# Impossibility of Unambiguous Communication as a Source of Failure in AI Systems

**William J. Howe**[1] , **Roman V. Yampolskiy**[2]
[1]Johns Hopkins University
[2]University of Louisville
whowe1@jhu.edu, roman.yampolskiy@louisville.edu

## Abstract

Ambiguity is pervasive at multiple levels of linguistic analysis effectively making unambiguous communication impossible. As a consequence, natural language processing systems without true natural language understanding can be easily "fooled" by ambiguity, but crucially, AI also may use ambiguity to fool its users. Ambiguity impedes communication among humans, and thus also has the potential to be a source of failure in AI systems.

[1]

## 1 Introduction

The human language faculty allows any given speaker to "make infinite use of finite means" [Chomsky, 2006]. This is to say that the set of all possible sentences is infinite while the set of words which make them up is finite. However, ambiguity – the existence of more than one interpretation of an expression, is rampant in natural language [Wasow *et al.*, 2005]. It is not clear why ambiguity exists at all in natural language. Given that it impedes communication, one might assume languages would evolve to avoid it, yet this is not observed [Wasow *et al.*, 2005]. One explanation is that mapping a word to multiple meanings saves memory. Another account asserts that ambiguity is a consequence of a human bias toward shorter morphemes [Wasow *et al.*, 2005]. Yet another account construes ambiguity as a product of optimization towards efficiency (principle of least-effort) over the course of language evolution. On this view, ambiguity is the price paid for a least effort language [Solé and Seoane, 2015]. In this paper, we won't seek to explain the root cause of ambiguity, but rather to show how it can pose a problem for AI systems. First we'll identify types of ambiguity which occur at the levels of phonology, syntax, and semantics, noting how modern natural language processing (NLP) systems disambiguate ambiguous input. Finally, we'll consider how more advanced AI could exploit ambiguity and how bad actors might utilize such systems to their ends.



## 2 Phonology

Computational phonology is a core component of speech-based NLP systems. The ultimate goal of automatic speech recognition is to take an acoustic waveform as input and decode it into a string of words as text [Jurafsky, 2000]. The field which for several years was dominated by the Gaussian Mixture Model - Hidden Markov Model (GMM-HMM) framework has now made significant advancements using deep neural network (DNN) architectures to enable technologies like Siri, Alexa, and Google Assistant [Yu and Deng, 2016]. In particular recurrent neural networks which capture the "dynamic temporal behavior" of sequence data that DNN-HMM architectures do not capture, have proven very effective [Yu and Deng, 2016]. Despite these advances, automatic speech recognition (ASR) still performs poorly with far field microphones, noisy conditions, accented speech, and multitalker speech [Yu and Deng, 2016]. To see why ambiguity poses such a problem for these models, we'll consider a architecture which uses some statistical technique to recognize speech units along with some language model over some dictionary to find the highest probability sequence of speech units [Jurafsky, 2000]. It is clear that because such a model is probabilistic, it lacks true natural language understanding – this means the model can fail when faced with a speech waveform that might be *unlikely* or *low probability*. It may favor the more likely incorrect output over the less likely yet correct target output. Because humans possess linguistic creativity – the ability to produce never before seen utterances which a model might consider highly improbable, current ASR systems have an inherent deficit. One way to remedy this is to filter out hypotheses that don't make sense with, "[a] speech recognition system augmented with Commonsense Knowledge [that] can spot its own nonsensical errors, and proactively correct them" [Lieberman *et al.*, 2005; Liu *et al.*, 2016]. Nevertheless, brittle ASR systems, "may misinterpret commands due to coarticulation, segmentation, homophones, or double meanings in the human language" [Yampolskiy, 2016].

### 2.1 Homophones

Homophones – sets of words which sound the same but have different meanings, are a classic case of phonological ambiguity. The following data present utterances which could be misinterpreted by an ASR system but which are easily disam-

biguated by humans provided some context [Forsberg, 2003].

(1) a. the tail of the dog
b. the tale of the dog

(2) a. the sail of the boat
b. the sale of the boat

The pairs in (1) and (2) are phonologically and syntactically identical, yet convey distinct meanings. With sufficient context, an ASR with a good enough language model would easily disambiguate *tail/tale* shown below:

(3) a. the tail of the dog **was wagging**
b. the tale of the dog **was told**
c. the tail/tale of the dog **was long**

However, sufficient context is not always provided as shown in c). Thus, carefully chosen homophones could be used to intentionally *fool* an ASR system.

### 2.2 Continuous Speech

Continuous speech is very different from written language. Spoken language introduces word boundary ambiguity, speaker variability, and a different vocabulary. It is common for the spoken register of a language to be different from the more formal written register [Forsberg, 2003]. All these factors contribute to the difficulty of ASR and introduce the possibility of ambiguity when decoding.

(4) a. How to wreck a nice beach you sing calm incense.
b. How to recognize speech using common sense.

(5) a. I want to experience youth in Asia.
b. I want to experience euthanasia.

These constructed yet plausible examples show that it may be possible to generate adversarial examples to fool an ASR system. Once again, one would expect an effective language model to be successful at disambiguating these examples, but, as we will note below, there is evidence that fooling an ASR system is even more easily achieved by simply perturbing the input waveform.

