ABSTRACT

Title of proposal: Gathering Language Data Using Experts

Denis Peskov, 2020

Dissertation directed by: Professor Jordan Boyd-Graber
Department of Computer Science
College of Information Studies
Language Science Center
Institute for Advanced Computer Studies

Natural Language Processing needs substantial amounts of data to make robust predictions. We compare projects that use various techniques—automatic generation, crowd-sourcing, and using domain experts—to generate large textual corpora. Specifically, we curate conversational and question answering NLP datasets.

Large-scale data collection is frequently done through crowd-sourcing, but our question-rewriting task notes the limitation of using this methodology for *generating* data. Standard inter-annotator agreement metrics, while useful for *annotation*, cannot easily evaluate *generated* data, causing a serious quality control issue. This problem is observed while formalizing a question-rewriting task; certain users provide low-quality rewrites—removing words from the question, copy and pasting the answer into the question—for this task without checks. We develop an interface to prevent bad submissions from happening and hand-review over 5,000 submissions.

An alternative low-cost, high-output approach to crowd-sourcing is automation. We explore this approach by creating a large-scale audio question answering

dataset through text-to-speech technology. We conclude that the cost-savings and scalability of automation come at the cost of data quality and naturalness.

We mitigate the quality control issues identified in crowd-sourcing and automation through exploring hybrid solutions. In one hybrid approach, Amazon customer service agents are used for curation and annotation of goal-oriented 81,000 conversations across six domains. By grounding the conversation with a reliable conversationalist—the Amazon agent—we create untemplated conversations and reliably identify low-quality conversations. The language generated from crowd workers is severely lower in quality and would not create natural dialogues.

But *natural* sources of data can be found in specialized communities of interest. We posit that **domain experts can be used to create large and varied datasets that do not require extensive quality control**. In a study on the game of Diplomacy, which investigates the language of trust and deception, Diplomacy community members generate a corpus of 17,000 messages that are self-annotated while playing a game. The language is varied in length, tone, vocabulary, punctuation, and even emojis! Additionally, we create real-time self-annotation system that annotates deception in a manner not possible through crowd-sourced or automatic methods.

We propose future work that leverages experts to create two new machine translation tasks: *coreference evalution* and *cultural adaptation*. Identifying relevant communities for a specific NLP task, and providing a service to them can set new standards for NLP corpora.

Gathering Natural Language Processing Data Using Experts

by

Denis Peskov

Dissertation proposal submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2020

Advisory Committee:
Professor Jordan Boyd-Graber, Chair
Professor Philip Resnik
Professor Michelle Mazurek

Table of Contents

List of Tables

List of Figures

Chapter 1: The Case for Upfront Investment in Data

Computer science can solve tasks across multiple areas: natural language processing, computer vision, biology, etc. Solving tasks for all these domains—translating a sentence between languages, distinguishing a cat and a dog, classifying a mutation—has two abstract and intertwined dependencies: model-building and data collection.[1] The relationship is intertwined since today's models are optimized to draw statistical conclusions from significant amounts of data through machine learning. But, even the most cutting edge modeling techniques are heavily dependent on having *realistic* and *accurate* data for solving a task. Large datasets to date have primarily been gathered through means of low-cost crowd-sourcing (Deng et al., 2009; Rajpurkar et al., 2016; Budzianowski et al., 2018). We argue that a new paradigm of high-quality, expert-reliant data collection can lead to long-term improvements in Natural Language Processing (NLP).

1.1 Where does Data come from?

In the overview, we discuss the two tasks necessary for data collection and explain the importance of data quality for computer science as a field.

Data creation can be broadly categorized into two categories: *generation* and *annotation*. We define *generation* as the creation of a data item that is not previously available (e.g., sequencing a genome, creating a new image, gathering a new sentence from a user, or automatically creating a sentence) (Atkins et al., 1992; Goodfellow et al., 2014; Zhu et al., 2018). We define *annotation* as the application of a label to an existing data item (e.g., classifying a part of the genome, labeling an image as a cat, or describing the sentiment of a sentence) (Deng et al., 2009; Finin et al., 2010; Kozomara and Griffiths-Jones, 2014). In many fields, data must be both *generated* to be representative of the task and then accurately *annotated* to be effective.

The demand of neural models for quantity has caused models to be trained on large, noisy data (Brown et al., 2020). The building blocks of other research areas—gene sequences in biology and individual pixels in computer vision—are not readily human interpretable by default. In more human-intuitive fields like natural language processing, the data has reached the scale where its veracity—the certainty and completeness of the data—cannot be assumed (Qiu et al., 2016). As a result, reformatting and preprocessing the data is necessary for training models (Hinton and

[1] Mitchell (1997) defines a machine learning model as,"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E". Data collection is the experience E.

Salakhutdinov, 2006). This obfuscation from data type or sheer quantity can mask biases and artifacts, as they are no longer obvious to the naked human eye (Pruim et al., 2015; Gururangan et al., 2018). Atkins et al. (1992) posit that,"there is in fact little danger of obfuscation for the major parameters that characterize a corpus: its size (in numbers of running words), and gross characterizations of its content." However, the objectivity of size is questionable; a corpus consisting of the same word repeated a million of times clearly differs from one with a million unique words. They crucially comment that the evaluation of corpora has not been standardized. This focus on size above quality has shaped data creation during the past decade. Since current approaches to machine learning often obscure how decisions are made by a model, the quality of the data is likely neither carefully evaluated by human nor machine.

The current paradigm of crowd-sourcing—"the practice of obtaining needed services, ideas, or content by soliciting contributions from a large group of people and especially from the online community rather than from traditional employees or suppliers"(Merriam-Webster)—for dataset creation has been the main impetus of unreliability in data. Specifically, Natural Language Processing has generally depended on low-cost crowd-sourcing following the popularity of ImageNet (Deng et al., 2009). However, the entirely crowd-sourced annotations thereof still have notable problems after a decade of updates (Yang et al., 2020) and should serve as a cautionary tale. A re-prioritization to working with communities of interest that have a non-financial incentive and verified contributors to generate realistic data is a solution to this problem.

We argue that investing in reliable data upfront, using experts, can address quality control issues and lead to further model accuracy. This improvement in the quality and diversity of data is a prudent long-term investment as high-quality datasets can have shelf-lives of decades (Marcus et al., 1993; Miller, 1995a) while model architectures are frequently supplanted (Vapnik, 1995; Kim, 2014). Additionally, experts can enable tasks in computer science not otherwise possible; generalists cannot annotate medical images and generalists that do not speak a given language cannot generate believable sentences.

1.2 Natural Language Processing

We focus on specifically Natural Language Processing (NLP) since computer science is too broad of a field to cover. We introduce the NLP tasks covered in our work, challenges faced in NLP due to data quality, and the various methods of data collection which impact this quality.

A large focus of NLP is on building models that exploit patterns in language data to solve a variety of tasks: question answering, conversational agents, machine translation, information extraction, etc. However, in the current paradigm of machine learning, models can only answer questions or make translations that they have seen before. This makes *robust* and *natural* data a prerequisite for any meaningful model.

The increasing dependence on neural models has exacerbated the focus on dataset size. Chapter 2 describes the history of data collection in NLP and explains why this dependence has grown over time. At the extreme end, GPT-3 is trained on 499 *billion* tokens, which is the closest anybody has come to training a model based on the entire Internet (Brown et al., 2020). However, not everything on the Internet is relevant or accurate. Training data containing low-quality data unsurprisingly leads to models learning controversial or false conclusions, with high levels of confidence (Wolf et al., 2017; Wallace et al., 2019a).

Additionally, many tasks in NLP depend on accurate annotation. As a thought experiment, if all verbs are labeled as nouns and all nouns are labeled as verbs in the training data, a perfectly designed language model would be confidently wrong in its predictions. Crowd-sourcing with generalists (Buhrmester et al., 2011) assumes that enough unspecialized workers will answer a question correctly. This is a valid assumption for unambiguous, multiple-choice annotation with a large amount pool of annotators. However, many tasks require language *generation*, which cannot be easily verified through IAA. As a result, the NLP corpora for a given task may not be reflective of the *actual* task. This motivates high-quality *generation* and *annotation* for NLP.

We propose an expert-driven paradigm for collecting NLP corpora. First, we show limitations of using unspecialized and automated methods of data collection in Chapter 3. Second, we discuss hybrid approaches—using verified experts paired with external, low-cost data sources (Vukovic and Bartolini, 2010)—in Chapter 4. Third, we describe an expert-designed experiment on evaluating coreference translation between German-English in Chapter 5. This type of work is impossible without collaboration between native speakers in both languages and indirectly evaluates the quality of training data. Fourth, we describe a completely expert-filled experiment on deception involving the board game of Diplomacy in Chapter 6; this project represents a task that could not be meaningful without the use of experienced board game players willing to dedicate a continuous month. We propose an additional project in Chapter 7 that can only be verified with an expert-defined gold standard.

1.3 Proposal

Our past work establishes that the quality of datasets can vary significantly based on who creates the data: experts or generalists. Chapter 7 proposes to extend the use of experts to another subfield of NLP: machine translation, which stands to benefit from increased scrutiny of data quality. This subfield now relies on a crowd-sourcing paradigm for both generation and annotation (Cer et al., 2017; Clark et al., 2020a) and stands to benefit from increased scrutiny. We propose cultural adaptation, a new complicated task that requires cultural experts for evaluation.

A challenge for modern data-hungry natural language processing (NLP) techniques is to replicate the impressive results for standard English tasks and datasets to other languages. Literally translating text into the target language is the most obvious solution. This can be the best option for tasks such as sentiment anal-

ysis (Araujo et al., 2016), but for other tasks such as question answering, literal translations might miss cultural nuance if you directly translate questions from English to German to provide additional training data. While this might allow question answering systems to answer questions about baseball and *Tom Hanks* in German, it does not fulfill the promise of a smart assistant answering a culturally-situated question about *Oktoberfest*. One can find applicable Named Entity modulations by referencing WikiData, a human-interpretable and human-verified representation of Wikipedia. We will want to investigate if this method generates better candidates than an embedding-based approach: namely word2vec. We focus on the task of cultural adaptation of entities: given an entity in English, what is the corresponding entity in a target language. For example, the German *Anthony Fauci* is *Christian Drosten*. An accurate evaluation of this approach requires using Germans and Americans with appropriate cultural backgrounds. This project will show that expert judgment can evaluate a new task in machine translation.

Chapter 2: Natural Language Processing Depends on Data

In this chapter, we discuss the history of NLP, the NLP tasks relevant for our work, and the three types of data collection discussed in this proposal.

The history of NLP outlined in Section 2.1 explains the current dependence on data. Developments in the fields of statistics and linguistics led to the use of raw training data for building of language models. But each NLP task requires its own bespoke training data, such as parallel training data for machine translation. Specifically, we discuss relevant past work for question answering, dialogue, and machine translation in Section 2.2 as background for our research. Certain tasks for these subfields are unable to be solved with naturally-found data and require dataset creation.

Different types of users can *generate* and *annotate* the data needed for these language models. Unspecialized users can be asked to solve tasks through *crowdsourcing* and automated methods can be used to generate data at scale (Section 2.3.3). Hybrid approaches combine cheap and large-scale methods with experts that verify the results (Section 2.3.4). Lastly, data can be gathered and annotated exclusively using experts (Section 2.3.5). We provide the necessary background and past work relevant to these three data pools in Section 2.3. We explain the models and metrics that are used in solving these tasks in Section 2.4.

2.1 How Language Models Begot Training Data

Our understanding of language has been quantified through formalizing tasks that provide evidence for a theory. These include the Shannon game (Shannon et al., 1949) and the Turing Test (Turing, 1950). NLP continues to explore and quantify language through the introduction of new tasks, such as question answering, machine translation, and dialog. Each of these tasks is "solved" through the construction of a system. However, building this system and then evaluating it depends on data.

The increasing importance of statistics in linguistics brought forth Natural Language Processing.[1] Performing language tasks with simplified rules and limited vocabulary was the paradigm for linguistics (Wittgenstein, 1953; Berko, 1958). Linguistics developed a statistical slant in the 20$^{\text{th}}$ century, mainly with the insights of

[1]The development of the computer and the nearly immediate connection to human language is the other major half. Alan Turing proposed the Turing Test to evaluate if a machine can converse in a manner indistinguishable from a human (Turing, 1950). The test explores if the variance among humans is large enough for a clever computer to fool a human judge. Obviously one cannot have a conversation with a machine in the first place without NLP!

Firth and Chomsky. J.R. Firth declared that, "you shall know a word by the company it keeps" (Firth, 1957). This insight serves as the foundation of embedding-based representations of language in modern-day NLP. Chomsky (1986)'s Universal Grammar serves as a stepping stone between linguistics and information theory. The existence of an innate predisposition to language in children, rather than a dependence on learning everything, precipitates the application of statistics to language. If grammar can be universal, why could statistics not be applied to all languages in a universal manner? These developments led to the emergence of language models built with data, rather than rules in NLP.

The language model has created the dependence on training data, with which this proposal is concerned. Statistical language modeling was proposed in the 1980s (Rosenfeld, 2000) and has slowly taken over linguistic journals as the dominant approach for solving language tasks. The co-occurrence of words in the form of a n-gram model became the paradigm.

$$P(w_i|h_i) = P(w_i \,|\, w_{i-n+1}, \ldots, w_{i-i}) \tag{2.1}$$

where w_i is the ith word in a sentence and h_i is the history of words that came before. Furthermore, this method can be applied to *any* symbols, and not just language, which has made NLP methods useful for fields like biology.

This type of language model is entirely dependent on training data due to its lack of any constructed rules or linguistic knowledge. A language model trained on inaccurate and nonsensical language data will confidently predict nonsense, as it has no understanding of rules, grammar, or language. A machine has no intrinsic understanding of what is signal and what is noise, and it is up to the intrepid scientist to specify how a snippet of language should be correctly understood by the machine. The probability of "computer science" occurring more often than "computer aardvark" in a language model is subject entirely to the training data rather than any ontological or linguistic truth. This is a key insight of information theory (Shannon et al., 1949), which reduces linguistic information to a numerical representation. Information theory is a logical successor to Zipf's Law (Zipf, 1935), which identifies that there is a strong relationship between the rank of a word and its frequency: the first-order word occurs notably more often than the second-order word, the second-order word occurs more often than the third-order word, and so on. This statistical distribution of language is necessary for machine learning to work and this insight applies not only to words, but to phrases (Williams et al., 2015), language learning (Powers, 1998), and many non-NLP phenomena such as website usage (Jiang et al., 2013).

The most obvious option for training this language model is to use easily-found, naturally-occurring data. The development of the Internet in particular led to an explosion of available textual data for language models. The amount of data created from 2010 to 2017 has increased 13-fold.[2] The latest raw text models are trained on *de facto* the entire Internet (Brown et al., 2020). There appears to be a limit to how much a language model can learn from statistics without understanding

[2]https://www.statista.com/statistics/871513/worldwide-data-created/

language, but that limit has not yet been ascertained.

Language models can be created for different NLP tasks, but each requires a different type of training data. For example, machine translation requires parallel text, which increases the standard for training data quality.

2.2 Tasks

We focus on three NLP tasks in our research: Machine Translation, Question Answering, and Dialogs.

2.2.1 Machine Translation

Machine translation was one of the earliest uses of NLP. One needs text from multiple languages for this task, which led to the collection of parallel texts across languages. We discuss several key datasets in the area.

Machine translation as a NLP task only dates back half a century. Yet it has already undergone dramatic changes in methodology. The Georgetown Machine Translation experiments translated dozens of sentences from Russian into English in 1954 (Hutchins, 2004). The system used a rules-based approach that encoded grammar and lexical endings to convert the input sentence to the target language. This proof of concept began a decade of research into the topic, until a realistic assessment of results concluded that machine translation could not be solved in several years, as initially presumed.

The rise of statistical machine translation began with the recognition that parallel French-English text from the Canadian parliament could be used to train more flexible models than previously possible (Berger et al., 1994). Thinking of languages as a noisy channel model—English is a garbled version of French—allowed researchers to align parallel corporate and *learn* how language can be automatically translated. The equation is:

$$\hat{e} = \arg\max_{e} p(e|f) \tag{2.2}$$

where e is the English word and f is the French word.

Since this development, parallel corpora have been sought after in every conceivable domain. The Bible, books, medical records, and the Internet predate NLP. The Bible (Resnik et al., 1999) is a prime example of a corpus that when annotated can provide parallel data for "2000 tongues".[3] Literature and movie captions (Varga et al., 2007), librettos (Dürr, 2005), medical information (Deléger et al., 2009), and the Internet (Resnik and Smith, 2003; Smith et al., 2013) can all be sources of parallel data. The independent growth of these corpora will provide language models with *found* data, which can be used for training supervision.

