neubig, Chris Dyer, Yoav Goldberg, Austin Matthews, Waleed Ammar, Antonios Anastasopoulos, Miguel Ballesteros, David Chiang, Daniel Clothiaux, Trevor Cohn, Kevin Duh, Manaal Faruqui, Cynthia Gan, Dan Garrette, Yangfeng Ji, Lingpeng Kong, Adhiguna Kuncoro, Gaurav Kumar, Chaitanya Malaviya, Paul Michel, Yusuke Oda, Matthew Richardson, Naomi Saphra, Swabha Swayamdipta, and Pengcheng Yin. Dynet: The dynamic neural network toolkit. *CoRR*, abs/1701.03980, 2017. URL http://arxiv.org/abs/1701.03980.

[43] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA.*, pages 311–318. ACL, 2002. URL http://www.aclweb.org/anthology/P02-1040.pdf.

[44] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[45] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics, 2018. ISBN 978-1-948087-27-8. URL https://aclanthology.info/papers/N18-1202/n18-1202.

[46] Carl Pollard and Ivan A. Sag. *Head-Driven Phrase Structure Grammar*. The University of Chicago Press, Chicago, 1994.

[47] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report.

[48] Christof Rapp. Aristotle's rhetoric. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, spring 2010 edition, 2010.

[49] Ehud Reiter. A structured review of the validity of BLEU. *Computational Linguistics*, 44(3), 2018. doi: 10.1162/coli_a_00322. URL https://doi.org/10.1162/coli_a_00322.

[50] Ehud Reiter and Robert Dale. Building applied natural language generation systems. *Natural Language Engineering*, 3(1):57–87, 1997. doi: 10.1017/S1351324997001502. URL https://doi.org/10.1017/S1351324997001502.

[51] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning internal representations by error propagation. In David E. Rumelhart and James L. Mcclelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*, pages 318–362. MIT Press, Cambridge, MA, 1986.

[52] Alexander M. Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 379–389, 2015. URL http://aclweb.org/anthology/D/D15/D15-1044.pdf.

[53] John R Searle. Minds, brains, and programs. *Behavioral and brain sciences*, 3 (3):417–424, 1980.

[54] Rico Sennrich. How grammatical is character-level neural machine translation? assessing MT quality with contrastive translation pairs. *CoRR*, abs/1612.04629, 2016. URL http://arxiv.org/abs/1612.04629.

[55] Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nadejde. Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain, April 2017. Association for Computational Linguistics. URL http://aclweb.org/anthology/E17-3017.

[56] Xing Shi, Inkit Padhi, and Kevin Knight. Does string-based neural MT learn source syntax? In Su et al. [60], pages 1526–1534. ISBN 978-1-945626-25-8. URL http://aclweb.org/anthology/D/D16/D16-1159.pdf.

[57] Hava T. Siegelmann. Recurrent neural networks and finite automata. *Computational Intelligence*, 12:567–574, 1996. doi: 10.1111/j.1467-8640.1996.tb00277.x. URL https://doi.org/10.1111/j.1467-8640.1996.tb00277.x.

[58] Noah A. Smith. *Linguistic Structure Prediction*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2011. doi: 10.2200/S00361ED1V01Y201105HLT013. URL https://doi.org/10.2200/S00361ED1V01Y201105HLT013.

[59] Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. A neural network approach to context-sensitive generation of conversational responses. In Rada Mihalcea, Joyce Yue Chai, and Anoop Sarkar, editors, *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 196–205. The Association for Computational Linguistics, 2015. ISBN 978-1-941643-49-5. URL http://aclweb.org/anthology/N/N15/N15-1020.pdf.

[60] Jian Su, Xavier Carreras, and Kevin Duh, editors. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, 2016. The Association for Computational Linguistics. ISBN 978-1-945626-25-8. URL http://aclweb.org/anthology/D/D16/.

[61] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112, 2014. URL http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.

[62] A. M. Turing. Computing machinery and intelligence. *Mind*, 59(236): 433–460, 1950. ISSN 00264423. URL http://www.jstor.org/stable/2251299.

[63] Oriol Vinyals and Quoc V. Le. A neural conversational model. *CoRR*, abs/1506.05869, 2015. URL http://arxiv.org/abs/1506.05869.

[64] Qingyun Wang, Xiaoman Pan, Lifu Huang, Boliang Zhang, Zhiying Jiang, Heng Ji, and Kevin Knight. Describing a knowledge base. In Emiel Krahmer, Albert Gatt, and Martijn Goudbeek, editors, *Proceedings of the 11th International Conference on Natural Language Generation, Tilburg University, The Netherlands, November 5-8, 2018*, pages 10–21. Association for Computational Linguistics, 2018. ISBN 978-1-948087-86-5. URL https://aclanthology.info/papers/W18-6502/w18-6502.

[65] Johnny Wei, Khiem Pham, Brendan O'Connor, and Brian Dillon. Evaluating grammaticality in seq2seq models with a broad coverage HPSG grammar: A case study on machine translation. In Tal Linzen, Grzegorz Chrupala, and Afra Alishahi, editors, *Proceedings of the Workshop: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2018, Brussels, Belgium, November 1, 2018*, pages 298–305. Association for Computational Linguistics, 2018. ISBN 978-1-948087-71-1. URL https://aclanthology.info/papers/W18-5432/w18-5432.

[66] Gail Weiss, Yoav Goldberg, and Eran Yahav. On the practical computational power of finite precision rnns for language recognition. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 740–745. Association for Computational Linguistics, 2018. ISBN 978-1-948087-34-6. URL https://aclanthology.info/papers/P18-2117/p18-2117.

[67] Gail Weiss, Yoav Goldberg, and Eran Yahav. Extracting automata from recurrent neural networks using queries and counterexamples. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pages 5244–5253, 2018. URL http://proceedings.mlr.press/v80/weiss18a.html.

[68] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff

Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016. URL http://arxiv.org/abs/1609.08144.

[69] Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay S. Pande. Moleculenet: A benchmark for molecular machine learning. *CoRR*, abs/1703.00564, 2017. URL http://arxiv.org/abs/1703.00564.

[70] Lili Yao, Nanyun Peng, Ralph M. Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. Plan-and-write: Towards better automatic storytelling. *CoRR*, abs/1811.05701, 2018. URL http://arxiv.org/abs/1811.05701.

Colophon

This thesis was typeset using LaTeX, originally developed by Leslie Lamport and based on Donald Knuth's TeX. The body text is set in 11 point Arno Pro, designed by Robert Slimbach in the style of book types from the Aldine Press in Venice, and issued by Adobe in 2007. A template, which can be used to format a PhD thesis with this look and feel, has been released under the permissive MIT (X11) license, and can be found online at github.com/suchow/ or from the author at suchow@post.harvard.edu.