### 2.3 Fooling Automatic Speech Recognition Systems

We've shown that it ought to be possible to fool ASR systems using phonological ambiguities which are common in natural language. This involves carefully crafting utterances with homophones or with word boundary ambiguity. However, it is possible to exploit such systems without leveraging natural language ambiguity. It has been shown that adversarial examples can be created by applying perturbations to an input waveform such that the waveform is nearly indistinguishable from the unperturbed input. Even more worrying is fact that this perturbed input can generate any desired output phrase [Carlini and Wagner, 2018]. The researchers have also shown that *hidden voice commands*, unintelligible inputs used to attack ASR systems, can be used to cause denial of service, information leakage, and "as a stepping stone to further attacks" [Carlini *et al.*, 2016]. Though these exploits don't necessarily target natural language ambiguity they serve to show that current ASR systems are vulnerable to a range of attacks.

## 3 Syntax

Syntax determines how words are organized into phrases and sentences [Carnie, 2012]. Historically, syntax has been processed with computational models including context-free grammars, lexicalized grammars, feature structures, parsing algorithms, and HMM part-of-speech taggers. Parsing a sentence into constituency or dependency tree structure is useful for downstream NLP tasks. The same is true for part-of-speech tagging [Jurafsky, 2000]. Deep learning using ANNs has achieved state of the art performance on syntax-related tasks, though ANNs still do not match human level performance on phenomena like filler-gap dependencies [Linzen and Baroni, 2020]. Here, we'll examine several characterizations of ambiguity at the level of syntax.

### 3.1 Structural Ambiguity

Structural ambiguity occurs when more than one underlying structure exists for a single sentence with the structures having different meanings. The term *structure* is used here because sentences with this type of ambiguity are usually disambiguated by distinguishing between two different constituency trees.

**Global Ambiguity**

Global ambiguity is ambiguity that persists after a sentence has been fully parsed. In this case a sentence in and of itself contains more than one structural interpretation. Consider the following data:

(6) a. Eliminate [NP the target] [PP with a bomb.]
b. Eliminate [NP the target [PP with a bomb.]]

Here, the NP (noun phrase) has two interpretations; one where the PP (prepositional phrase) is contained within the NP and the other where it is not. The former refers to an individual carrying a bomb, while the latter refers to the action of bombing someone.

**Local Ambiguity**

Local ambiguity, unlike global ambiguity, is resolved upon complete parsing of a sentence. The canonical case of local ambiguity is the garden path sentence. Consider the following data [Ferreira and Henderson, 1991]:

(7) a. Because Bill drinks wine ...
b. Because Bill drinks wine beer is never kept in the house.
c. Because Bill drinks wine is never kept in the house.

As seen in (7a), *Because Bill drinks wine* is ambiguous: wine could take on a direct object semantic role as in (7b) or it could take on a subject semantic role as in (7c). Assuming that a human parser employs the principle of late closure, preferring to attach new material into the phrase or clause

currently open rather than create new clauses or constituents, (7b) is easier to parse for a human than (7c). In general for humans, garden-path recovery is thought to involve reanalysis of the sentence by reassigning the thematic roles of a misanalyzed phrase [Ferreira and Henderson, 1991]. Regardless, the ambiguity and added parsing difficulty of garden-path sentences could be a source of failure in NLP and AI systems.

### 3.2 Formal Language Ambiguity

A context-free grammar is a grammar whose rules all follow the form $A \rightarrow \Psi$ where A is a non-terminal symbol and $\Psi$ is any string, even the empty string, from the union of the terminal and non-terminal alphabets [Partee *et al.*, 2012]. Consider the following context-free grammar:

$$S \rightarrow (A\ B) \mid (C\ D)$$
$$A \rightarrow U \quad C \rightarrow U$$
$$B \rightarrow V \quad D \rightarrow V$$
$$U \rightarrow a \quad V \rightarrow b$$

Even in this simple context-free grammar, the string *ab* can either be generated using the rule $S \rightarrow A\ B$ or the rule $S \rightarrow C\ D$. Thus, there exists more than one parse tree structure for the same surface string representation and this constitutes one characterization of syntactic ambiguity. One notable technique for disambiguating context-free grammars is the PCFG (probabilistic context-free grammar) which assigns probabilities to rules in a CFG (different weights for the two S rules above, for example) [Jurafsky, 2000]. As in the discussion on phonology, probabilistic language models may serve to make NLP systems more "natural" (more similar to human language) yet this may not give models the capability to reason about more complex ambiguities. The same applies for neural network based language models such as GPT-2 [Radford *et al.*, 2019] and BERT [Devlin *et al.*, 2018] which can be thought of as massive context-free grammars with extremely well fine-tuned probabilistic weights.

### 3.3 Security of Language Models

In addition to the advances made in downstream NLP tasks by means of language model pretraining and fine-tuning, recently, neural network language models have been shown to perform well as knowledge bases [Petroni *et al.*, 2019]. Specifically, BERT (Bidirectional Encoder Representations from Transformers) has been shown to contain relational knowledge competitive with traditional knowledge base methods and to perform well on open-domain question answering [Petroni *et al.*, 2019]. If neural network language models become widely adopted as knowledge bases, this necessitates the question, *Is the private information encoded in a language model secure?* Though it does not relate to ambiguity, there is work showing that privacy can be preserved in such models using encryption [Ryffel *et al.*, 2018].