Data generation has become necessary for this subfield given the large amount of data required, and all the possible languages to cover. The Workshop on Machine

[3] In this case, only for a dozen tongues.

Dataset	# of Sentences	Data Source
ContraPro	12,000	Found
Canadian Parliament	1,300,000	Found
EuroParl	11,000,000	Found
TyDi	204,000	Crowd
MLAQ	12,000	Hybrid
XQuAD	1,190	Expert

Table 2.1: A tabular summary of machine translation datasets.

What is the English meaning of caliente?
What is the meaning of caliente (in English)?
What is the English translation for the word "caliente"?

Table 2.2: Three questions from TREC 2000 data that are believably varied. The test questions were carefully crafted by experts.

Translation facilitates model-building for machine translation (Koehn and Monz, 2006). Statistical Machine Translation has been supplanted by neural machine translation (Wu et al., 2016). MLQA and XQuAD automatically generate paired questions through machine translation (Lewis et al., 2019; Artetxe et al., 2019) TyDi (Clark et al., 2020a) gives crowd-sourced users prompts from Wikipedia articles. Our proposal introduces a new machine translation task, cultural adaptation, that requires collecting translations from cultural experts for gold standard evaluation.

2.2.2 Question Answering

Another task heavily dependent on training data is Question Answering (QA). In the current machine learning paradigm, QA can only answer a question with a previously seen answer. Therefore, the coverage of questions and answers is important as models trained on trivia questions cannot answer inquiries about medical symptoms, and vice versa. We discuss the relevant history of question answering and review the most relevant datasets.

The Text Retrieval Conference established Question Answering as an annual, formalized task (Voorhees et al., 1999). The questions were carefully curated every year and modifications to the question answering task were made. Table 2.2 shows examples of questions that are intended to fool systems reliant on literal information extraction.

The neural era ushered in larger more diverse Question Answering datasets, with SQuAD (Rajpurkar et al., 2016, 2018a) being the most popular leaderboard for models. The amount of questions went from being measured in the *hundreds* to being measured in the *hundreds of thousands*. Example questions are provided in Table 2.3. Large influential question answering datasets include SQuAD 1.0 (Rajpurkar et al., 2016), SQuAD 2.0 (Rajpurkar et al., 2018a), MS Marco (Bajaj et al., 2016), TriviaQA (Joshi et al., 2017) QuAC (Choi et al., 2018), Quizbowl (Rodriguez

Questions	Answers
"Which laws faced significant opposition?"	later laws
"What was the name of the 1937 treaty?"	Bald Eagle Protection Act

Table 2.3: The paper examples from SQuAD. In contrast with Table 2.2, these questions are done through crowd-sourcing and Wikipedia and are not carefully planned.

Dataset	# of Questions	Data Source
CoQA	8,000	Crowd
SQuAD 1.0	100k	Crowd
SQuAD 2.0	50k	Crowd
QuAC	100k	Crowd
TriviaQA	95k	Hybrid
Quizbowl	100k	Hybrid
Natural Questions	300k	Hybrid
MS Marco	1000k	Found
TREC-8	200	Expert
Trick Me	651	Expert

Table 2.4: A tabular summary of key question answering datasets. The datasets described as hybrid all scrape or use naturally-occurring language and then supplement it with crowd-sourced annotation.

et al., 2019), and Natural Questions (Kwiatkowski et al., 2019). We summarize the size of these datasets and their user pools in Table 2.5.

These datasets are frequently instances of machine reading comprehension (Rajpurkar et al., 2016, MRC), which requires that computers can take a single question and select the answer from a passage of text. However, QA models struggle to generalize when questions do not look like the standalone questions systems in training data: e.g., new genres, languages, or closely-related tasks (Yogatama et al., 2019). Unlike MRC, *conversational question answering* requires models to link questions together to resolve the conversational dependencies between them: each question needs to be understood in the conversation context. For example, the question *"What was he like in that episode?"* cannot be understood without knowing what *"he"* and *"that episode"* refer to, which can be resolved using the conversation context. CoQA creates conversational question answering around different domains–Wikipedia, children's stories, News Articles, Reddit,literature, and science articles–by pairing Mechanical Turk crowd-sourced workers together (Reddy et al., 2019).

Creating questions in languages other than English is another current research direction as touched upon in Section 2.2.1. MLQA (Lewis et al., 2019), XQuAD(Artetxe et al., 2019), and TyDi (Clark et al., 2020a) are recent examples.

Recent work has began to acknowledge that crowd-sourced users may not be an optimal source for data or participants. Wallace et al. (2019b) work with

Dataset	# of Questions	Data Source
DSTC2	1,612	Found
Ubuntu Dialog	930,000	Found
Reddit	256,000,000	Found
OpenSubtitles	316,000,000	Found
DSTC2	1,612	Crowd
CoQA	8,000	Crowd
MultiWOZ	8,438	Crowd

Table 2.5: A tabular summary of key question answering datasets. The datasets described as hybrid all scrape or use naturally-occurring language and then supplement it with crowd-sourced annotation.

the Quizbowl community to rewrite questions be adversarial. Clark et al. (2020b) emphasize that natural speakers of a language must be used to write authentic questions in languages outside of English, although the source of theses speakers is still crowd-sourced unverified users as they do not have other scalable access to speakers of typologically diverse languages. Boyd-Graber (2020) calls into question the paradigm of using crowd-sourced workers as the measure for human baselines, rather than evaluating through a play test.

2.2.3 Dialogs

Like question answering, conversational datasets have been gathered for different purposes and with different techniques. We provide a brief history of conversational datasets and summarize the relevant datasets.

Existing *found* conversational data has been repurposed as NLP datasets. Ubuntu threads provide millions of conversations of technical support (Lowe et al., 2015). Reddit, a collection of threaded comments about diverse subjects, and OpenSubtitles, collections of movie and television subtitles, provide millions of sentences as training data (Henderson et al., 2019).

However, *found* datasets cannot cover all domains and languages. Therefore, *generating* conversational datasets becomes a NLP need. The Dialog State Tracking Challenge (Henderson et al., 2014) formalizes the dialog task on an annual basis and creates several relatively-small, crowd-sourced datasets focusing on different conversational tasks. MultiWOZ proposes a framework for simulated conversations, which is necessary for domains containing sensitive data that cannot be released (Budzianowski et al., 2018).

2.3 Data Collection Type

Training and test data for machine learning can come from one of four sources: automation, crowd-sourcing, a hybrid mix of the crowd with experts, and exclusively experts. We discuss the seminal work for each of these data pools.

2.3.1 Finding

Reusing existing text through scraping websites or forums and re-purposing historical documents can create datasets with little effort. We define the this type of data as *found*.

The Internet contains enormous amounts of information that is varying in quality. Amazon reviews (McAuley et al., 2015), Twitter (Banda et al., 2020), and Wikipedia (Vrandečić and Krötzsch, 2014) provide language from unverified users on the Internet. These datasets are large, but contain noise due to having a low barrier to entry for contributors.

Higher quality datasets often come from organizations that have an incentive to control or report their data. Enron emails are original emails collected into a dataset (Klimt and Yang, 2004). EuroParl is collected from professionally translated official documents (Koehn, 2005). Literature comes from a verified author (Iyyer et al., 2016), as does journalism (Lewis et al., 2004). The Titanic had an accurate list of passengers. The United Nations maintains detailed datasets about global populations. New York City releases the Taxi and Limousine Commission data. The World Trade Organization releases a comprehensive collection of legal disputes.

The original source of this type data can be experts (e.g., World Trade Organization lawyers and translators) or they can be unverified online users (e.g., Reddit users). Since this data was not intentionally intended for NLP, *annotation* is often required. Additionally, found data can be created by experts or unverified generalists, depending on the task and the desired quality.

2.3.2 Automation

Not all data necessary for NLP can be found. Therefore, data *generation* becomes necessary. Synthetic data can be created according to fixed rules or templates, which we refer to as automation. Augmentation is a frequent phrasing of this way of creating data (Kafle et al., 2017). This method can create datasets of any scale, but it does not guarantee their authenticity.

Templates can be used to create datasets unlimited in scale, but dubious in realism. Filatova et al. (2006) generate questions using specific verbs for various domains: airplane crashes, earthquakes, presidential elections, terrorist attacks. In their own words, their automatically created templates are "not easily readable by human annotators" and the evaluation requires a lengthy discussion. Examples of questions generated though templates include the following nonsensical questions about specific earthquakes:

- *Is it near a fault line?*
- *Is it near volcanoes?*

Chapter 3 describes our project in which text-to-speech is used to create a dataset of 500,000 audio files. While large, our dataset is limited to a single female voice and read in a cadence notably different from that of realistic Quizbowl experts. Additionally, our automation method depends on the existence expert-written questions in the first place. However, to create a dataset of the same size with human

experts would require thousands of hours. Mozafari et al. (2014) propose using active learning to minimize the human effort needed to gather large-scale datasets; one gathers annotations for a subset of the data and then extrapolates those labels to similar unlabeled data. This serves as a segue into the next type of data creation method: crowd-sourcing.

2.3.3 Crowd-Sourcing

We define crowd-sourcing and automatic data generation techniques, explain their history, and comment on the repercussions of the wide-spread use of this data pool in NLP today. Crowd-sourcing is "the practice of obtaining needed services, ideas, or content by soliciting contributions from a large group of people and especially from the online community rather than from traditional employees or suppliers" (Merriam-Webster). Crowd-sourcing, in the applied sense, relies on unspecialized users and is the most popular way to create new datasets in NLP today.

The reliance on crowd-sourcing low-cost labor is a phenomenon just over a decade old. Deng et al. (2009) built ImageNet using Mechanical Turk—a crowd-sourcing marketplace that makes it easier for individuals and businesses to outsource their processes and jobs to a distributed workforce who can complete these tasks virtually (Amazon, 2021)— crowd-sourcing for annotating WordNet with images, which ushered in this paradigm. Visual classification tasks are maximally simple in nature since annotators are asked to decide if an image contains a Burmese cat. Figure 2.1 shows their interface. Despite this, disagreement is a major problem and a minimum of 10 users are used to guarantee a level of confidence. Even with constant updates, the dataset still has limitations a decade later from the initial scaling methodology used to create it (Yang et al., 2020).

Crowd-sourcing spread to other disciplines other than machine vision as a source for research data. Buhrmester et al. (2011) claim that Amazon Mechanical Turk gathers "high-quality data inexpensively and rapidly" for psychology. However, the evidence for this claim stems from having participants fill out a survey and is primarily evaluated on the time required, rather than the quality of the final result. In their survey, users report that their motivation for using Mechanical Turk is higher on a Likert scale for enjoyment than for payment. Given that nearly every NLP task requires that users complete a large amount of previous tasks (1000+) and with a nearly perfect accuracy (90%+), this claim seems unlikely to hold for the average producer of NLP data. As a note of caution, Mason and Suri (2012) claim that spammers are likely to target surveys on Mechanical Turk.

Crowd Flower, renamed as Figure Eight, is a platform similar to Mechanical Turk, but with a focus on quality control. While Mechanical Turk keeps track of Human Intelligence Tasks (HIT)—the name for each individual task—accuracy rates, this metric depends on task providers to manually evaluate the data and provide feedback about the worker. Needless to say, this level of oversight is unlikely to be done carefully for thousands of tasks. Crowd Flower's innovation is to include a test set with each task which monitors that users' responses correspond to gold labels. As early adopters of crowd-sourcing, Finin et al. (2010) use Crowd Flower for

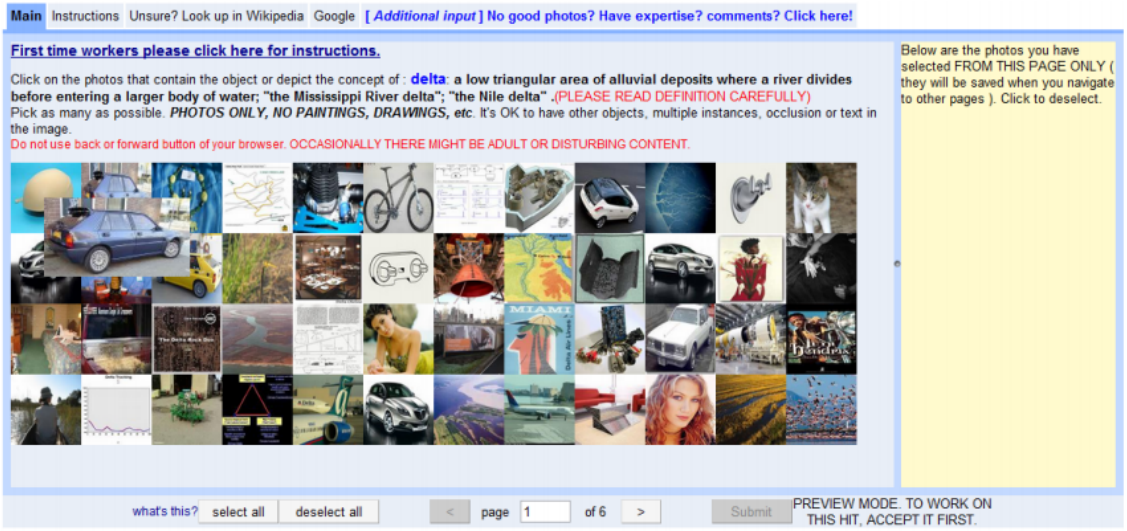

Figure 2.1: Deng et al. (2009) pioneers Mechanical Turk use for Computer Science. Simple *annotation* tasks can be done reliably with crowd-sourcing since selecting if an image belongs to a WordNet category (e.g., car, bicycle, delta) is a relatively objective and straightforward task. However, many NLP tasks are not so clear-cut.

annotating named entities in Twitter. However, most annotations are completed by a few prolific workers, which opens up the dataset to potential biases. Furthermore, creating a crowd-sourced dataset with Crowd Flower is possible for *annotation* but not for *generation*.

From computer vision annotation, crowd-sourcing transferred over to natural language processing (Callison-Burch et al., 2015). Using a nonprofessional user pool is the default manner for collecting large datasets for NLP as it can generated and annotated quickly and cheaply. As on example, large question answering datasets involving Wikipedia and search engines—SQuAD, SearchQA—use crowd-sourcing to generate questions (Rajpurkar et al., 2016; Dunn et al., 2017a).

The two main benefits to this data source are the cost and the rapid rate of data collection. The cost is unquestionably lower for an employer or researcher to use the crowd rather than internal employees. Crowd workers are paid a fraction of what full-time employees would receive for the same task and do not receive any benefits (Whiting et al., 2019).[4] Largely due to the variations in cost-of-living around the world and flexibility of the work, the pay is appealing to some workers. The demographics of the platform more accurately model the United States than the average college student, at least for psychology experiments (Buhrmester et al., 2011). As a result, Amazon Mechanical Turk has over a hundred-thousand workers, thousands of which are available at any moment (Difallah et al., 2018). Modular tasks can be completed in hours in crowd-sourcing, as thousands of temporary workers complete tasks faster than a handful of employees.

The con to crowd-sourcing is that quality control becomes the central challenge

[4]This clearly is not a pro from the worker's perspective.