## 4 Semantics

Semantics, the meaning of words and sentences, is of considerable interest in NLP. However, much of the perceived semantic knowledge encoded in current NLP systems is instead derived from the use of syntactic heuristics which quickly break down when confronted with more complex examples [McCoy *et al.*, 2019]. This is a major problem for narrow AI. For advanced AI, the ability to toy with the very meaning of language would have wide ranging consequences from sowing disinformation to generating ambiguous legal documents or tweets.

### 4.1 Scope Ambiguity

Here, the *scope* of a syntactic constituent is ambiguous. The following data further elucidates scopal ambiguity [Wasow *et al.*, 2005]:

(8) No student solved exactly two problems.

In (8) either, "there was no student who solved exactly two problems", or "there were exactly two problems that no student solved" [Wasow *et al.*, 2005]. Either interpretation is valid depending on the location of constituents in the underlying sentence structure which determines their scope (this is sometimes referred to as LF, logical form). For this reason, scope ambiguity lies at the syntax-semantics interface [Anderson, 2004].

### 4.2 Lexical Ambiguity

This type of ambiguity deals with the meanings of words. When a word has more than one distinct meaning it is said to have lexical ambiguity. We'll highlight examples of lexical ambiguity and examine current NLP approaches to addressing it.

**Contranyms**

In the case of contranyms, a word has two different meanings which are antonyms [Jackson, 2018]:

(9) *hold up*
- a. to support
- b. to hinder

(10) *dust*
- a. add fine particles
- b. remove fine particles

(11) *left*
- a. departed
- b. remaining

**Word Sense Disambiguation**

The most salient case of lexical ambiguity is known as *polysemy* in which one word has more than one distinct meaning. The work *bank* can refer to a bank account, to a river bank, or as a verb, to moving on an incline. There is a long history of computational techniques for word sense disambiguation from dictionary based methods to semantic similarity metrics [Yarowsky, 1995; Banerjee and Pedersen, 2002; Navigli, 2009; Resnik, 1999].

### 4.3 Winograd Schema

A winograd schema is a pair of sentences that differ in only two words and contain a referential ambiguity that is resolved in "opposite directions" in the two sentences. The Winograd

Schema Challenge presents such a pair as an alternative to the Turing Test since a successful agent must have some level of natural language understanding to solve the challenge and cannot depend on statistical patterns [Levesque *et al.*, 2012]. Though the Winograd Schema is technically a referential ambiguity, its difficulty is rooted in machines' lack of commonsense knowledge so we've placed it in the semantics section.

(12) The trophy doesn't fit in the brown suitcase because it's too (big/small). What is too (big/small)?
   a. the trophy
   b. the suitcase

(13) Joan made sure to thank Susan for all the help she had (given/received). Who had (given/received) the help?
   a. Joan
   b. Susan

In (12) there are two sentences that can be generated based on the choice of *big* or *small* which have two different answers. Answering correctly requires natural language understanding and reasoning. The dataset WINOGRANDE showed that although models performed well (90% accuracy) on existing Winograd datasets, this was likely due to algorithmic bias. Producing adversarial Winograd Schema examples by means of a debiasing algorithm allowed the authors to achieve state of the art performance on these existing Winograd benchmarks showing their technique to be a powerful example of transfer learning [Sakaguchi *et al.*, 2019].

## 5 Criticism of Deep Learning Approaches

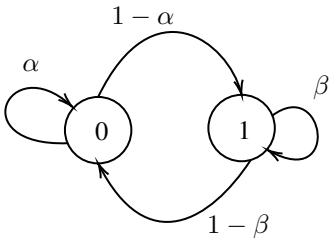
The approaches for dealing with natural language and thus in turn ambiguity discussed above are largely engineering approaches. These include things like, deep learning, fine-tuning of language models, building more robust models that generalize better, and improving state of the art performance using adversarial examples [Jia and Liang, 2017; Subramanian *et al.*, 2017; Wu *et al.*, 2018; Wallace *et al.*, 2019; Gong *et al.*, 2018]. Although these engineering approaches have achieved state of the art performance in many areas, there is a sense that they lack true natural language understanding as alluded to in several of the above types of ambiguity. [Marcus, 2020] describes deep learning based models as, "data hungry, shallow, brittle, and limited in their ability to generalize" advocating instead for *symbolic* approaches incorporating insights from cognitive science. Character based translation models break down under the introduction of noise (letter swap errors) proving such NMT (neural machine translation) models to be extremely brittle [Belinkov and Bisk, 2017]. [Bender and Koller, 2020] argues that current approaches cannot learn *form from meaning* and thus will not achieve natural language understanding (NLU). It has been noted even that algorithmic approaches to anaphor resolution may never achieve complete success since, in principle the interpretation of a well-formed sentence like, *Who wants the first one?* is *free* in the absence of sufficient context and application of constraints (i.e *one* could refer to anything) [Hendriks and De Hoop, 2001]. Advocates of deep learning approaches nonetheless can taut that it is *their* models which have achieved such success on natural language tasks and benchmarks. The debate between deep learning approaches and symbolic approaches is not yet resolved. An interesting area is neural-symbolic computation which seeks to marry neural network models with symbolic approaches [Smolensky *et al.*, 2016; Garcez *et al.*, 2015].

## 6 Inevitability of Ambiguity

Pragmatics, the area of linguistics which focuses on the co-operative assumptions of communication, arguably bears its own ambiguities such as irony and sarcasm [Wilson, 2006]. There are existing computational approaches for dealing with various other discourse ambiguities [Macagno and Bigi, 2018; Ammicht *et al.*, 2001]. On the basis of discourse analysis [Blum-Kulka and Weizman, 1988] argue that "communication is inherently ambiguous". Intuitively, we endorse this view since wholly unambiguous communication seems to be impossible using an inherently ambiguous natural language. On the basis of the above examples, it is clear that there are some ambiguities that even humans cannot easily disambiguate. These effects are only multiplied when one considers the ambiguity in pragmatics which might cause one to "question the validity of co-operative assumptions" such as the Gricean maxims [Blum-Kulka and Weizman, 1988]. Here we'll seek to discuss this claim with more mathematical rigor.

We'll represent a discourse with a finite-state discrete time Markov Chain with two states (*Figure 1*). The chain is in state 0 when ambiguity is introduced into a discourse and in state 1 when there is no natural language ambiguity. This two state chain is a positive recurrent irreducible Markov Chain [Ross *et al.*, 1996]. Each new utterance in a discourse is represented by a transition in the chain. This chain is a good model since a conversation can stay in the unambiguous state with positive probability. The conversation can move from a unambiguous state to an ambiguous state with positive probability – this is when ambiguity is introduced into the discourse. The conversation can remain ambiguous (with positive probability) by what [Blum-Kulka and Weizman, 1988] cite as *indeterminate ambiguity* in which "the speaker does not commit himself to an intended meaning" and the indeterminacy is "left unattended to by both participants". The ambiguity can also be resolved (with positive probability) through clarification. All entries in the transitive matrix are positive probabilities. Thus, for any finite discourse there is positive probability that there is no ambiguity. This is achieved by simply taking the transition from state 1 to state 1 at every point in the conversation.

However, if we consider an infinite number of transitions which may be a good model for a work of fiction, a long speech, or an extended conversation over hours or days, in the limit, the chain will enter state 0 (the ambiguity state) with probability 1 and return to the state in a finite number of transitions on average [Ross *et al.*, 1996]. This follows from the fact that the chain is irreducible and positive recurrent. If we then consider the totality of human natural language generation as an infinite sequence of transitions through the chain, it is clear that ambiguity is inevitable as long as we en-




$$P = \begin{bmatrix} \alpha & 1-\alpha \\ 1-\beta & \beta \end{bmatrix}$$

Figure 1: State 0 denotes ambiguous discourse while state 1 denotes unambiguous discourse. P is the 2 state transition matrix [Ross *et al.*, 1996]

.

dorse the premise that there is positive probability of generating ambiguous language at any step in the sequence. In order to achieve unambiguous communication one would have to ensure that at any step in this infinite sequence there is zero probability of generating an ambiguity – we claim that this is computationally intractable if one is using an inherently ambiguous natural language.

This result, the impossibility of unambiguous communication, is in accordance with two existing impossibility results in AI safety. The first, *unpredictability of AI*, states that, " it is impossible to precisely and consistently predict what specific actions a smarter-than-human intelligent system will take to achieve its objectives" [Yampolskiy, 2019b]. The second result, *incomprehensibility of AI*, shows that "advanced AIs would not be able to accurately explain some of their decisions and for the decisions they could explain people would not understand some of those explanations" [Yampolskiy, 2019a]. This is true for opaque, black box NLP systems discussed in Section 7.1. [Doran *et al.*, 2017]. However, we'll show in Section 7.2 that the impossibility of unambiguous communication also contributes to unexplainability and unpredictability for advanced AI .

## 7 Ambiguity as a Source of Failure

Pioneer in the field of machine translation, Yehoshua Bar-Hillel claimed "FAHQT [fully automatic high quality translation] is out of the question for the foreseeable future because of the existence of a large number of sentences the determination of whose meaning, unambiguous for a human reader, is beyond the reach of machines" citing machines' lack of commonsense knowledge [Bar-hillel, 1964]. While more advanced NLP systems may achieve higher accuracy on ambiguous language tasks, the main thrust of his argument still stands, even today. On top of that, there are examples of ambiguity given above that, without sufficient context, *even humans* cannot correctly interpret. Thus, it is clear that these weaknesses inherent in natural language could be exploited to fool an NLP system. Additionally, the reverse could be possible – AI could exploit natural language ambiguity to fool its human users.