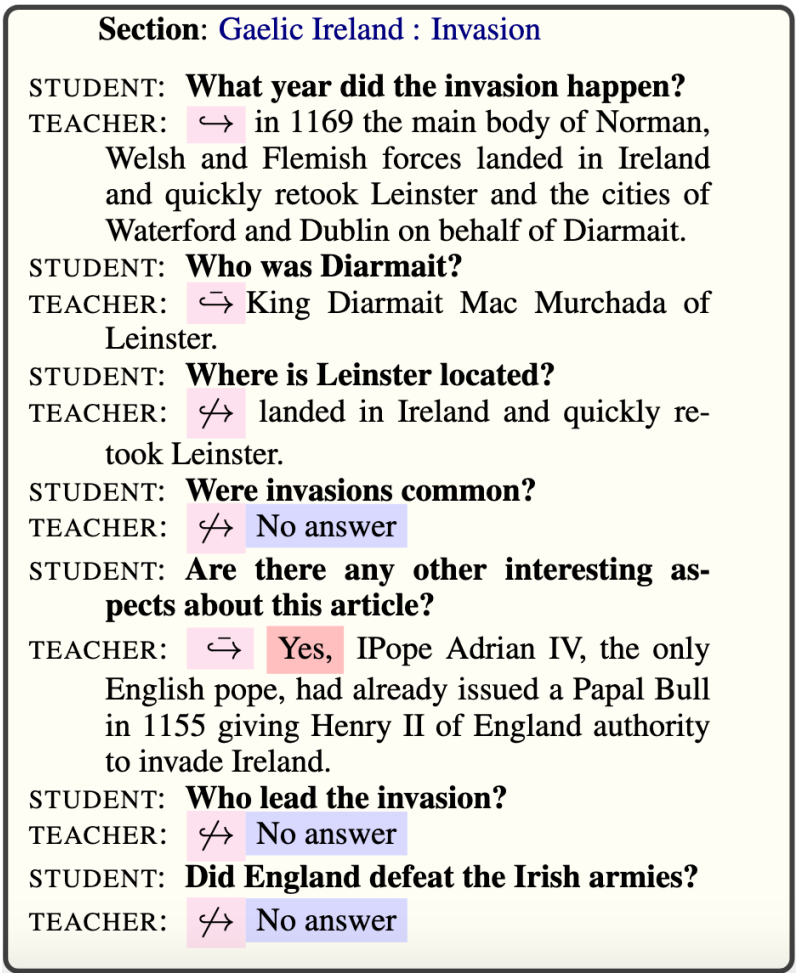

Figure 2.2: Crowd-sourcing can also be used to generate large-scale NLP data. However, *generation* creates a quality issue not present in *annotation*. In this particular example, Choi et al. (2018) highlight that the teacher does not provide quality responses. However, the student's conversation is quite unnatural and has grammatical issues.

for crowd-sourcing NLP data. Mathematically, average accuracy needs to exceed 50% for reliable annotators to overcome their noisy peers (Kumar and Lease, 2011). Given that certain tasks are highly sparse, this is not a threshold that is always achievable. Zaidan and Callison-Burch (2011) show that data gathered from crowd-sourcing for machine translation nets a BLEU score nearly half the size of professional translators, and only one point higher than an automatic machine translation approach. Other studies have shown that users tend to voluntarily provide inaccurate data (Suri et al., 2011) and misrepresent their background (Chandler and Paolacci, 2017; Sharpe Wessling et al., 2017). Last, there is an upper-bound to the complexity of crowd-sourced tasks. Crowd workers have been shown to become less reliable and efficient for tasks that are not straightforward (Finnerty et al., 2013). Figure 2.2 shows that more complicated NLP task instructions are not followed in good faith.

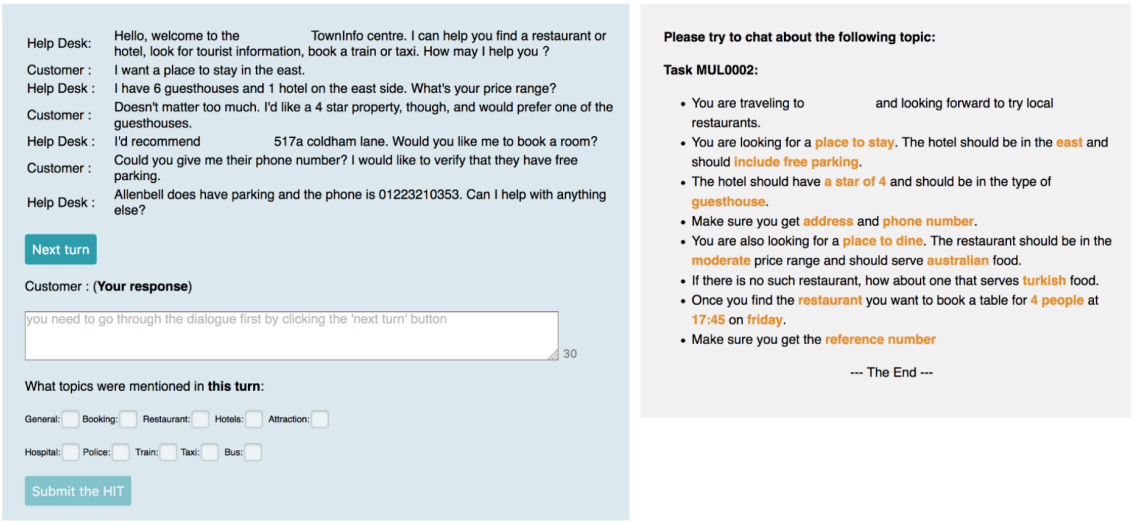

Figure 2.3: Hybrid approaches try to control the quality of language *generated* by the crowd. MultiWoz (Budzianowski et al., 2018), creates a rigid template for the user conversation, avoiding the worst quality issues at the expense of user creativity.

As a tangential consideration, legal regulation may ultimately limit the effectiveness of this technique, since it is completely unregulated by current employment practices (Wolfson and Lease, 2011).

Chapter 3 reveals quality issues in this technique through a project that crowd-sources question. We use Mechanical Turk's crowd to rewrite sequential questions into a standalone format. However, extensive manual review is necessary to remove the low-quality contributions from the data pool. Hybrid methods can provide quality control for crowd-sourcing.

2.3.4 Hybrid

Hybrid approaches aim to enhance crowd-sourcing by overseeing unspecialized labor or automatic methods with expert knowledge. This combination lowers cost and allows for data scaling, while maintaining a certain level of quality control. We define hybrid user pools and discuss past projects.

We define hybrid data collection sources as any that combine a cost-saving pool, such as crowd-sourcing or automation, with expert supervision. This is a natural extension of crowd-sourcing and does not require as detailed of a historical overview: once quality issues were noted, attempts were made to remedy them. For *generation*, crowd-sourced workers can be combined with trained agents to create data for a given NLP task. For *annotation*, crowd-sourced workers can be supervised by trained experts.

As an illustrative example, Zaidan and Callison-Burch (2011) propose an oracle-based approach to identify the high quality crowd-sourced workers and rely on their judgments. The paper claims that crowd-sourcing can lead to a notable reduction in cost without a complete loss in quality. Their approach crucially depends

on having expert (professional) translations as a reference point.

Numerous other approaches have proven successful for a myriad of tasks. Kochhar et al. (2010) use a hierarchical system for database, specifically Freebase, population. First, an item is populated by automatic methods, then issues are escalated to volunteer users, and any remaining issues are escalated to trained experts. Ade-Ibijola et al. (2012) design a system for essay-grading that allows for teacher oversight and compare their results to area experts. Hong et al. (2018) optimize the productivity of medical field experts by providing additional reference resources and standardizing databases. FEVER (Thorne et al., 2018a) relies on super-annotators on one percent of the data as a comparison point for all other annotations for FEVER. Errors made by crowd-sourced workers on Named Entity Recognition can be clustered and identified, which in turn can be escalated to a skilled arbitrator to improve task guidance (Nguyen et al., 2019). Having an expert-written template that crowd workers must follow eliminates the worst-quality submissions (Budzianowski et al., 2018). This example is provided in Figure 2.3. Combining trained and untrained workers can be used for generating Wizard-of-Oz personal assistant dialogs (Byrne et al., 2019).

Furthermore, there are two crowd-sourcing platforms whose business model relies on this hybrid approach. Crowd Flower, mentioned in Section 2.3.3, attempts to booster the reliability the crowd by requiring the task master to create gold-standard test questions, which are interspersed among the data being collected (Vakharia and Lease). While not necessarily using experts, this provides an automatic quality filter that down-weights the reliability of annotations made by the least accurate–as determined by the gold-standard test set—annotators. Crucially, this approach can only work for *annotation*, as generation quality cannot be quickly assessed. ODesk is a crowd-sourcing platform that provides a hybrid approach, as it relies on crowd-sourcing from the Internet, but vets the participants to have a matching skill-set for the task (Vakharia and Lease).

2.3.5 Expert

We define "experts", provide a brief summary of relevant datasets, and introduce a dataset *generated* and *annotated* by domain experts.

We formally define "expert" as "a person with a high level of knowledge or skill relating to a particular subject or activity"Cambridge Dictionary. For NLP, this requires that the person have some sort of incentive to *accurately*, as opposed to quickly, complete their task. These experts can be trained or they can be found in specialized communities of interest. The amount of expert-only datasets for NLP are limited due to the high cost associated with hiring experts and quality assurance. Given the increasing investment and interest in the field, this route for data collection will be the best long-term investment. We discuss existing sources of this kind of data, methods for generating language data, and methods for annotating language data.

Language recorded *naturally* for other purposes has led to datasets that have withstood the test of time. The United Nations, New York City, and the World

Message	Sender's intention	Receiver's perception
If I were lying to you, I'd smile and say "that sounds great." I'm honest with you because I sincerely thought of us as partners.	Lie	Truth
You agreed to warn me of unexpected moves, then didn't . . . You've revealed things to England without my permission, and then made up a story about it after the fact!	Truth	Truth
. . . I have a reputation in this hobby for being sincere. Not being duplicitous. It has always served me well. . . . If you don't want to work with me, then I can understand that . . .	Lie	Truth
(Germany attacks Italy)		
Well this game just got less fun	Truth	Truth
For you, maybe	Truth	Truth

Table 2.6: In contrast to the previous conversations involving crowd workers, conversations involving experts *generate* creative, and even humorous, language. Additionally, the *annotation* of truthfulness is not possible with crowd-sourcing, since it requires the *generator's* real-time knowledge. This conversation snippet is from the Diplomacy project discussed in Chapter 6.

Trade Organization are all organizations that release reliable large-scale data, as discussed in Section 2.3.1.

However, existing, or *found*, data sources do not cover all NLP tasks and domains. Therefore, *generation* by experts is necessary. The best example of this in NLP is WordNet, which was built in the 1980s. The ontology was carefully crafted using a small batch of Princeton psychology graduate students—arguably some of the best experts in the English language and unarguably participants with a strong incentive to provide meaningful data—over an extended period of time (Miller, 1995a).

Annotations are possible to collect from non-experts, but often at the expense of their accuracy. Programmers can self-annotate their code for easier future accessibility (Shira and Lease, 2010). Hate speech annotation is more accurate with expert annotators than amateur ones (Waseem, 2016). In the medical field, the lack of expert annotation poses a barrier to large-scale NLP clinical solutions (Chapman et al., 2011). Unsurprisingly, doctor annotation is more accurate than online generalist annotation for medical diagnoses (Cheng et al., 2015).

Multiple studies comparing the quality of crowd-sourced work and expert work have been done. Mollick and Nanda (2016) compare expert to crowd judgment for the funding of theater productions. They conclude that most decisions are aligned between the two pools, but that crowds are more swayed by superficial presentation than underlying quality. Leroy and Endicott (2012) compare annotations of text difficulty between a medical librarian and a non-expert user and do not see a large difference on a small sample size.

Chapter 6 presents a project that works with the Diplomacy, a popular board-game, community to *generate* and *annotate* a natural conversational dataset for the task of deception. The language in this dataset is realistic and impossible to generate with unspecialized crowd users. An example conversation is provided in Table 6.1.

2.4 Models & Metrics

Data does not exist in a vacuum. Therefore, we summarize popular models used with the data, and the metrics used to evaluate models and data.

2.4.1 Logistic Regression

According to Jurafsky and Martin (2000), the logistic regression is a basic *discriminative* model, meaning that it can classify items into one of several classes. It relies on using features x to predict class y by learning a vector of weights, w, and a bias term, b according to:

$$z = w \cdot x + b \tag{2.3}$$

z is then passed through a sigmoid function to transform the values to a probability:

$$y = \sigma(z) = \frac{1}{(1 + e^{(} - z))} \tag{2.4}$$

There are two phases to logistic regression: training and test. During training, stochastic gradient descent and cross-entropy loss learn the optimal weights of w and b. Cross-entropy loss calculates the difference between the predicted $\hat{y}$ and the true y. The gradient descent algorithm (Ruder, 2016) finds the minimum loss.

At test time, for each example the highest probability label is predicted. Multinomial logistic regression allows for the prediction of more than two classes.

Other important parts of logistic regression, and machine learning more broadly, are batching—calculating gradient across multiple examples at once to have a better estimate in which direction to adjust weights—and regularization (Tibshirani, 1996)—penalizing large weights in the function to generalize results from the training data to unseen data.

The logistic regression model is interpretable since the weight of each feature is transparent in the final prediction. Certain features have higher weights than other ones. This has made the logistic regression a popular baseline model for machine learning. Its interpretability with the current state-of-the-art model: neural networks.

2.4.2 Neural Models

Neural networks are an old idea that gained widespread adoption the last decade. The idea of a perceptron was proposed as early as the 1940s (McCulloch

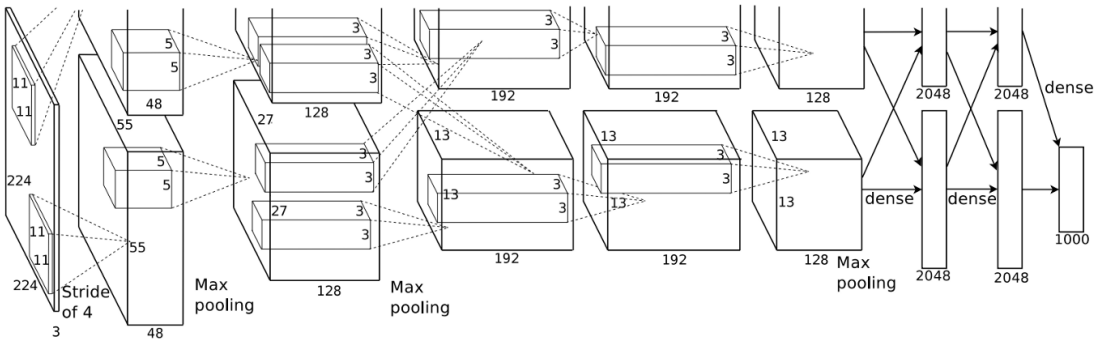

Figure 2.4: Krizhevsky et al. (2012)'s CNN architecture.

and Pitts, 1943). Backpropogation, the training algorithm behind a neural network, was proposed in 1986 (Rumelhart et al., 1986). However, it was not until the 21st century that computing infrastructure allowed neural networks to be effectively applied. AlexNet applied to the ImageNet classification dataset shows a sizable improvement over past machine learning methods (Krizhevsky et al., 2012).

Neural networks are a more powerful classifier than logistic regressions and can be shown to learn any function. Additionally, they often avoid dependence on carefully crafted features and learn their own representations for the task(Jurafsky and Martin, 2000). Further research into *deep learning* created deeper and computationally more expensive neural networks, specifically for machine vision. From there, the application of neural networks branched out into other domains, including NLP.

All neural networks depend on a *loss function* and *backpropogation* The *loss function* tells the neural network how quantitatively wrong a prediction is. Popular loss functions include Cross Entropy Loss—often used for logistic regression and classification tasks— and Mean Squared Error (Sammut and Webb, 2010). *Backpropogation* percolates weight adjustment with the chain rule throughout the entire network. This is based on the derivative of the error, which is calculated through the *loss function*. Additionally, rather than relying on n-gram language models (Section 2.1), neural language models reference prior context as *embeddings* that represent the word(s). This means that the neural network can understand that "cat" and "dog" are similar, and can be treated similarly, whereas a n-gram model assumes independence. word2vec (Mikolov et al., 2013a) and GloVe (Pennington et al., 2014) embeddings are commonly used pre-trained embeddings. This powerful innovation allows has led to the current state-of-the-art dependence on Transformers (Section 2.4.5).

Model architectures have evolved over time in NLP. Convolutional Neural Networks (CNN) (Krizhevsky et al., 2012) applied to ImageNet kicked off the applications of deep neural networks. Figure 2.4 shows the architecture of that model. A CNN has several convolution layers that alter the input, as well as pooling layers that condense the input. This architecture is relevant for machine vision in particular since clusters of pixels, rather than an individual one are important for

understanding the content of an image.

We focus on their successors: Deep Averaging Networks (Section 3.1) and Recurrent Neural Networks (Section 2.4.4) in our research.

2.4.3 Deep Averaging Network

The Deep Averaging Network, or DAN, classifier proposes a simple architecture with comparable results to more complicated neural models. It has three sections: a "neural-bag-of-word" (NBOW) encoder, which composes all the words in the document into a single vector by averaging the word vectors; a series of hidden transformations, which give the network depth and allow it to amplify small distinctions between composed documents; and a softmax predictor that outputs a class.

The encoded representation $\mathbf{r}$ is the averaged embeddings of input words. The word vectors exist in an embedding matrix $\mathbf{E}$, from which we can look up a specific word w with $\mathbf{E}[w]$. The length of the document is N. To compute the composed representation r, the DAN averages all of the word embeddings:

$$\mathbf{r} = \frac{\sum_i^N \mathbf{E}[w_\mathrm{i}]}{N} \tag{2.5}$$

The network weights $\mathbf{W}$, consist of a weight-bias pair for each layer of transformations ($\mathbf{W^{(h_i)}}$, $\mathbf{b^{(h_i)}}$) for each layer i in the list of layers L. To compute the hidden representations for each layer, the DAN linearly transforms the input and then applies a nonlinearity: $\mathbf{h_0} = \sigma(\mathbf{W^{(h_0)}r} + \mathbf{b^{(h_0)}})$. Successive hidden representations h_i are: $\mathbf{h_i} = \sigma(\mathbf{W^{(h_i)}h_{i\text{-}1}} + \mathbf{b^{(h_i)}})$. The final layer in the DAN is a softmax output: $\mathbf{o} = \mathrm{softmax}(\mathbf{W^{(o)}h_L} + \mathbf{b^{(o)}})$. This model is used and modified in Chapter 3.

2.4.4 Sequence to Sequence

Unlike the DAN, Recurrent Neural Networks (RNN) (Elman, 1990) take into account the sequence of the input, which is important given the ordered nature of language.

The long short-term memory (LSTM) (Gers et al., 1999) modifies the RNN by allowing it to discard past information.

According to Goldberg (2017), *Sequence to Sequence* refers to a model that ingests a sequence of text and then generates a sequence of text, rather than a single classification, as an output. The architecture necessary for this is called Encoder-Decoder, as the text input is first encoded—meaning a sequence of text has been transformed into a numerical representation—and then decoded—this representation is then transformed back into text. Machine translation (Section 2.2.1) is a clear example where this applies. If a sentence in German needs to be transformed into English, then the German sentence is first encoded into a numerical representation and then decoded into an English sentence. *Attention* (Bahdanau et al., 2014) looks at different parts of the encoded sequence at each stage in the decoding process. Visualizing attention provides a mild level of interpetability as the model looks at a

specific part of the input. We use these models in Chapters 4 and 6, as the current state of the apart for NLP.

2.4.5 Transformers

The Transformer model simplifies the architecture and dispenses with recursions and convolutions (Vaswani et al., 2017), relying instead entirely on attention.

ELMo (Peters et al., 2018), used in Chapter 4, improves on GloVe embeddings (Pennington et al., 2014) by allowing a word's embedding to adjust to the context, rather than being committed to having a single word sense. BERT improves the embeddings further by looking at context bidirectionally, meaning that words that follow a word influence its embedding. These pre-trained embeddings can be further fine-tuned to accommodate a specific domain's context.

2.4.6 Evaluation

But how does one evaluate a model, or the underlying quality of data? Model evaluation is specific to a general task: classifying images correctly for ImageNet or answering a question for SQuAD. There is a goal of achieving the highest quantitative accuracy on a particular task (Wang et al., 2019a); qualitative analysis of *what* was answered correctly in contrast to another model is often an after-thought (Linzen, 2020).

Data evaluation is necessary for crowd-sourcing. For annotation, one can compare the annotations of users to one another using *Inter-Annotator Agreement* (IAA). Nowak and Rüger (2010) show that for simple image classification tasks, the majority vote of unspecialized users is comparable to expert annotation.

However, there is no obvious metric to compute IAA for *generation*. In question answering, one may limit the possible answers to existing pages in Wikipedia, or some other finite source, to avoid string matching problems. But, language is complex and multiple users could write equally valid questions that do not appear similar at the character level. Table 2.2 is one such example.

The interest in neural techniques and a black box mindset precipitated an ever-increasing race for data; the largest dataset, not the best model architecture may be the key differentiating factor. But how to evaluate the influence of data rather than architecture is an open research question.

Chapter 3: Automation and Crowd-Sourcing for Data

Two cheap methods of creating large neural-scale datasets are automatic generation of synthetic data and crowd-sourcing generalist users. We discuss a large dataset created with Text-To-Speech technology, and the limitations thereof beginning with Section 3.1.[1] We discuss crowd-sourcing for *generating* questions in Section 3.6. [2] These are two methods that are meant to create large-scale datasets, but at the expense of naturalness or quality. While they are able to create large datasets—hundreds of thousands of questions in this Chapter—the quality control process is manual, time-consuming, and subject to error. Both projects require having an expert, a trivia player and native English speaker, verify the generated data.

3.1 Automated Data Creation for Question Answering

Progress on question answering (QA) has claimed human-level accuracy. However, most factoid QA models are trained and evaluated on clean text input, which becomes noisy when questions are spoken due to Automatic Speech Recognition (ASR) errors. This consideration is disregarded in trivia match-ups between machines and humans: IBM Watson (Ferrucci, 2010) on Jeopardy! and QB matches between machines and trivia masters (Boyd-Graber et al., 2018) provide text data for machines while humans listen. A fair test would subject both humans and machines to speech input.

Unfortunately, there are no large *spoken* corpora of factoid questions with which to train models; text-to-speech software can be used as a method for generating training data at scale for question answering models (Section 3.7). Although synthetic data is less realistic than true human-spoken questions it easier and cheaper

[1] Denis Peskov, Joe Barrow, Pedro Rodriguez, Graham Neubig, and Jordan Boyd-Graber. 2019. Mitigating noisy inputs for question answering. In Conference of the International Speech Communication Association

Peskov is responsible for the data creation, the gathering of users from users, running the neural models, figure and table design, and majority of paper writing.

[2] Ahmed Elgohary, Denis Peskov, and Jordan Boyd-Graber. 2019. Can You Unpack That? Learning to Rewrite Questions-in-Context. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. 5920–5926

Peskov is responsible for manual quality control in the data collection process, analysis of the data and model predictions, part of paper writing, and figure+table design.

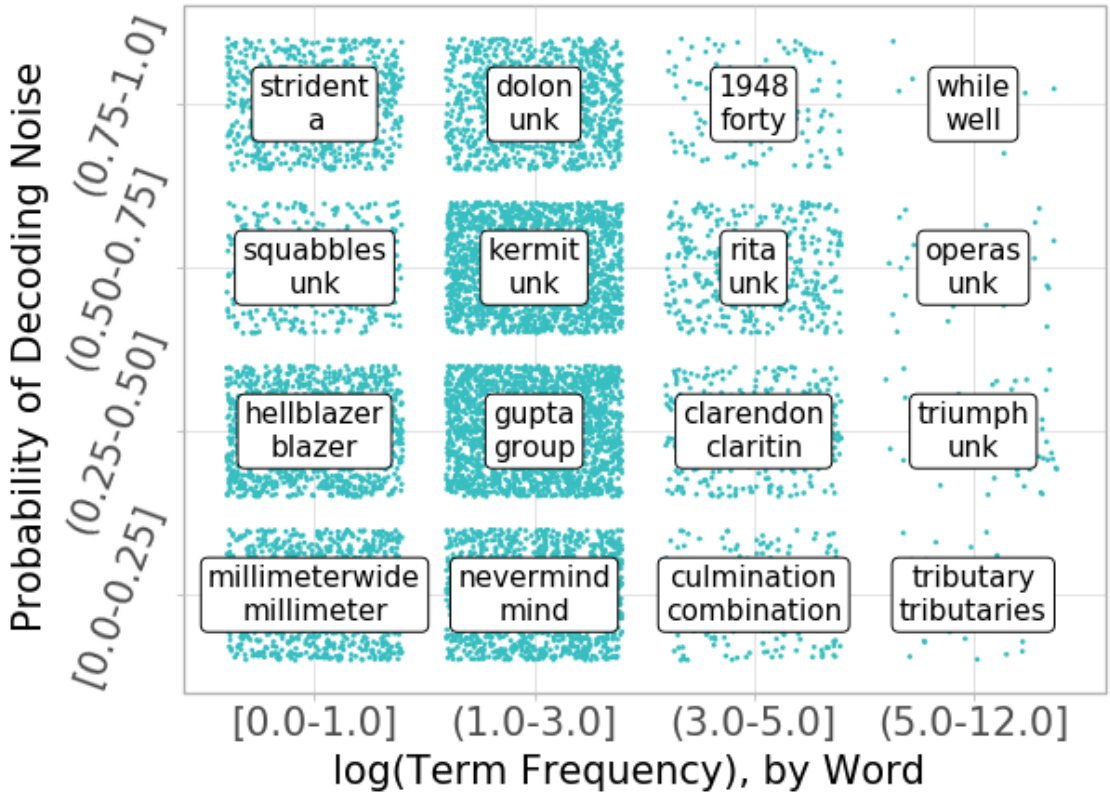

Figure 3.1: ASR errors on QA data: original spoken words (top of box) are garbled (bottom). While many words become into "noise"—frequent words or the unknown token—consistent errors (e.g., "clarendon" to "clarintin") can help downstream systems. Additionally, words reduced to <*unk*> (e.g., "kermit") can be useful through forced decoding into the closest incorrect word (e.g., "hermit" or even "car").

to collect at scale, which is important for training. These synthetic data are still useful; in Section 3.4.1, models trained on synthetic data are applied to human spoken data from QB tournaments and Jeopardy!

Noisy ASR is particularly challenging for QA systems (Figure 3.1). While humans and computers might know the title of a "revenge novel centering on Edmund Dantes by Alexandre Dumas", transcription errors may mean deciphering "novel centering on edmond dance by alexander <*unk*>" instead. Dantes and Dumas are low-frequency words in the English language and hence likely to be misinterpreted by a generic ASR model; however, they are particularly important for answering the question. Additionally, the introduction of distracting words (e.g., "dance") causes QA models to make errors (Jia and Liang, 2017). Section 3.2.1 characterizes the signal in this noise: key terms like named entities are often missing, which is detrimental for QA.

Previous approaches to mitigate ASR noise for answering mobile queries (Mishra and Bangalore, 2010) or building bots (Leuski et al., 2009) typically use unsupervised methods, such as term-based information retrieval. Our datasets for training and

evaluation can produce *supervised* systems that directly answer spoken questions. Machine translation (Sperber et al., 2017) also uses ASR confidences; we evaluate similar methods on QA.

Specifically, some accuracy loss from noisy inputs can be mitigated through a combination of forcing unknown words to be decoded as the closest option (Section 3.3.2), and incorporating the uncertainties of the ASR model directly in neural models (Section 3.3.3). The forced decoding method reconstructs missing terms by using terms audibly similar to the transcribed input. Word-level confidence scores incorporate uncertainty from the ASR system into a Deep Averaging Network, introduced earlier in Background Section. Section 3.4 compares these methods against baseline methods on our synthetic and human speech datasets for Jeopardy! and QB.

3.2 Spoken question answering datasets

Neural networks require a large training corpus, but recording hundreds of thousands of questions is not feasible. Crowd-sourcing with the required quality control (speakers who say "cyclohexane" correctly) is expensive. As an alternative, we generate a data-set with Google Text-to-Speech on 96,000 factoid questions from a trivia game called QB (Boyd-Graber et al., 2018), each with 4–6 sentences for a total of over 500,000 sentences.[3] We then decode these utterances using the Kaldi chain model (Peddinti et al., 2015), trained on the Fischer-English dataset (Cieri et al., 2004) for consistency with past results on mitigating ASR errors in MT (Sperber et al., 2017). This model has a Word Error Rate (WER) of 15.60% on the eval2000 test set. The WER increases to 51.76% on our QB data, which contains out of domain vocabulary. The most BLEU improvement in machine translation under noisy conditions could be found in this middle WER range, rather than in values below 20% or above 80% (Sperber et al., 2017). Retraining the model on the QB domain would mitigate this noise; however, in practice one is often at the mercy of a pre-trained recognition model due to changes in vocabularies or speakers. Intentional noise has been added to machine translation data (Michel and Neubig, 2018; Belinkov and Bisk, 2018). Alternate methods for collecting large scale audio data include Generative Adversarial Networks (Donahue et al., 2018) and manual recording (Lee et al., 2018).

The task of QA requires the system to provide a correct answer out of many candidates based on the question's wording. We test on two varieties of different length and framing. QB questions, which are generally four to six sentences long, test a user's depth of knowledge; early clues are challenging and obscure but they progressively become easy and well-known. Competitors can answer these types of questions at any point. Computer QA is competitive with the top players (Yamada et al., 2018). Jeopardy! questions are single sentences and can only be answered after the question ends. To test this alternate syntax, we use the same method of data generation on a dataset of over 200,000 Jeopardy questions (Dunn et al.,

[3] `http://cloud.google.com/text-to-speech`

2017b).

3.2.1 Why QA is challenging for ASR

ASR changes the features of the recognized text in several important ways: the overall vocabulary is quite different and important words are corrupted. First, it reduces the overall vocabulary. In our dataset, the vocab drops from 263,271 in the original data to a mere 33,333. This is expected, as ASR only has 42,000 words in its vocab, so the long tail of the Zipf's curve is lost. Second, unique words—which may be central to answering the question—are lost or misinterpreted; over 100,000 of the words in the original data occur only once. Finally, ASR systems tend to delete unintentionally delete words, which makes the sentences shorter. In our QB data, the average number of words decreases from 21.62 to 18.85 per sentence.

The decoding system is able to express uncertainty by predicting $<unk>$. These account for slightly less than 10% of all our word tokens, but is a top-2 prediction for 30% of the 260,000 original words. For QA, words with a high TF-IDF measure are valuable. While some words are lost, others can likely be recovered: "hellblazer' becoming "blazer", "clarendon" becoming "claritin". We evaluate this by fitting a TF-IDF model on the Wikipedia dataset and then comparing the average TF-IDF per sentence between the original and the ASR data. The average TF-IDF score, the most popular metric for evaluating how important a word is for a document, drops from 3.52 to 2.77 per sentence.

3.3 Mitigating noise

This section discusses two approaches to mitigating the effects of missing and corrupted information caused by ASR systems. The first approach—forced decoding—exploits systematic errors to arrive at the correct answer. The second uses confidence information from the ASR system to down-weight the influence of low-confidence terms. Both approaches improve accuracy over a baseline DAN model and show promise for short single-sentence questions. However, a IR approach is more effective on long questions since noisy words are completely avoided during the answer selection process.

3.3.1 IR baseline

The IR baseline reframes Jeopardy! and QB QA tasks as document retrieval ones with an inverted search index. We create one document per distinct answer; each document has a text field formed by concatenating all questions with that answer together. At test time questions are treated as queries, and documents are scored using BM25 (Ramos, 2003; Robertson et al., 2009). We implement this baseline with Elastic Search and Apache Lucene.

Table 3.1: As original data are translated through ASR, it degrades in quality. One-best output captures per-word confidence. Full lattices provide additional words and phone data captures the raw ASR sounds. Our confidence model and forced decoding approach could be used for such data.

Clean	For 10 points, name this revenge novel centering on Edmond Dantes, written by Alexandre Dumas
1-Best	for$^{0.935}$ ten$^{0.935}$ points$^{0.871}$ same$^{0.617}$ this^{1} ... revenge novel centering on <*unk*> written by alexander <*unk*> ...
"Lattice"	for$^{0.935}$ [eps]$^{0.064}$ pretend$^{0.001}$ ten$^{0.935}$ ... pretend point points point name same named name names this revenge novel ...
Phones	f_B$^{0.935}$ er_E$^{0.935}$ t_B$^{0.935}$ eh_I^{1} n_E$^{0.935}$ ... p_B oy_I n_I t_I s_E sil s_B ey_I m_E dh_B ih_I s_E r_B iy_I v_I eh_I n_I jh_E n_B aa_I v_I ah_I l_I ...

3.3.2 Forced decoding

We have systematically lost information. We could predict the answer if we had access to certain words in the original question and further postulate that wrong guesses are better than knowing that a word is unknown.

We explore commercial solutions—Bing, Google, IBM, Wit—with low transcription errors. However, their APIs ensure that an end-user often cannot extract anything more than one-best transcriptions, along with an aggregate confidence for the sentence. Additionally, the proprietary systems are moving targets, harming reproducibility.

We use Kaldi (Povey et al., 2011) for all experiments. Kaldi is a commonly-used, open-source tool for ASR; its maximal transparency enables approaches that incorporate uncertainty into downstream models. Kaldi provides not only top-1 predictions, but also confidences of words, entire lattices, and phones (Table 3.1). Confidences are the same length as the text, range from 0.0 to 1.0 in value, and correspond to the respective word or phone in the sequence.

The typical end-use of an ASR system wants to know when when a word is not recognized. By default, a graph will have a token that represents an unknown; in Kaldi, this becomes <*unk*>. At a human-level, one would want to know that an out of context word happened.

However, when the end-user is a downstream model, a systematically wrong prediction may be better than a generic statement of uncertainty. So by removing all reference to <*unk*> in the model's Finite State Transducer, we force the system to decode "Louis Vampas" as "Louisiana" rather than <*unk*>. The risk we run with this method is introducing words not present in the original data. For example, "count" and "mount" are similar in sound but not in context embeddings. Hence, we need a method to downweight incorrect decoding.