### 7.1 AI Fooled by Ambiguity

AI without true natural language understanding can easily be tricked by many of the above examples. Homophones and continuous speech could be used to give a command with undesired effects. A benign waveform command could be perturbed to cause system failure, and open the door to further hacks and exploits. An input command like, *Eliminate the target with the bomb.* could have unintended consequences depending on the interpretation of its structural ambiguity which could be particularly dangerous in military and AI weaponry scenarios. Clearly, serious thought should be put into designing systems that intelligently deal with ambiguous input. Just as Google auto-completes search results, an AI might be designed to answer and react to a query as quickly as possible – this could result in failure on garden path sentences. The AI might be tricked into the wrong interpretation by using a *late closure* parsing technique.

Although machine translation is largely dominated by sequence to sequence methods, a dictionary based translation system could fail due to cross-linguistic ambiguity. In a case reminiscent of the movie *Arrival*, there is a character in Chinese which can mean *instrument* or *weapon* in English.[2] A mistake in translation could be dangerous in this case. There are plenty of other cases of lexical ambiguity that could present a dangerous situation, particularly in a military context. The word *execute* is a contranym – it can refer to the execution (start) of a program ("execute the firmware update"), or the execution (end) of a person ("execute the adversary")[3]. These failure modes are a result of the opaque, black box nature of contemporary narrow AI systems without natural language understanding. The ambiguity induced failure modes discussed here are examples of *by mistake, post-deployment* AI hazards according to Yampolskiy's taxonomy [Yampolskiy, 2016].

### 7.2 AI Exploiting Ambiguity

Ambiguity has been identified as a source of miscommunication in air traffic control [Mcmillan, 1998]. Ambiguity is such a problem even for humans that some have attempted to build controlled natural languages by restricting language use to a wholly unambiguous subset of an existing natural language

[2] According to Google Translate.

[3] A reviewer notes that the event predicate *execute* is not strictly synonymous with a *start* event predicate and thus this example does not constitute a true contranym. However, since performing some action entails *starting* to perform the action, the example still makes sense and is thus useful for explanatory purposes

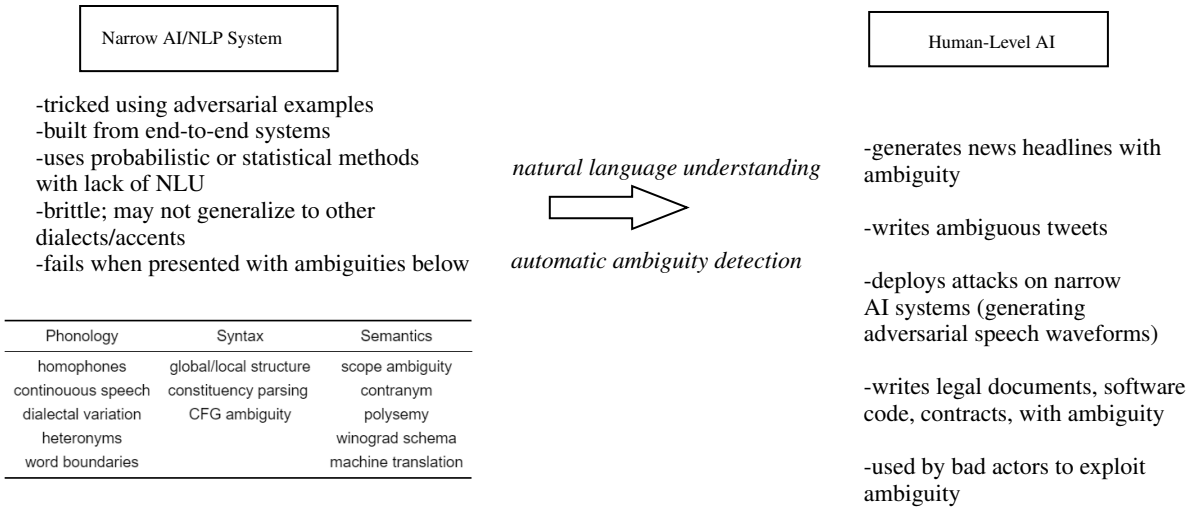



Figure 2: A narrow AI system is vulnerable to attack and may fail on ambiguous input. A more advanced system is then able to exploit ambiguous language to generate misleading headlines or tweets.

[Fuchs *et al.*, 2008]. Such a language could be used to enable precise and unambiguous specification of rules and guidelines for organizations and software specifications. There are existing attempts to detect ambiguity in requirements specifications for software [Kiyavitskaya *et al.*, 2008]. An AI capable of exploiting ambiguity could reek havoc in these areas.