3.3.3 Confidence augmented DAN

We build on Deep Averaging Networks (Iyyer et al., 2015, DAN), assuming that deep bag-of-words models can improve predictions and be robust to corrupted phrases. The errors introduced by ASR can hinder sequence neural models as key phrases are potentially corrupted and syntactic information is lost.

We modify the original DAN model, introduced in Background Section 3.1, to use word-level confidences from the ASR system as a feature. In increasing order of complexity, the variations are: a Confidence Informed Softmax DAN, a Confidence Weighted Average DAN, and a Word-Level Confidence DAN. We represent the confidences as a vector $\mathbf{c}$, where each cell c_{i} contains the ASR confidence of word w_{i}.

The simplest model averages the confidence across the whole sentence and adds it as a feature to the final output classifier. For example in Table 3.1, "for ten points" averages to 0.914. We introduce an additional weight in the output $\mathbf{W^c}$, which adjusts our prediction based on the average confidence of each word in the question.

However, most words have high confidence, and thus the average confidence of a sentence or question level is high. To focus on which words are uncertain we weight the word embeddings by their confidence attenuating uncertain words before calculating the DAN average.

Weighting by the confidence directly removes uncertain words, but this is too blunt an instrument, and could end up erasing useful information contained in low-confidence words, so we instead learn a function based on the raw confidence from our ASR system. Thus, we recalibrate the confidence through a learned function f:

$$f(\mathbf{c}) = \mathbf{W^{(c)}c} + \mathbf{b^{(c)}} \tag{3.1}$$

and then use that scalar in the weighted mean of the DAN representation layer:

$$\mathbf{r^{**}} = \frac{\sum_i^N \mathbf{E}[w_{\text{i}}] * f(c_{\text{i}})}{N}. \tag{3.2}$$

In this model, we replace the original encoder $\mathbf{r}$ with the new version $\mathbf{r^{**}}$ to learn a transformation of the ASR confidence that down-weights uncertain words and up-weights certain words. This final model is referred to as our "Confidence Model".

Architectural decisions are determined by hyperparameter sweeps. They include: having a single hidden layer of 1000 dimensionality for the DAN, multiple drop-out, batch-norm layers, and a scheduled ADAM optimizer. Our DAN models train until convergence, as determined by early-stopping. Code is implemented in PyTorch (Paszke et al., 2017), with TorchText for batching.[4]

[4]Code, data, and additional analysis available at `https://github.com/DenisPeskov/QBASR`

3.4 Results

Achieving 100% accuracy on this dataset is not a realistic goal, as not all test questions are answerable (specifically, some answers do not occur in the training data and hence cannot be learned by a machine learning system). Baselines for the DAN (Table 3.2) establish realistic goals: a DAN trained and evaluated on the *same train and dev set*, only in the original non-ASR form, correctly predicts 54% of the answers. Noise drops this to 44% with the best IR model and down to $\approx 30\%$ with neural approaches.

Since the noisy data quality makes full recovery unlikely, we view any improvement over the neural model baselines as recovering valuable information. At the question-level, strong IR outperforms the DAN by around 10%.

Since IR can avoid all the noise while benefiting from additional independent data points, it scales as the length of data increases. There is additional motivation to investigate this task at the sentence-level. Computers can beat humans at the game by knowing certain questions immediately; the first sentence of the QB question serves as a proxy for this threshold. Our proposed combination of forced decoding with a neural model led to the highest test accuracy results and outperforms the IR one at the sentence level.

A strong TF-IDF IR model can top the best neural model at the multi-sentence question level in QB; multiple sentences are important because they progressively become easier to answer in competitions. However, our models improve accuracy on the shorter first-sentence level of the question. This behavior is expected since IR methods are explicitly designed to disregard noise and can pinpoint the handful of unique words in a long paragraph; conversely they are less accurate when they extract words from a single sentence.

3.4.1 Qualitative Analysis & Human Data

The synthetic dataset facilitates large-scale machine learning, but ultimately we care about performance on human data. For QB we record questions read by domain experts at a competition. To account for variation in speech, we record five questions across ten different speakers, varying in gender and age; this set of fifty questions is used as the human test data. Table 3.3 provides examples of variations. For Jeopardy! we manually parsed a complete episode by question.

The predictions of the regular DAN and the confidence version can differ. For input about The House on Mango Street, which contains words like "novel", "character", and "childhood" alongside a corrupted name of the author, the regular DAN predicts The Prime of Miss Jean Brodie, while our version predicts the correct answer.

3.4.2 Discussion & Future Work

Confidences are a readily human-interpretable concept that may help build trust in the output of a system. Transparency in the quality of up-stream content

	QB				Jeopardy!	
	Synth		Human		Synth	Human
Model	Start	End	Start	End		
Methods Tested on Clean Data						
IR	0.064	0.544	0.400	1.000	0.190	0.050
DAN	0.080	0.540	0.200	1.000	0.236	0.033
Methods Tested on Corrupted Data						
IR base	0.021	0.442	0.180	0.560	0.079	0.050
DAN	0.035	0.335	0.120	0.440	0.097	0.017
FD	0.032	0.354	0.120	0.440	0.102	0.033
Confidence	0.036	0.374	0.120	0.460	0.095	0.033
FD+Conf	0.041	0.371	0.160	0.440	0.109	0.033

Table 3.2: Both forced decoding (FD) and the best confidence model improve accuracy. Jeopardy only has an At-End-of-Sentence metric, as questions are one sentence in length. Combining the two methods leads to a further joint improvement in certain cases. IR and DAN models trained and evaluated on clean data are provided as a reference point for the ASR data.

Speaker	Text
Base	John Deydras, an insane man who claimed to be Edward II, stirred up trouble when he seized this city's Beaumont Palace.
S1	unk an insane man who claimed to be the second unk trouble when he sees unk beaumont → Richard_I_of_England
S2	john dangerous insane man who claims to be the second stirring up trouble when he sees the city's beaumont → London
S3	unk dangerous insane man who claim to be unk second third of trouble when he sees the city's unk palace → Baghdad

Table 3.3: Variation in different speakers causes different transcriptions of a question on Oxford. The omission or corruption of certain named entities leads to different predictions, which are indicated with an arrow.

can lead to downstream improvements in a plethora of NLP tasks.

Exploring sequence models or alternate data representations may lead to further improvement. Including full lattices may mirror past results for machine translation (Sperber et al., 2017) for the task of question answering. Phone-level approaches work in Chinese (Lee et al., 2018), but our phone models had lower accuracies than the baseline, perhaps due to a lack of contextual representation. Using unsupervised approaches for ASR (Wessel and Ney, 2004; Lee et al., 2009) and training ASR models for decoding QB or Jeopardy! words are avenues for further exploration.

3.5 Can Question Answering Audio be Automated?

Question answering, like many NLP tasks are impaired by noisy inputs. Introducing ASR into a QA pipeline corrupts the data. A neural model that uses the ASR system's confidence outputs and systematic forced decoding of words rather than unknowns improves QA accuracy on QB and Jeopardy! questions. Our methods are task agnostic and can be applied to other supervised NLP tasks. Larger *human-recorded* question datasets and alternate model approaches would ensure spoken questions are answered accurately, allowing human and computer trivia players to compete on an equal playing field. Text-to-Speech technology can create a large dataset, but the unvarying pronunciation, speed, and voice—every single TTS voice is female—ultimately inhibits this approach from being a gold-standard.

3.6 Crowd-Sourcing for Question Generation

Some tasks cannot be automatically generated from templates and require human discretion. One cost-efficient, scalable pool for human input are crowd-sourcing platforms, specifically Mechanical Turk (Buhrmester et al., 2011). We summarize a data collection project that used unspecialized workers to rewrite trivia questions.

Question Answering (QA) is an AI complete problem (Webber, 1992), but existing QA datasets do not rise to the challenge: they lack key NLP problems like anaphora resolution, coreference disambiguation, and ellipsis resolution. The logic needed to answer these types of questions requires deeper NLP understanding that simulates the context in which humans naturally answer questions.

Background Section 2.2.2 distinguishes between machine reading comprehension (MRC) and the nascent area of conversational question answering (CQA). However, we observe that CQA questions can be rewritten as stand-alone MRC questions and provide additional training data. We reduce challenging, interconnected CQA examples to independent, stand-alone MRC to create CANARD—**C**ontext **A**bstraction: **N**ecessary **A**dditional **R**ewritten **D**iscourse—a new dataset[5] that rewrites QUAC (Choi et al., 2018) questions. We crowd-source context-independent paraphrases of QUAC questions and use the paraphrases to train and evaluate question-in-context rewriting. In the process, we observe the behavior of crowd users and the quality of their output.

Section 3.7 constructs CANARD, a new dataset of question-in-context with corresponding context-independent paraphrases. Section 6.5 analyzes our rewrites (and the underlying methodology) to understand the linguistic phenomena that make CQA and using crowd-sourcing for *generation* difficult.

[5] `http://canard.qanta.org`

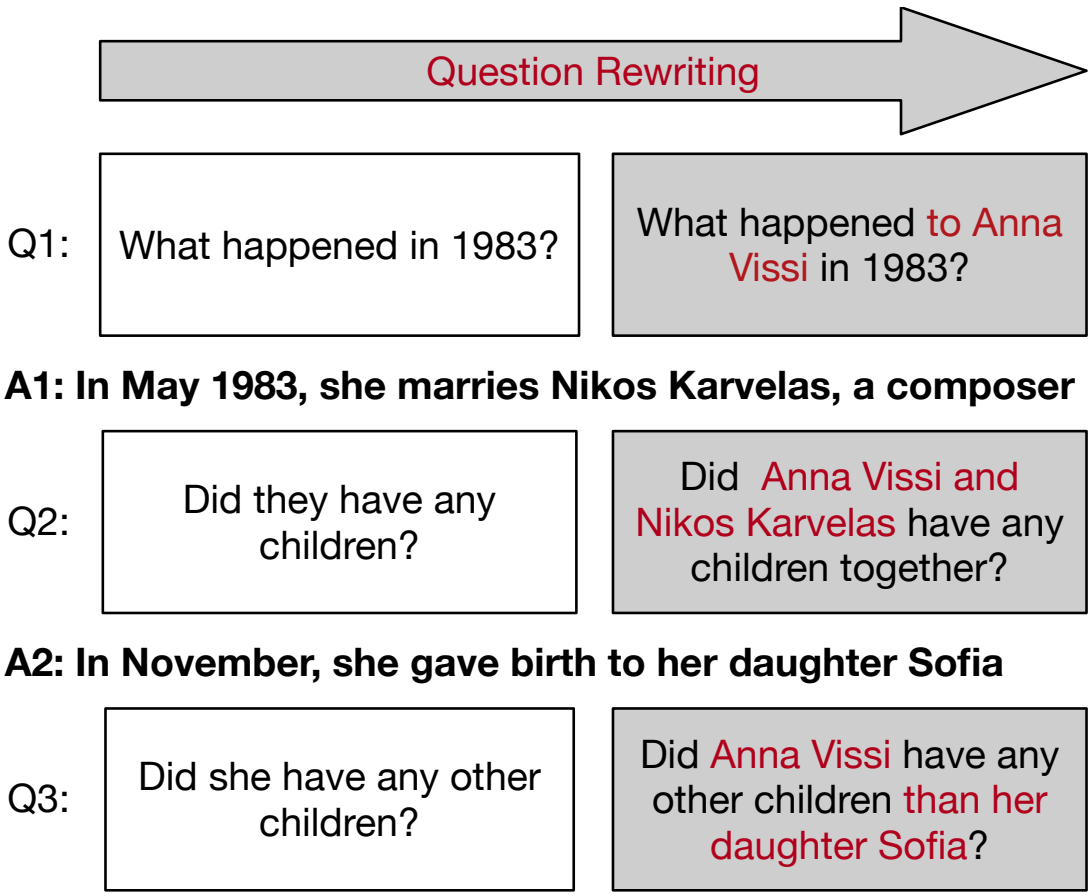

Figure 3.2: Question-in-context rewriting task. The input to each step is a question to rewrite given the dialog history which consists of the dialog utterances (questions and answers) produced before the given question is asked. The output is an equivalent, context-independent paraphrase of the input question.

3.7 Dataset Construction

We elicit paraphrases from human crowdworkers to make previously context-dependent questions *unambiguously* answerable. Through this process, we resolve difficult coreference linkages and create a pair-wise mapping between ambiguous and context-enriched questions. We derive CANARD from QUAC (Choi et al., 2018), a sequential question answering dataset about specific Wikipedia sections. QUAC uses a pair of workers—a "student" and a "teacher"—to ask and respond to questions. The "student" asks questions about a topic based on only the title of the Wikpedia article and the title of the target section. The "teacher" has access to the full Wikipedia section and provides answers by selecting text that answers the question. With this methodology, QUAC gathers 98k questions across 13,594 conversations. We take their entire dev set and a sample of their train set and create a custom JavaScript

Characteristic	Ratio
Answer Not Referenced	0.98
Question Meaning Unchanged	0.95
Correct Coreferences	1.0
Grammatical English	1.0
Understandable w/o Context	0.90

Table 3.4: Manual inspection of 50 rewritten context-independent questions from CANARD suggests that the new questions have enough context to be independently understandable.

task in Mechanical Turk that allows workers to rewrite these questions. JavaScript hints help train the users and provides automated, real-time feedback.

We provide workers with a comprehensive set of instructions and task examples. We ask them to rewrite the questions in natural sounding English while preserving the sentence structure of the original question. We discourage workers from introducing new words that are unmentioned in the previous utterances and ask them to copy phrases when appropriate from the original question. These instructions ensure that the rewrites only resolve conversation-dependent ambiguities. Thus, we encourage workers to create minimal edits; in Section 6.4, we take advantage of this to use BLEU for evaluating model-generated rewrites.

We display the questions in the conversation one at a time, since the rewrites should include only the previous utterance. After a rewrite to the question is submitted, the answer to the question is displayed. The next question is then displayed. This repeats until the end of the conversation. The full set of instructions and the data collection interface are provided in the appendix.

We apply quality control throughout our collection process. During the task, JavaScript checks automatically monitor and warn about common errors: submissions that are abnormally short (e.g., 'why'), rewrites that still have pronouns (e.g., 'he wrote this album'), or ambiguous words (e.g., 'this article', 'that'). Many QUAC questions ask about 'what/who else' or ask for 'other' or 'another' entity. For that class of questions, we ask workers to use a phrase such as 'other than', 'in addition to', 'aside from', 'besides', 'together with' or 'along with' with the appropriate context in their rewrite.

We gather and review our data in batches to screen potentially compromised data or low quality workers. A post-processing script flags suspicious rewrites and workers who take and abnormally long or short time. We flag about 15% of our data. *Every* flagged question is manually reviewed by one of the authors and an entire HIT is discarded if one is deemed inadequate. We reject 19.9% of submissions and the rest comprise CANARD. Additionally, we filter out under-performing workers based on these rejections from subsequent batches. To minimize risk, we limit the initial pool of workers to those that have completed 500 HITs with over 90% accuracy and offer competitive payment of $0.50 per HIT.

We verify the efficacy of our quality control through manual review. A ran-

ORIGINAL: Was this an honest mistake by the media?
REWRITE: Was the claim of media regarding Leblanc's room come to true?
ORIGINAL: What was a single from their album?
REWRITE: What was a single from horslips' *album*?
ORIGINAL: Did they marry?
REWRITE: Did Hannah Arendt and Heidegger marry?

Table 3.5: Not all rewrites correctly encode the context required to answer a question. We take two failures to provide examples of the two common issues: Changed Meaning (top) and Needs Context (middle). We provide an example with no issues (bottom) for comparison.

dom sample of fifty questions sampled from the final dataset is reviewed for desirable characteristics by a native English speaker in Table 3.4. Each of the positive traits occurs in 90% or more of the questions. Based on our sample, our edits retain grammaticality, leave the question meaning unchanged, and use pronouns unambiguously. There are rare occasions where workers use a part of the answer to the question being rewritten or where some of the context is left ambiguous. These infrequent mistakes should not affect our models. We provide examples of failures in Table 3.5.

We use the rewrites of QUAC's development set as our test set (5,571 question-in-context and corresponding rewrite pairs) and use a 10% sample of QUAC's training set rewrites as our development set (3,418); the rest are training data (31,538).

3.8 Dataset and Model Analysis

We analyze our dataset with automatic metrics after validating the reliability of our data (Section 3.7). We compare our dataset to the original QUAC questions and to automatically generated questions by our models. Then, we manually inspect the sources of rewriting errors in the seq2seq baseline.