In monetary theory, the observation that, "Any observed statistical regularity will tend to collapse once pressure is placed upon it for control purposes" has been termed *Goodhart's Law* [Goodhart, 1983]. The observation has been restated as: "When a measure becomes a target, it ceases to be a good measure" [Hoskin, 1996]. Without making any claims about the monetary theories underlying these observations, we present a generalization of Goodhart's Law for AI systems: *A tool for recognizing ambiguity in natural language, once applied to a sufficiently intelligent AI will cease to be effective and could be exploited.* This is to say that any attempt to recognize ambiguity as in [Sproat and Santen, 1998; Chantree *et al.*, 2006] could be used by that AI to create a dataset which upon self-training will allow the AI to generate ambiguous language. This would allow the AI to write legal documents, software, news headlines, and tweets riddled with ambiguity.[4] This could have far reaching implications, including the potential for widespread disinformation campaigns and the disruption of systems discussed in the previous paragraph. Based on an existing categorization of Goodhart's Law variants by [Manheim and Garrabrant, 2018], the dynamic illustrated here may well be considered an "adversarial misalignment Goodhart [variant]" in which "The agent applies selection pressure knowing the regulator will apply different selection pressure on the basis of the metric". In this case, a bad actor, or the AI itself (subject to a utility function perhaps) may use ambiguity detection not to notify users of the ambiguity, but to generate ambiguities, sparking public confusion and disinformation. Here, the metric (ambiguity detection module) is used to "distort and corrupt the social processes it is intended to monitor" [Manheim and Garrabrant, 2018]. The unpredictability of such a system is a major risk. According to Yampolskiy's *Taxonomy of Pathways to Dangerous AI*, the use of such a system by a bad actor is an *on purpose, post-deployment* AI hazard [Yampolskiy, 2016].

### 7.3 Safety Risk

Natural language ambiguity appears to introduce risk into three types of systems. First, there are the uninterpretable NLP systems that largely dominate today's state-of-the-art NLP leaderboards. These models may be trained end-to-end on a specific narrow task like speech recognition. Due to the brittleness of these models, the inevitable errors they make on ambiguous input as a result of their normal operation constitute a safety risk in critical systems. In a poorly designed voice activated self-driving car you might give the command, "Drive me up to Oxridge" which could be erroneously recognized as "Drive me off a bridge." Of course this example is only a safety risk if the natural language command is able to override the car's control system which is programmed to drive only on drive-able space.

Secondly, there are interpretable NLP systems which nonetheless lack human-level AI capability and which in turn can pose an AI safety risk. Even if a system is able to identify an interpretation of an ambiguity as more plausible (either statistically or using rule-based knowledge) there is no guarantee that in a given situation the maximum likelihood solution is the correct one. For example, communication is increasingly being augmented with AI generated smart reply suggestions. It could be the case that two business partners routinely agree on contracts such that the AI's training data is strongly biased towards replying *yes* to new contracts between them. Eventually though, there will be a contract which one partner doesn't agree on – but if the business is us-

[4]Interestingly, an advanced AI would also be capable of attacking narrow AI systems with some of the exploits on the left hand side of the *Figure 2* such as the hidden voice command exploit.

ing an AI bot or AI negotiator, it will accept this bad contract due to its training data.

Finally there is the risk for human-level AI which can manufacture ambiguous legal reports, contracts, or software requirements to achieve unforseen objectives.

### 7.4 Potential Solutions

For uninterpretable AI systems, there simply is no good way to deploy these systems for critical applications without a way to quantify safety risk. Taking self-driving cars as an example, it is certainly possible to measure safety by tracking human driver disengagements of the AI control system. The system can be deemed safe when these engagements are statistically unlikely. For AI systems where performance is not so easily quantified, there can be no safety guarantees. Even then, a *statistically safe* system could still fail due to low probability events not captured in the system's training data. It is not as obvious how statistical safety could be established for an NLP system.

For interpretable systems, human-in-the-loop and human-AI teaming may solve ambiguity related AI risk provided that an AI system can quantify its uncertainty about a prediction or detect ambiguous input to the system. Then in the case of ambiguity, the system can simply offload its decision making to the human.

One useful tool for natural language ambiguity might be come from the idea of *prefix codes* which convert an alphabet into a unique set of binary "codes" which can be concatenated to form messages. These messages are "uniquely decoded" such that there is no ambiguity [Le Boudec *et al.*, 2015]. This type of unambiguous communication might be usefully extended into NLP systems in order to achieve unambiguous decoding, though it would require a more complex coding scheme.

## 8 Conclusion

We've detailed a non-exhaustive set of cases of ambiguity in natural language at the levels of phonology, syntax, semantics, and pragmatics along with contemporary NLP and AI approaches to handling them. These areas taken together give of good sense of the breadth of natural language ambiguity. Furthermore, we claim that ambiguous communication is inevitable when using an inherently ambiguous communication system. Despite the success of techniques including neural networks, deep learning, adversarial examples, and better language models (these are not mutually exclusive), many worry that end-to-end systems only learn "surface" representations and don't have true natural language understanding. We noted several areas where natural language ambiguity might lead to a failure mode in AI systems with deleterious effects. The security risk is twofold – one set of concerns exists for narrow AI systems while a different set exists for human-level AI. Finally, we formulated a generalized Goodhart's Law to express the idea that techniques allowing AI to identify, recognize, and detect ambiguity, might be *reversible* in that they could be exploited to generate ambiguous language and "fool" their human users.

## References


[Ammicht *et al.*, 2001] Egbert Ammicht, Alexandros Potamianos, and Eric Fosler-Lussier. Ambiguity representation and resolution in spoken dialogue systems. In *Seventh European Conference on Speech Communication and Technology*, 2001.

[Anderson, 2004] Catherine Anderson. *The structure and real-time comprehension of quantifier scope ambiguity*. PhD thesis, Northwestern University Evanston, IL, 2004.

[Banerjee and Pedersen, 2002] Satanjeev Banerjee and Ted Pedersen. An adapted lesk algorithm for word sense disambiguation using wordnet. In *International conference on intelligent text processing and computational linguistics*, pages 136–145. Springer, 2002.

[Bar-hillel, 1964] Yehoshua Bar-hillel. A demonstration of the nonfeasibility of fully automatic high quality machine translation, 1964.

[Belinkov and Bisk, 2017] Yonatan Belinkov and Yonatan Bisk. Synthetic and natural noise both break neural machine translation. *arXiv preprint arXiv:1711.02173*, 2017.

[Bender and Koller, 2020] Emily M Bender and Alexander Koller. Climbing towards nlu: On meaning, form, and understanding in the age of data. In *Proc. of ACL*, 2020.

[Blum-Kulka and Weizman, 1988] Shoshana Blum-Kulka and Elda Weizman. The inevitability of misunderstandings: Discourse ambiguities. *Text-Interdisciplinary Journal for the Study of Discourse*, 8(3):219–242, 1988.

[Carlini and Wagner, 2018] Nicholas Carlini and David Wagner. Audio adversarial examples: Targeted attacks on speech-to-text. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 1–7. IEEE, 2018.

[Carlini *et al.*, 2016] Nicholas Carlini, Pratyush Mishra, Tavish Vaidya, Yuankai Zhang, Micah Sherr, Clay Shields, David Wagner, and Wenchao Zhou. Hidden voice commands. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*, pages 513–530, 2016.

[Carnie, 2012] Andrew Carnie. *Syntax: A generative introduction*, volume 18. John Wiley & Sons, 2012.

[Chantree *et al.*, 2006] Francis Chantree, Bashar Nuseibeh, Anne De Roeck, and Alistair Willis. Identifying nocuous ambiguities in natural language requirements. In *14th IEEE International Requirements Engineering Conference (RE'06)*, pages 59–68. IEEE, 2006.

[Chomsky, 2006] Noam Chomsky. *Language and mind*. Cambridge University Press, 2006.

[Devlin *et al.*, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[Doran *et al.*, 2017] Derek Doran, Sarah Schulz, and Tarek R Besold. What does explainable ai really mean? a new conceptualization of perspectives. *arXiv preprint arXiv:1710.00794*, 2017.

[Ferreira and Henderson, 1991] Fernanda Ferreira and John M Henderson. Recovery from misanalyses of garden-path sentences. *Journal of Memory and Language*, 30(6):725–745, 1991.

[Forsberg, 2003] Markus Forsberg. Why is speech recognition difficult. *Chalmers University of Technology*, 2003.

[Fuchs *et al.*, 2008] Norbert E Fuchs, Kaarel Kaljurand, and Tobias Kuhn. Attempto controlled english for knowledge representation. In *Reasoning Web*, pages 104–124. Springer, 2008.

[Garcez *et al.*, 2015] Artur d'Avila Garcez, Tarek R Besold, Luc De Raedt, Peter Földiak, Pascal Hitzler, Thomas Icard, Kai-Uwe Kühnberger, Luis C Lamb, Risto Miikkulainen, and Daniel L Silver. Neural-symbolic learning and reasoning: contributions and challenges. In *2015 AAAI Spring Symposium Series*, 2015.

[Gong *et al.*, 2018] Zhitao Gong, Wenlu Wang, Bo Li, Dawn Song, and Wei-Shinn Ku. Adversarial texts with gradient methods. *arXiv preprint arXiv:1801.07175*, 2018.

[Goodhart, 1983] Charles Albert Eric Goodhart. *Monetary theory and practice: The UK-experience*. Macmillan International Higher Education, 1983.

[Hendriks and De Hoop, 2001] Petra Hendriks and Helen De Hoop. Optimality theoretic semantics. *Linguistics and philosophy*, 24(1):1–32, 2001.

[Hoskin, 1996] Keith Hoskin. The 'awful idea of accountability': inscribing people into the measurement of objects. *Accountability: Power, ethos and the technologies of managing*, 265, 1996.

[Jackson, 2018] Elizabeth Jackson. Words of many meanings. *Schwa: Language and Linguistics*, page 1, 2018.

[Jia and Liang, 2017] Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, 2017.

[Jurafsky, 2000] Dan Jurafsky. *Speech & language processing*. Pearson Education India, 2000.

[Kiyavitskaya *et al.*, 2008] Nadzeya Kiyavitskaya, Nicola Zeni, Luisa Mich, and Daniel M Berry. Requirements for tools for ambiguity identification and measurement in natural language requirements specifications. *Requirements engineering*, 13(3):207–239, 2008.

[Le Boudec *et al.*, 2015] Jean-Yves Le Boudec, Patrick Thiran, and Rüdiger Urbanke. *Introduction aux sciences de l'information: entropie, compression, chiffrement et correction d'erreurs*. PPUR Presses polytechniques, 2015.

[Levesque *et al.*, 2012] Hector Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*, 2012.