3.8.1 Anaphora Resolution and Coreference

Our rewrites are longer, contain more nouns and less pronouns, and have more word types than the original data. Machine output lies in between the two human-generated corpora, but quality is difficult to assess. Figure 3.3 shows these statistics. We motivate our rewrites by exploring linguistic properties of our data. Anaphora resolution and coreference are two core NLP tasks applicable to this dataset.

Pronouns occur in 53.9% of QUAC questions. Questions with pronouns are more likely to be ambiguous than those without any. Only 0.9% of these have pronouns that span more than one category (e.g., 'she' and 'his'). Hence, pronouns within a single sentence are likely unambiguous. However, 75.0% of the aggregate history has pronouns and the percentage of mixed category pronouns increase to 27.8% of our data. Therefore, pronoun disambiguation potentially becomes a prob-

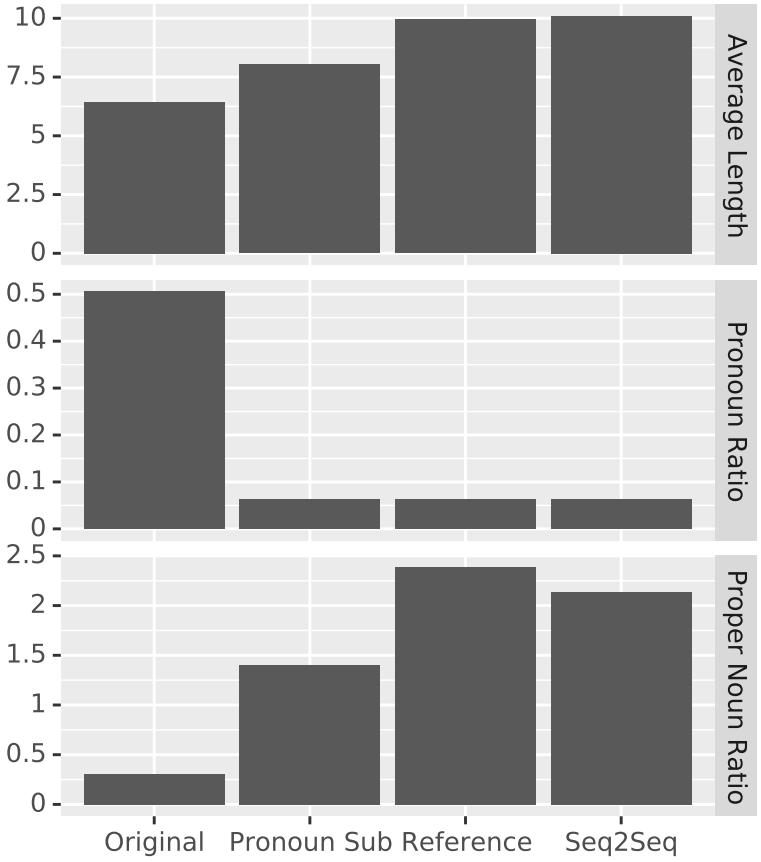

Figure 3.3: Human rewrites are longer, have fewer pronouns, and have more proper nouns than the original QUAC questions. Rewrites are longer and contain more proper nouns than our Pronoun Sub baseline and trained Seq2Seq model.

lem for a quarter of the original data. An example is provided in Table 3.6.

Approximately one-third of the questions generated by our pronoun-replacement baseline are within 85% string similarity to our rewritten questions. That leaves two-thirds of our data that cannot be solved with pronoun resolution alone.

3.9 Conclusion

In this chapter, we cover two types of low-cost dataset construction techniques: automation and generalist crowd-sourcing. The advantages of this method are cost and scalability, which is demanded by the current paradigm of neural models. This however comes at the expense of quality. A limitation of our past work in *automation* is generalization: text-to-speech only has female voices and is consistently decoded, while the voices of real humans are decoded with large variations. Unseen data points are likely to confound a model trained on unnatural data. Additionally, automated data creation still depends on having quality source data, that often has to come from expert users. In this project, we are able to record *found* questions

Label	Text
QUESTION	How long did he stay there?
REWRITE	How long did Cito Gaston stay at the Jays?
HISTORY	*Cito Gaston* **Q:** What did Gaston do after the world series? . . . **Q:** Where did he go in 2001? **A:** In 2002, he was hired by the Jays as special assistant to president and chief executive officer Paul Godfrey.

Table 3.6: An example that had over ten flagged proper nouns in the history. Rewriting requires resolving challenging coreferences.

that were already written by Quizbowl experts. Writing hundreds of thousands of our questions would not have been tractable. This suggests that expert design is necessary for automation, as discussed in Chapter 5.

A limitation of generalist *crowd-sourcing* is the inability to automatically quality control *generated* data. Our work requires *manual* analysis of each sentence submitted by the crowd; this is time-intensive and subject to error. Additionally, it requires real-time task monitoring and user exclusion as otherwise malicious users can quickly contribute a large part of your crowd-sourced task. There is no full-proof way to ensure quality in tasks involving crowd-sourcing *generation*. However, this method seems to generate more diverse and lengthy sentences than a comparable automation technique. One way to handle the quality control issue is by using an expert for quality assessment, which we discuss in Chapter **??**.

Chapter 4: Mixed Types of Users

As a dovetail between crowd-driven and expert-driven data sources, we propose an intermediate solution that pairs a person from the crowd with an expert. This reflects the attitude of a customer, simulated by a worker from the crowd, interacting with a customer service agent, simulated by an actual professional customer service agent. The resulting dataset provides an stark contrast in the language generated by anonymous crowd workers and experts.[1]

4.1 Introduction

Modern Natural Language Understanding (NLU) frameworks for dialogues are by definition data hungry. They require large amounts of training data representative of goal oriented conversations reflecting both context and diversity. But human responses in goal-oriented dialogues are less predictable than automated systems (Bordes et al., 2016). For example, "Please do this" cannot be interpreted without a broader context. Only by seeing previous utterances, such as requests to book a flight on a specific day to a specific destination, can this task be performed. Additionally, a single intent can be phrased in multiple ways depending on context; "book my flight", "finalize my reservation", "Yes, the 6 pm one" may all be referring to a flight-booking intent. Hence, entire conversations, rather than independent utterances, must be collected. Such data is even more pertinent to modeling NLU and related tasks as they require large, varied, and ideally human-generated datasets. Moreover, recent work (Dong et al., 2015; Devlin et al., 2019) has shown the benefit of applying joint-training and transfer learning techniques to natural language processing tasks. However, these approaches have yet to become widely used in dialogue tasks, due to a lack of large-scale datasets. Furthermore, the latest state of the art end-to-end neural approaches benefit from such training data even more so than past work on goal-oriented dialogues structured around slot filling (Lemon et al., 2006; **?**). One way to simulate data—and not risk releasing personally identifying information—for a domain is to use a Wizard-of-Oz data gathering tech-

[1]Denis Peskov, Nancy Clarke, Jason Krone, Brigi Fodor, Yi Zhang,Adel Youssef, and Mona Diab. Multi-domain goal-oriented dialogues(multidogo): Strategies toward curating and annotating large scale dialogue data. In Proceedings of the 2019 Conference on Empir-ical Methods in Natural Language Processing and the 9th Interna-tional Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 4518–4528, 2019.
Peskov planned and implemented majority of crowd-sourcing tasks, supervised the data collection thereof, wrote part of the task guidelines, performed data analysis, and wrote the majority of paper.

Role	Turn	Annotations
A	Hey there! Good morning. You're connected to LMT Airways. How may I help you?	DA = { elicitgoal }
C	Hi, I wonder if you can confirm my seat assignment on my flight tomorrow?	IC = { SeatAssignment }
A	Sure! I'd be glad to help you with that. May I know your last name please?	DA = { elicitslot }
C	My last name is Turker.	IC = { contentonly }, SL = {Name : Turker }
A	Alright Turker! Could you please share the booking confirmation number?	DA = { elicitslot }
C	I believe it's AMZ685.	IC = { contentonly }, SL = { Confirmation Number : AMZ685 }
...	...	...

Table 4.1: A segment of a dialogue from the airline domain annotated at the turn level. This data is annotated with agent dialogue acts (DA), customer intent classes (IC), and slot labels (SL). Roles C and A stand for "Customer" and "Agent", respectively.

nique, which requires that participants in a conversation fulfill a role (Kelley, 1984). This approach has been used in popular public goal-oriented datasets: DSTC and `MultiWOZ` (Williams et al., 2016; Budzianowski et al., 2018).

Conversations between people and automated systems occur with increasing frequency, especially in customer service. Customers reach out to agents, which could be automated bots or real individuals, to achieve a domain-specific goal. This creates a disparate conversation: agents are incentivized to operate within a set procedure and convey a patient and professional tone. In contrast, customers do not have this incentive. However, to date, the largest available multi-domain goal-oriented dialogue dataset assigns similar dialogue act annotations to both agents and customers (Budzianowski et al., 2018).

To solve the aforementioned challenges, we present our efforts to curate, annotate, and evaluate a large scale multi-domain set of goal oriented dialogues. The dataset is primarily gathered from workers in the crowd paired with professional annotators. The dataset elicited, `MultiDoGO`, comprises over 86K raw conversations of which 54,818 conversations are annotated at the turn level. We investigate multiple levels of annotation granularity. We annotate a subset of the data on both turn and sentence levels. A turn is defined as a sequence of one or more speech/text sentences by a participant in a conversation. A sentence is a period delimited sequence of words in a turn. A turn may comprise one or more sentences. We do use the term utterance to refer to a unit (turn or sentence, spoken or written by a participant).[2] In our devised annotation strategy, we distinguish between dialogue speech acts for agents vs. customers. In `MultiDoGO`, the agents' speech acts [DA]

[2]We acknowledge that the term utterance is controversial in the literature (**?**)

are annotated with generic class labels common across all domains, while customer speech acts are labeled with intent classes [IC]. Moreover, we annotate customer utterances with the appropriate slot labels [SL], which consist of the SL span and corresponding tokens with that SL tag. We present the strategies we use to curate and annotate such data given its contextual setting. We furthermore illustrate the efficacy of our devised approaches and annotation decisions against intrinsic metrics and via extrinsic evaluation, namely by applying neural baselines for DA, IC and SL classification leveraging joint models.

4.2 Existing Dialogue Datasets

There are multiple existing goal-oriented dialogue collections generated by humans through Wizard-of-Oz techniques. The Dialog State Tracking Challenge, *aka* Dialog Systems Technology Challenge, (DSTC) spans 8 iterations and entails the domains of bus timetables, restaurant reservations, and hotel bookings, travel, alarms, movies, etc. (Williams et al., 2016). Frames (Asri et al., 2017) has 1369 dialogues about vacation packages. `MultiWOZ` contains 10,438 dialogues about Cambridge hotels and restaurants (Budzianowski et al., 2018). There are several dialogue datasets that specialize in a single domain. ATIS (Hemphill et al., 1990) comprises speech data about airlines structured around formal airline flight tables. Similarly, the Google Airlines dataset purportedly contains 400,000 templated dialogues about airline reservations (Wei et al., 2018).[3] The Ubuntu Dialogue Corpus has over a million dialogues about Ubuntu technical support (Lowe et al., 2015).

On the other hand, Chit-chat style dialogues without goals have been popular since ELIZA and have been investigated with neural techniques (**???**). However, these datasets cannot be used for modeling goal-oriented tasks. Related dialogue dataset collections used for Sequential Question Answering rely on dialogue to answer questions, but the task is notably different from our use case of modeling goal oriented conversational AI, hence leading to different evaluation considerations (**?**Choi et al., 2018).

4.3 `MultiDoGO` Dataset Curation

4.3.1 Data Collection Procedure

We employ both internal data associates, who we train, and crowd-sourced workers from Mechanical Turk (MTurkers) to generate conversational data using a Wizard-of-Oz approach. In each conversation, the data associates assumes the role of an agent while the MTurkers act as customers. In an effort to source competent MTurkers, we re-quire that each MTurker have a Human Intelligence Task (HIT) accuracy minimum of 90%, a location in the United States, and have completed a significant number of HITs in the past. To facilitate goal-oriented conversations between the customer and agent, we give each agent a prompt listing the supported

[3]The Google Airlines dataset has not been released to date.

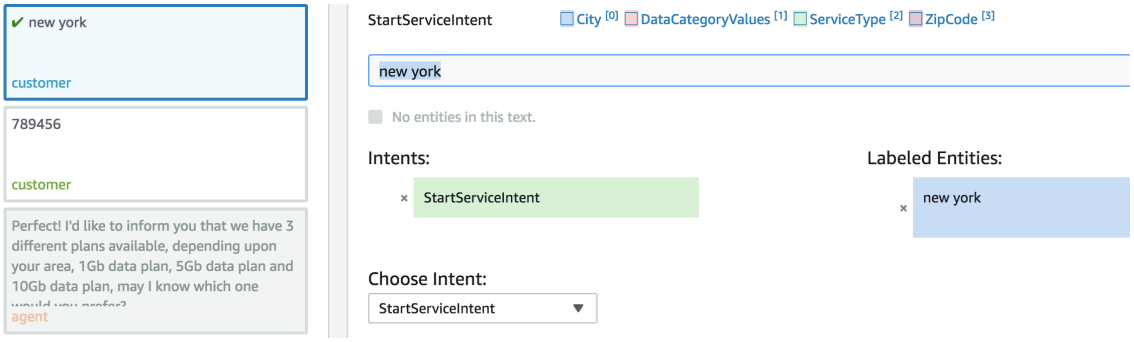

Figure 4.1: Crowd sourced annotators select an intent and choose a slot in our custom-built Mechanical Turk interface. Entire conversations are provided for reference. Detailed instructions are provided to users, but are not included in this figure. Options are unique per domain.

request types (dialog acts)and pieces of information (slots) needed to complete each request. We also specify criteria such as minimal conversation length, number of goals,number of complex requests, etc, to increase conversation diversity. See Figure 2 for an example prompt. In addition, we explicitly request that neither agents nor customers use any personally identifiable information. At an implementation level,we create a custom, web interface for the MTurkers and data associates that displays our instructions next to the current dialogue. This allows each participant to quickly refer to our guidelines with-out stopping the conversation. Despite following a familiar wizard-of-oz elicitation procedure, and curating data for multiple domains in a fashion similar to previous data collection efforts such as `MultiWOZ`, `MultiDoGO` comprises more varied domains, it is collected at an unprecedented scale, and it is curated with control over generating explicit biases in the conversations to allow for diverse conversation representation. To our knowledge this is a novel collection strategy as we explicitly guide/prod the participants in a dialogue to engage in conversations with specific biases such as intent change, slot change, multi-intent, multiple slot values, slot overfilling and slot deletion. For example, in the Fast Food domain, participants were instructed to pretend that they were ordering fast food from a drive-thru. After making their initial order, they were instructed to change their mind about what they were ordering:"I'd like a burger. No wait, can you make that a chicken sandwich?". In the Financial domain, we asked participants to make sure that they requested multiple intents such as "I'd like to find my routing number and check my balance."[4] To that end, our collection procedure deliberately attempts to guide the dialogue flow to ensure diversity in dialogue policies.

4.4 Data Annotation

We discuss the *annotation* needed for our dataset. Of particular interest, a direct comparison of using experts versus the crowd is made in in Section 7.2.

[4]For a full list of conversational biases with examples, please see the appendix.

4.4.1 Annotated Dialogue Tasks

Our dataset has three types of annotation: Agent dialogue acts [DA], customer intent classes [IC], and slot labels [SL]. We intentionally decouple Agent and customer speech act tags into the categories DA and IC, respectively, to produce more fine-grained speech act tags than past iterations of dialog datasets. Intuitively, agent DAs are consistent across domains and more abstract in nature, since agents have a standard form of response. On the other hand, customer ICs are domain-specific and can entail reserving a hotel room or ordering a burger, depending on the domain. A conversation example with annotations is provided in Table 4.1.

Agent Dialogue Acts (DA) Agent dialogue acts are the most straightforward of our annotation tasks. There are eight possible DAs in all domains: *ElicitGoal, ElicitSlot, ConfirmGoal, ConfirmSlot, EndGoal, Pleasantries, Other*. The names are self-explanatory. *Elicit Goal/Slot* indicates that the agent is gathering information. *Confirm Goal/Slot* indicates that the agent is confirming previously provided information. The *EndGoal* and *Pleasantries* tags, identify non-task related actions. *Other* indicates that the selected utterance was not one of the other possible tags. Agent dialogue acts are consistent across domains and are often abstract (e.g. ElicitIntent, ConfirmSlot).

Customer Intent Classes (IC): Unlike Agent DA, customer IC vary for each domain and are more concrete. For example, the Airline domain has a "BookFlight" IC, Fast Food has an "OrderMeal" IC, and Insurance has an "OrderPolicy" IC in our annotation schema. Customer intents can overlap across domains (e.g. OpeningGreeting, ClosingGreeting) and other times be domain specific (e.g. RequestCreditLimitIncrease, OrderBurger, BookFlight).

Slot Labels (SL): Slot Labeling is a task contingent on Customer Intent Classes. Certain intents require that additional information, namely slot values, be captured. For instance, to open a bank account, one must solicit the customer's social security number. Slots can overlap across intents (e.g. Name, SSN Number) or they can be unique to a domain-specific intent (e.g. CarPolicy).

4.4.2 Data Annotation Procedure

Our annotators use a web interface, depicted in Figure 4.1, to select the appropriate intent class for an utterance out of a list of provided options. To annotate slot labels, our annotators use their cursors to highlight slot value character spans within an utterance and then select the corresponding slot label from a list of options. The output of this slot labeling process is a list of ⟨slot-label, slot-value, span⟩ triplets for each utterance.

4.4.3 Annotation Design Decisions

Decoupled Agents and Customers Label Sets Agents and customers have notably different goals and styles of communication. However, past dialogue datasets do not make this distinction at speech act schema level. Specificity is important for

ISAA		
DA	IC	SL
0.701	0.728	0.695

Table 4.2: Dialogue act (DA), Intent class (IC), and slot labeling (SL) Inter Source Annotation Agreement (ISAA) scores quantifying the agreement of crowd sourced and professional annotations.

handling unique customer requests, but a relatively formulaic approach is required of agents across different industries. Our distinction between the customer and agent roles creates training data for a bot that explicitly simulates agents.

Annotation Unit Granularity: Sentence vs. Turn Level An important decision, which is often under discussed, is the proper semantic unit of text to annotate in a dialogue. Commonly, datasets provide annotations at the turn level (Budzianowski et al., 2018; Asri et al., 2017; **?**). However, turn level annotations can introduce confusion for IC datasets, given multiple intents may be present in different sentences of a single turn. For instance, consider the turn "I would like to book a flight to San Francisco. Also, I want to cancel a flight to Austin." Here, the first sentence has the BookFlight intent and the second sentence has the CancelFlight intent. An turn level annotation of this utterance would yield the multi-class intent (BookFlight, CancelFlight). In contrast, a sentence level annotation of this utterance identifies that the first sentence corresponds to BookFlight while the second corresponds to CancelFlight. We annotate a subset our data, 2,500 conversation per domain for 15,000 conversations in total, at the sentence as well as turn level to access the impact of this design choice on downstream performance. The remainder of our dataset is annotated only at the turn level.

Professional vs. Crowd-Sourced Workers for Annotation For annotation, we compare and contrast professional annotators to crowd sourced annotators on a subset of data. Professional annotators assign DA, IC, and SL tags to the 15,000 conversations annotated at both the turn and sentence level; statistics for these conversations are given in Table 4.7. In an effort to decrease annotation cost, we employ crowd source annotators via Mechanical Turk to label an additional 54,818 conversations rated as Good or Excellent quality during data collection. We provide statistics for this set of crowd annotated data in Table 4.3. To compare the quality of crowd sourced annotations against professional annotations, we use both strategies to annotate a shared subset of 8,450 conversations. We devise an Inter Source Annotation Agreement (ISAA) metric to quantify the agreement of these crowd sourced and professionally sourced annotations. ISAA is a relaxation of Cohen κ, intended to count partial agreement of multi-tag labels. ISAA defines two sets of tags, A and B, to be in agreement if there is at least one "shared" tag in both A and B. A and B reflect the majority labels agreed upon per source (professionals or crowd workers). Using ISAA we find that crowd sourced and professional annotations have a substantial degree of shared annotations. We report ISAA for the DA, IC, and SL tasks in Table 4.2.

Domain	Elicited	Good/Excellent	IC/SL	DA/IC/SL
Airline	15100	14205	7598	6287
Fast Food	9639	8674	7712	4507
Finance	8814	8160	8002	6704
Insurance	14262	13400	7799	7434
Media	33321	32231	19877	12891
Software	5562	4924	3830	2753
Total	**86698**	**81594**	**54818**	**40576**

Table 4.3: Total number of conversations per domain: raw conversations Elicited; Good/Excellent is the total number of conversations rated as such by the agent annotators; (IC/SL) is the number of conversations annotated for Intent Classes and Slot Labels only; (DA/IC/SL) is the total number of conversations annotated for Dialogue Acts, Intent Classes, and Slot Labels.

Bias	Airlines	Fast Food	Finance	Insurance	Media	Software
IntentChange		1443				
MultiIntent	2200	1913	1799	1061	607	2295
MultiValue		354				
Overfill			1486	2763		
SlotChange	4207	2011	2506	3321	570	2085
SlotDeletion		333				
Total	6407	6054	5791	7145	1177	4380

Table 4.4: Number of conversations per domain collected with specific biases. Fast Food had the maximum number of biases. MultiIntent and SlotChange are the most used biases.

4.4.4 Quality Control

We institute three processes to enforce data quality. During data collection, our data associates report on the quality of each conversation. Specifically, the data associates grade the conversation on a scale from “Unusable”, “Poor", “Good", to “Excellent". They were provided with guidelines to help decide on the chosen rating such as coherence, whether the dialogue achieved the purported goal, etc. To ensure high data quality we only utilize conversations with “Good" or “Excellent" ratings in subsequent annotation.

Secondly, each conversation is annotated at least twice. We resolve inconsistent annotations by selecting the annotation given by the majority of annotators per item. We calculate inter-annotator agreement with Fleiss κ and find “substantial agreement”, according to the metric. Our annotators must pass a qualification test as well as maintain an on-going level of accuracy in randomly distributed test questions throughout their annotation. Third, we pre-process our data to remove issues such as duplicate conversations and improperly entered slot value spans. We refer readers to our discussion of pre-processing in Section 4.5 for further detail.

Metric	DSTC 2	WOZ2.0	M2M	MULTIWOZ	MULTIDOGO
Number of Dialogues	1,612	600	1,500	8,438	40,576
Total Number of Turns	23,354	4,472	14,796	115,424	813,834
Total Number of Tokens	199,431	50,264	121,977	1,520,970	9,901,235
Avg. Turns per Dialog	14.49	7.45	9.86	15.91	20.06
Avg. Tokens Per Turn	8.54	11.24	8.24	13.18	12.16
Total Unique Tokens	986	2,142	1,008	24,071	70,003
Number of Unique Slots	8	4	14	25	73
Number of Slot Values	212	99	138	4,510	55,816
Number of Domains	1	1	1	7	6
Number of Tasks	1	1	2	2	3

Table 4.5: `MultiDoGO` is several times larger in nearly every dimension to the pertinent datasets as selected by Budzianowski et al. (2018). We provide counts for the training data, except for FRAMES, which does not have splits. Our number of unique tokens and slots can be attributed to us not relying on carrier phrases.

4.4.5 Dataset Characterization and Statistics

`MultiDoGO` dataset is more diverse by virtue of covering more domains, but more importantly, it is more controlled since it was curated rather than being scraped from existing data sources that are not necessarily synchronous (Ubuntu). Table 4.3 shows the statistics for `MultiDoGO` raw conversations harvested, rated as Excellent or Good, and annotated for DA, IC and SL. Table 4.4 shows the number of conversations per domain reflecting the specific biases used.

`MultiDoGO` is several orders of magnitude larger than comparable datasets as reflected in nearly every dimension: the number of conversations, the length of the conversation, the number of domains, and the diversity of the utterances used. Table 4.5 illustrates a comparative statistics to existing data sets.

Domain	#Conv	#Turn	#Turn/Conv	#Sentence	#Intent	#Slot
Airline	2,500	39,616	15.8 (15)	66,368	11	15
Fast Food	2,500	46,246	18.5 (18)	73,305	14	10
Finance	2,500	46,001	18.4 (18)	70,828	18	15
Insurance	2,500	41,220	16.5 (16)	67,657	10	9
Media	2,500	35,291	14.1 (14)	65,029	16	16
Software	2,500	40,093	16.0 (15)	70,268	16	15

Table 4.6: Data statistics by domain. Conversation length is shown in *average (median)* number of turns per conversation. Inter-annotator agreement (IAA) is measured with Fleiss' κ for the three annotation tasks: Agent DA (DA), Customer IC (IC), and Slot Labeling (SL).

We provide summary statistics for the subset of our data annotated at both turn and sentence granularity in Table 4.7. This describes the total size of the data per domain in number of conversations, turns, the unique number of intents and slots, and inter-annotator agreement (IAA) for both turn and sentence level annotations. It is worth observing that the DA annotations achieve a much higher

Domain	Turn-level IAA	Sentence-level IAA
Airline	0.514/0.808/0.802	0.670/0.788/0.771
Fast Food	0.314/0.700/0.624	0.598/0.725/0.607
Finance	0.521/0.827/0.772	0.700/0.735/0.714
Insurance	0.521/0.862/0.848	0.703/0.821/0.826
Media	0.499/0.812/0.725	0.678/0.802/0.758
Software	0.508/0.748/0.745	0.709/0.764/0.698

Table 4.7: Inter-annotator agreement (IAA) is measured with Fleiss' κ for the three annotation tasks: Agent DA (DA), Customer IC (IC), and Slot Labeling (SL).

Agent Instructions

Imagine you work at a bank. Customers may contact you about the following set of issues: checking account balances (checking or savings), transferring money between accounts, and closing accounts.

GOAL: Answer the customer's question(s) and complete their request(s).

For any request, you will need to collect at least the following information to be able to identify the customer: name, account PIN *or* last 4 digits of SSN.

For giving information on balances, or for closing accounts, you will also need the last 4 digits of the account number.

For transferring money, you will also need: last 4 digits of account to move from, last 4 digits of account to move to, and the sum of money to be transferred.

Your customer may ask you to do only one thing; that's okay, but make sure you confirm you achieved everything the Customer wanted before completing the conversation. Don't forget to signal the end of the conversation (see General guidelines)

Figure 4.2: Agents are provided with explicit fulfillment instructions. These are quick-reference instructions for the Finance domain. Agents serve as one level of quality control by evaluating a conversation between Excellent and Unusable.

IAA in Sentence level annotations compared to Turn level annotation, most notably in the Fast Food domain. IC and SL annotations reflect a slightly higher IAA in Turn level annotation granularity compared to Sentence level.

4.5 Dialogue Classification Baselines

To establish baseline performance for the `MultiDoGO` dataset we pre-process, create dataset splits, and evaluate the performance of three baseline models for each domain.

Pre-processing: We pre-process the corpus of dialogues for each domain to

		Airline			Fast Food			Finance		
Model	Annot	DA	IC	SL	DA	IC	SL	DA	IC	SL
MFC	S	60.57	33.69	38.71	57.14	25.42	61.92	51.73	37.37	34.07
LSTM	S	97.20	90.84	74.16	90.40	86.09	72.93	93.90	90.06	69.09
ELMO	S	**97.32**	**91.88**	**86.55**	**91.03**	**87.95**	**77.51**	**94.07**	**91.15**	**77.36**
MFC	T	33.04	32.79	37.73	33.07	25.33	61.84	36.52	38.16	34.31
LSTM	T	**84.25**	89.15	75.78	**66.41**	87.35	73.57	76.19	92.30	70.92
ELMO	T	84.04	**89.99**	**85.64**	65.69	**88.96**	**79.63**	**76.29**	**94.50**	**79.47**
		Insurance			Media			Software		
Model	Annot	DA	IC	SL	DA	IC	SL	DA	IC	SL
MFC	S	56.87	38.37	53.75	57.02	30.42	82.06	58.14	33.32	53.96
LSTM	S	**94.73**	93.30	75.27	**94.27**	92.35	90.84	93.22	90.95	69.48
ELMO	S	94.63	**94.27**	**88.45**	**94.27**	**93.32**	**93.99**	**93.66**	**92.25**	**76.04**
MFC	T	36.39	39.42	54.66	29.90	31.82	78.83	36.79	33.78	54.84
LSTM	T	**75.37**	94.75	76.84	**77.94**	94.35	87.33	**83.32**	89.78	72.34
ELMO	T	75.34	**95.39**	**89.51**	77.81	**94.76**	**91.48**	82.97	**90.85**	**76.48**

Table 4.8: Dialogue act (DA), Intent class (IC), and slot labeling (SL) F1 scores by domain for the majority class, LSTM, and ELMobaselines on data annotated at the sentence (S) and turn (T) level. Bold text denotes the model architecture with the best performance for a given annotation granularity, i.e. sentence or turn level. Red highlight denotes the model with the best performance on a given task across annotation granularities.

	Airline		Fast Food		Finance		Insurance		Media		Software	
A	Single	Joint	Single	Joint	Single	Joint	Single	Joint	Single	Joint	Single	Joint
S	97.32	**97.44**	91.03	**91.26**	94.07	**94.27**	94.63	**94.99**	94.27	**94.47**	93.66	**94.00**
T	84.04	**84.64**	**65.69**	65.35	**76.29**	75.68	75.34	**75.89**	77.81	**78.56**	82.97	**83.76**

Table 4.9: Joint training of ELMo on all agent DA data leads to a slight increase in test performance. However, we expect stronger joint models that leverage transfer learning should see a larger improvement. Bold text denotes the training strategy, i.e. single domain (Base) or multi-domain (Joint), with the best performance for a given annotation granularity. Red highlight denotes the strategy with the highest DA F1 score across annotation granularities.

remove duplicate conversations and utterances with inconsistent annotations. The most common source of inconsistent annotations in our dataset is imprecise selection of slot label spans by annotators, which results in sub-token slot labels. While much of this inconsistent data could likely be recovered by mapping each character span to the nearest token span, we drop these utterances to ensure these errors have no effect on our experimental results. Our post-processed data is pruned to approximately 90% of the original size. We form splits for each domain at the conversation level by randomly assigning 70% of conversations to train, 10% to development, and 20% to test. Conversation level splits enable the application of contextual models to our dataset, as each conversation is assigned to a single split. However, our conversation level splits result in imbalanced intent and slot label distributions.

Models: We evaluate the performance of two neural models on each domain. The first is a bi-directional LSTM (Hochreiter and Schmidhuber, 1997) with GloVe

word embeddings, a hidden state of size 512, and two fully connected output layers for slot labels and intent classes respectively. The second model, ELMo, is similar to the LSTM architecture but it additionally uses pre-trained ELMo (Peters et al., 2018) embeddings in addition to GloVe word embeddings, which are kept frozen during training. We combine these ELMo and GloVe embeddings via concatenation. As a sanity check, we also include a most frequent class (MFC) baseline. The MFC baseline assigns the most frequent class label in the training split to every utterance u' in the test split for both DA and IC tasks. To adapt the MFC baseline to SL, we compute the most frequent slot label MFC(w) for each word type w in the training set. Then given a test utterance u', we assign the pre-computed, most frequent slot MFC(w') to each word $w' \in u'$ if w' is present in the training set. If a given word $w' \in u'$ is not present the training set, we assign the *other* slot label, which denotes the absence of a slot, to w'. We use the AllenNLP (**?**) library to implement these models and evaluate our performance. We use the Adam optimizer (**?**) with a learning rate of 0.001 to train the LSTMand ELMomodels for 50 epochs, using batch sizes 256 and 128, respectively. In addition, we employ early stopping on the validation loss with a tolerance of 10 epochs to prevent over fitting.

Evaluation Metrics: We report micro F1 score to evaluate DA and IC performance of our models. Similarly, we use a span based F1 score, implemented in the seqeval[5] library, to evaluate SL performance.

4.5.1 Results

DA/IC/SL Results. Table 4.8 presents the MFC, LSTM, and ELMoresults for each domain, on the subset of 15,000 conversations annotated at both the turn and sentence levels. In general for both granularities Turn and Sentence, both LSTM, and ELMooutperform MFC significantly across all domains. Relative to the LSTM, we find that ELMoobtains a modest increase in IC accuracy of 0.41 to 2.20 F1 points and a significant increase in SL F1 score on all domains. Concretely, ELMoboosts SL F1 performance by 3.16 to 13.17 F1 points. We see the biggest SL gains on the Insurance domain, where sentence level ELMoachieves the 13.17 point F1 gain and turn level ELMoachieves a 12.67 point F1 gain. Performance gains on the Airline domain are also large; here, ELMoincreases sentence and turn level SL F1 score by 12.38 and 9.86 F1 points, respectively. Both LSTMand ELMoyield similar performance in terms of F1 score on DA classification for which the difference in performance of these models is within one F1 point across all domains. In general, the Fast Food domain yields the overall lowest absolute F1 scores. Recall that Fast Food had the most diverse dialogues (biases) as per Table 4.4 and the lowest IAA as per Table 4.7.