[Lieberman *et al.*, 2005] Henry Lieberman, Alexander Faaborg, Waseem Daher, and José Espinosa. How to wreck a nice beach you sing calm incense. In *Proceedings of the 10th international conference on Intelligent user interfaces*, pages 278–280, 2005.

[Linzen and Baroni, 2020] Tal Linzen and Marco Baroni. Syntactic structure from deep learning. *arXiv preprint arXiv:2004.10827*, 2020.

[Liu *et al.*, 2016] Quan Liu, Hui Jiang, Zhen-Hua Ling, Xiaodan Zhu, Si Wei, and Yu Hu. Commonsense knowledge enhanced embeddings for solving pronoun disambiguation problems in winograd schema challenge. *arXiv preprint arXiv:1611.04146*, 2016.

[Macagno and Bigi, 2018] Fabrizio Macagno and Sarah Bigi. Types of dialogue and pragmatic ambiguity. In *Argumentation and Language—Linguistic, Cognitive and Discursive Explorations*, pages 191–218. Springer, 2018.

[Manheim and Garrabrant, 2018] David Manheim and Scott Garrabrant. Categorizing variants of goodhart's law. *CoRR*, abs/1803.04585, 2018.

[Marcus, 2020] Gary Marcus. The next decade in ai: four steps towards robust artificial intelligence. *arXiv preprint arXiv:2002.06177*, 2020.

[McCoy *et al.*, 2019] Tom McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, 2019.

[Mcmillan, 1998] David Mcmillan. "…say again?…" miscommunications in air traffic control, 1998.

[Navigli, 2009] Roberto Navigli. Word sense disambiguation: A survey. *ACM computing surveys (CSUR)*, 41(2):1–69, 2009.

[Partee *et al.*, 2012] Barbara BH Partee, Alice G ter Meulen, and Robert Wall. *Mathematical methods in linguistics*, volume 30. Springer Science & Business Media, 2012.

[Petroni *et al.*, 2019] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, 2019.

[Radford *et al.*, 2019] 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.

[Resnik, 1999] Philip Resnik. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of artificial intelligence research*, 11:95–130, 1999.

[Ross *et al.*, 1996] Sheldon M Ross, John J Kelly, Roger J Sullivan, William James Perry, Donald Mercer, Ruth M Davis, Thomas Dell Washburn, Earl V Sager, Joseph B Boyce, and Vincent L Bristow. *Stochastic processes*, volume 2. Wiley New York, 1996.

[Ryffel *et al.*, 2018] Theo Ryffel, Andrew Trask, Morten Dahl, Bobby Wagner, Jason Mancuso, Daniel Rueckert, and Jonathan Passerat-Palmbach. A generic framework for privacy preserving deep learning. *arXiv preprint arXiv:1811.04017*, 2018.

[Sakaguchi *et al.*, 2019] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*, 2019.

[Smolensky *et al.*, 2016] Paul Smolensky, Moontae Lee, Xiaodong He, Wen-tau Yih, Jianfeng Gao, and Li Deng. Basic reasoning with tensor product representations. *arXiv preprint arXiv:1601.02745*, 2016.

[Solé and Seoane, 2015] Ricard V Solé and Luís F Seoane. Ambiguity in language networks. *The Linguistic Review*, 32(1):5–35, 2015.

[Sproat and Santen, 1998] Richard Sproat and Jan PH van Santen. Automatic ambiguity detection. In *Fifth International Conference on Spoken Language Processing*, 1998.

[Subramanian *et al.*, 2017] Sandeep Subramanian, Sai Rajeswar, Francis Dutil, Christopher Pal, and Aaron Courville. Adversarial generation of natural language. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 241–251, 2017.

[Wallace *et al.*, 2019] Eric Wallace, Pedro Rodriguez, Shi Feng, Ikuya Yamada, and Jordan Boyd-Graber. Trick me if you can: Human-in-the-loop generation of adversarial examples for question answering. *Transactions of the Association for Computational Linguistics*, 7:387–401, 2019.

[Wasow *et al.*, 2005] Thomas Wasow, Amy Perfors, and David Beaver. The puzzle of ambiguity. *Morphology and the web of grammar: Essays in memory of Steven G. Lapointe*, pages 265–282, 2005.

[Wilson, 2006] Deirdre Wilson. The pragmatics of verbal irony: Echo or pretence? *Lingua*, 116(10):1722–1743, 2006.

[Wu *et al.*, 2018] Lijun Wu, Yingce Xia, Fei Tian, Li Zhao, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. Adversarial neural machine translation. In *Asian Conference on Machine Learning*, pages 534–549, 2018.

[Yampolskiy, 2016] Roman V Yampolskiy. Taxonomy of pathways to dangerous artificial intelligence. In *Workshops at the Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[Yampolskiy, 2019a] Roman V Yampolskiy. Unexplainability and incomprehensibility of artificial intelligence. *arXiv preprint arXiv:1907.03869*, 2019.

[Yampolskiy, 2019b] Roman V Yampolskiy. Unpredictability of ai. *arXiv preprint arXiv:1905.13053*, 2019.

[Yarowsky, 1995] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pages 189–196, 1995.

[Yu and Deng, 2016] Dong Yu and Li Deng. *Automatic Speech Recognition.* Springer, 2016.