Sentence vs. Turn Level Annotation Units. Regarding the performance of the LSTMand ELMomodels on sentence vs. turn level annotation units, our results suggest that turn level annotations increase the difficulty of the DA classification task. This finding is evidenced by DA performance of our models on the Fast Food domain, for which F1 score is up to 25 F1 points lower for turn level annotations

[5]https://github.com/chakki-works/seqeval

than sentence level annotations. We believe the increased difficulty of turn level DA relative to sentence level DA is driven by a corresponding increase in the confusability of turn level dialogue acts. This assertion of greater turn level DA confusability is supported by the lower inter annotator agreement (IAA) scores on turn level DA, which range from 0.314 to 0.521, relative to IAA scores for sentence level DA, which range from 0.598 to 0.709. This experimental result highlights the importance of collecting sentence level annotations for conversational DA datasets. Somewhat surprisingly, our models achieve similar IC F1 and SL F1 scores on turn and sentence level annotations. We hypothesize that the choice of annotation unit has a lesser impact on the IC and SL tasks because customer utterances are more likely to focus on a single speech act, whereas Agent utterances may be more complex in comparison and include a greater number of speech acts.

Joint Training on Agent DA. Agent DA classification naturally lends itself to joint training, given agent DAs are shared among all domains. To explore the benefits of multi-domain training, we jointly train an agent DA classification model on all domains and report test results for each domain separately. These results are provided in Table 4.9. This straightforward technique leads to a consistent but less than one point improvement in F1 scores. We expect that more sophisticated transfer learning methods (**?**Howard and Ruder, 2018) could generate larger improvements for these domains.

Overall, our results demonstrate that there is still headroom for performance improvement, especially for the SL task, across all domains. Consequently, `MultiDoGO` should be a relevant benchmark for developing new state-of-the-art NLU models for the foreseeable future.

4.6 Future Directions

The data collection and annotation methodology that we use to gather `MultiDoGO` can efficiently scale across languages. Several pilot experiments aimed at collecting Spanish dialogues in the same domains have shown preliminary success in quality assessment. The production of a NLU dataset with parallel data in multiple languages would be a boon to the cross-lingual research community. To date, cross-lingual NLU research (Upadhyay et al., 2018; Schuster et al., 2018) has relied on much smaller parallel corpora.

4.7 Conclusion

We present `MultiDoGO`, a new Wizard-of-Oz dialogue dataset that is the largest human-generated, multi-domain corpora of conversations to date. The scale and range of this data provides a test-bed for future work in joint training and transfer learning. Moreover, our comparison of sentence and turn level annotations provides insight into the effect of annotation granularity on downstream model performance.

By pairing crowd-sourced labor (Chapter 3) with experts(Chapter 6, we balance the cost, diversity, and quality of these conversations in a scalable manner.

We show that by adopting a modular annotation strategy, the crowds can reliably *annotate* dialogues at a level commensurate with trained professional annotators. Without any oversight, our data would be just as large, but it could not be trusted.

However, there is a stark difference in quality of the *generated* language between the crowd-sourced workers and the experts, in this case Amazon Customer Service agents. The crowd-sourced workers have a financial incentive to complete the task as quickly as possible and contribute sentences that are often prosaic, ungrammatical, or repeated. This begs the question, can we create datasets using only experts and avoid quality control issues altogether?

Chapter 5: Expert Design

We need an evaluation methodology to show that expert-sourced datasets are quantitatively superior to generalist-sourced ones. Since most datasets are evaluated on the same types of data—SQuAD test data is comparable to the training data—this difference is not readily captured by standard quantitative metrics like accuracy or F_1. We propose a new dataset similar to Checklist (Ribeiro et al., 2020) for testing coreference in machine translation in Section 5.1. Genuinely varied, realistic data should create models that are robust to minor variations.

This dataset is *designed* by experts: specifically native German and native English speakers, even if the methodology is *automated*. While a similar dataset of the same size could be created without knowledge of either language, the templates used as test data would prove be nonsensical or unnatural.

5.1 Meaningful Model Evaluation in Machine Translation

Due to the intrinsic evaluation of many datasets, higher standards of evaluation would better understand the strength of machine learning models, and indirectly the data used to train them. A model that has memorized several key answers upon which it is then tested is not necessarily *learning*. A raw analysis of data overlap appears this is at least partially a problem (Lewis et al., 2020). Datasets meant to effectively and robustly evaluate trained datasets can determine how much of a problem this poses *ex-post-facto*.

Machine translation is a complex task that requires diverse linguistic knowledge and data in multiple languages, making it a good task for evaluating data quality. We focus on German-English coreference resolution as a representative task. The seemingly straightforward translation of the English pronoun *it* into German requires knowledge at the syntactic, discourse and world knowledge levels for proper pronoun coreference resolution (CR). A German pronoun can have three genders, determined by its antecedent: masculine (er), feminine (sie) and neuter (es).

Accuracy in machine translation is at an all-time high with the rise of neural architectures; but does accuracy alone suffice? Previous work (Hardmeier and Federico, 2010; Miculicich Werlen and Popescu-Belis, 2017; Müller et al., 2018) proposed evaluation methods for specifically pronoun translation. This has been of special interest in context-aware NMT models that are capable of using discourse-level information. Despite promising results, the question remains: Are transformers (Vaswani et al., 2017) truly *learning* this task, or are they exploiting simple heuris-

tics to make a coreference prediction? If so, they are learning heuristics, then these must stem from limitations in the underlying data, which in turn suggests a need for more natural and higher-quality training data. To empirically answer this question, we propose extending ContraPro (Müller et al., 2018)—a contrastive challenge set for automatic English→German pronoun translation evaluation—by making small adversarial changes in the contextual sentences.[1]

Our adversarial attacks on ContraPro will show if context-aware Transformer NMT models can easily be misled by simple and unimportant changes to the input. However, interpreting the results obtained from adversarial attacks can be difficult. Positive results will show that NMT uses brittle heuristics to solve CR, without identifying the exact heuristic. Modifying ContraPro alone will not test specific phenomena.

For this reason, we propose an *independent* set of templates for coreferential pronoun translation evaluation to systematically investigate which heuristics are being used. Inspired by previous work on CR (Raghunathan et al., 2010; Lee et al., 2011), we will create templates tailored to evaluating the specific steps of an idealized CR pipeline. We will call this collection ContraCAT, **C**ontrastive **C**oreference **A**nalytical **T**emplates. The templates will be constructed in a completely controlled manner, enabling us to easily create large number of coherent test examples and provide unambiguous conclusions about the CR capabilities of NMT. While this methodology depends on automation, a technique called into question in Chapter 3, the templates will be crucially written in collaboration with a native German speaker. The procedure used in creating these templates can be adapted to many language pairs with little effort.

We also propose a simple data augmentation approach using fine-tuning. This methodology should not change the way CR is being handled by NMT and support the hypothesis that automated data techniques have limited applicability. We will publicly release a new dataset, ContraCAT, and the adversarial modifications to ContraPro.

We motivate coreference resolution as a task in Section 5.2, discuss ContraPro in Section 5.5.1, explain our proposed templates in Section refsec:templates,

5.2 Why is Coreference Resolution Relevant?

Addressing discourse phenomena is important for high-quality MT. Apart from document-level coherence and cohesion, anaphoric pronoun translation has proven to be an important testing ground for the ability of context-aware NMT to model discourse. Anaphoric pronoun translation is the focus of several works in context-aware NMT (Bawden et al., 2018; Voita et al., 2018; Stojanovski and Fraser, 2018; Miculicich et al., 2018; Voita et al., 2019; Maruf et al., 2019).

[1]Equal effort between Denis Peskov, Benno Krojer, Dario Stojanovski, and supervised by Alex Fraser. 2020. In International Conference on Computational Linguistics
Peskov is responsible for part of template design, selecting concrete nouns for the templates, paper writing, and the video.

Start: Original sentence	The cat and the actor were hungry. It (?) was hungrier.
Step 1: Markable Detection	The **cat** and the **actor** were hungry. It (?) was hungrier.
Step 2: Coreference Resolution	The cat and the actor were hungry. **It** was hungrier.
Step 3: Language Translation	Der Schauspieler und die Katze waren hungrig. Er / **Sie** / Es war hungriger.

Table 5.1: A hypothetical CR pipeline that sequentially resolves and translates a pronoun.

However, the choice of an evaluation metric for CR is nontrivial. BLEU-based evaluation is insufficient for measuring improvement in CR (Hardmeier, 2012) without carefully selecting or modifying test sentences for pronoun translation (Voita et al., 2018; Stojanovski and Fraser, 2018). Alternatives to BLEU include F_1, partial credit, and oracle-guided approaches (Hardmeier and Federico, 2010; Guillou and Hardmeier, 2016; Miculicich Werlen and Popescu-Belis, 2017). However, Guillou and Hardmeier (2018) show that these metrics can miss important cases and propose semi-automatic evaluation. In contrast, our evaluation will be *completely* automatic. We focus on scoring-based evaluation (Sennrich, 2017), which works by creating contrasting pairs and comparing model scores. Accuracy is calculated as how often the model chooses the correct translation from a pool of alternative incorrect translations. This is an evaluation metric applicable for multiple forms of *generated* NLP data.

Bawden et al. (2018) manually create such a contrastive challenge set for English→French pronoun translation. ContraPro (Müller et al., 2018) follows this work, but creates the challenge set in an automatic way.

We show that making small variations in ContraPro substantially changes the scores. Our work is related to adversarial datasets for testing robustness used in Natural Language Processing tasks such as studying gender bias (Zhao et al., 2018; Rudinger et al., 2018; Stanovsky et al., 2019), natural language inference (Glockner et al., 2018) and classification (Wang et al., 2019b).

Jwalapuram et al. (2019) propose a model for pronoun translation evaluation trained on pairs of sentences consisting of the reference and a system output with differing pronouns. However, as Guillou and Hardmeier (2018) point out, this fails to take into account that often there is not a 1:1 correspondence between pronouns in different languages and that a system translation may be correct despite not containing the exact pronoun in the reference, and incorrect even if containing the pronoun in the reference, because of differences in the translation of the referent. Moreover, introducing a separate model which needs to be trained before evaluation adds an extra layer of complexity in the evaluation setup and makes interpretability

more difficult. In contrast, templates can easily be used to pinpoint specific issues of an NMT model. Our templates follow previous work (Ribeiro et al., 2018; McCoy et al., 2019; Ribeiro et al., 2020) where similar tests are proposed for diagnosing NLP models.

5.3 Do Androids Dream of Coreference Translation Pipelines?

Imagine a hypothetical coreference pipeline that generates a pronoun in a target language, as illustrated in Table 5.1. **First**, markables (entities that can be referred to by pronouns) are tagged in the source sentence (we restrict ourselves to concrete entities as we wish to detect gender). Then, the subset of animate entities are detected, and human entities are separated from other animate ones (since *it* cannot refer to a human entity). **Second**, coreferences are resolved in the source language. This entails addressing phenomena such as world knowledge, pleonastic *it*, and event references. **Third**, the pronoun is translated into the target language. This requires selecting the correct gender given the referent (if there is one), and selecting the correct grammatical case for the target context (e.g., accusative, if the pronoun is the grammatical object in the target language sentence).

This idealized pipeline would produce the correct pronoun in the target language. The coreference steps resemble the rule-based approach implemented in Stanford CoreNLP's CorefAnnotator (Raghunathan et al., 2010; Lee et al., 2011). However, NMT models are unable to decouple the individual steps of this pipeline. We propose to isolate each of these steps through targeted examples.

5.4 Model

We use a transformer model for all experiments and train a sentence-level model as a baseline. The context-aware model in our experimental setup is a concatenation model (Tiedemann and Scherrer, 2017) (CONCAT) which is trained on a concatenation of consecutive sentences. CONCAT is a standard transformer model and it differs from the sentence-level model only in the way that the training data is supplied to it. The training examples for this model are modified by prepending the previous source and target sentence to the main source and target sentence. The previous sentence is separated from the main sentence with a special token <SEP>, on both the source and target side. This also applies to how we prepare the ContraPro and ContraCAT data. We train the concatenation model on OpenSubtitles2018 data prepared in this way. We remove documents overlapping with ContraPro.

5.5 Adversarial Attacks

ContraPro, a contrastive challenge set, has limitations and our methodology for creating our own dataset addresses them.

5.5.1 About ContraPro

ContraPro is a contrastive challenge set for English→German pronoun translation evaluation. The set consists of English sentences containing an anaphoric pronoun "it" and the corresponding German translations. It contains three contrastive translations, differing based on the gender of the translation of *it*: *er*, *sie*, or *es*. The challenge set artificially balances the amount of sentences where *it* is translated to each of these three German pronouns. The appropriate antecedent may be in the main sentence or in a previous sentence. For evaluation, a model needs to produce scores for all three possible translations, which are compared against ContraPro's gold labels.

We create automatic adversarial attacks on ContraPro that modify theoretically inconsequential parts of the context sentence before the occurrence of it. Contrary to expectations, accuracy degrades in all adversarial attacks.

5.5.2 Adversarial Attack Generation

Our three modifications are:

1. **Phrase Addition**: Appending and prepending phrases containing implausible antecedents: The Church is merciful but that's not the point. It always welcomes the misguided lamb.
2. **Possessive Extension**: Extending original antecedent with possessive noun phrase: I hear ~~her~~ the doctor's voice! It resounds to me from heights and chasms a thousand times!
3. **Synonym Replacement**: Replacing original German antecedent with synonym of different gender (note: der Vorhang (masc.) and die Gardine (fem.) are synonyms meaning curtain):
   The curtain rises. It rises. → ~~Der Vorhang~~ Die Gardine geht hoch. ~~Er~~ Sie geht hoch.

Phrase Addition can be applied to all 12,000 ContraPro examples. The second and third attack can only be applied to 3,838 and 1,531 examples, due to the required sentence contingencies.

5.5.2.1 Phrase Addition

This attack modifies the previous sentence by appending phrases such as "*. . . but he wasn't sure*" and also prepending phrases such as "it is true:. . . ". A range of other simple phrases can be used, which we leave out for simplicity. All phrases we tried provided lower scores. These attacks either introduce a human entity or an event reference *it* (e.g., *"it is true"*) which are both not plausible antecedents for the anaphoric *it*.

5.5.2.2 Possessive Extension

This attack introduces a new human entity by extending the original antecedent A with a possessive noun phrase e.g., "*the woman's A*". Only two-thirds of the 12,000 ContraPro sentences are linked to an antecedent phrase. Grammar and misannotated antecedents exclude half of the remaining phrases. We put POS-tag constraints on the antecedent phrases before extending them. This filters our subset to 3,838 modified examples. Our possessive extensions can be humans (*the woman's*), organisations (*the company's*) and names (*Maria's*).

5.5.2.3 Synonym Replacement

This attack modifies the original German antecedent by replacing it with a German synonym of a different gender. For this we first identify the English antecedent and its most frequent synset in WordNet (Miller, 1995b). We obtain a German synonym by mapping this WordNet synsets to GermaNet (Hamp and Feldweg, 1997) synsets. Finally, we modify the correct German pronoun translation to correspond to the gender of the antecedent synonym.

Approximately one quarter of the nouns in our ContraPro examples are found in GermaNet. In 1,531 cases, a synonym of different gender could be identified.

Understanding the pronoun/noun relationship is needed to score well on the Synonym Replacement attack. This attack gets to the core of whether NMT uses CR heuristics instead.

We evaluate a random sample of 100 auto-modified examples as a quality control metric. There are 11 issues with semantically-inappropriate synonyms. Overall, in 14 out of 100 cases, the model switches from correct to incorrect predictions because of synonym-replacement. Only 4 out of these 14 cases come from the questionable synonyms, showing that the drop in ContraPro scores is meaningful.

5.5.2.4 Evaluating Adversarial Attacks

Intuitively, the adversarial attacks should not contribute to large drops in scores, since no meaningful changes are being made. If the model accuracy drops some, but not all the way to the baseline, we can conclude that the concatenation model handles CR, but likely with brittle heuristics. If the model accuracy drops all the way to the original sentence-level baseline, then the model is memorizing the inputs. These results can expose potential issues with the model, but it will still be difficult to pinpoint the specific problems. This reveals a larger issue with pronoun translation evaluation that cannot be addressed with simple adversarial attacks on existing general-purpose challenge sets. We propose ContraCAT, a more systematic approach that targets each of the previously outlined CR pipeline steps with data synthetically generated from corresponding templates.

Automatic adversarial attacks offer less freedom than templates as many systematic modifications cannot be applied to the average sentence. Thus, our ContraCAT templates will be built on the hypothetical coreference pipeline in Section 5.3

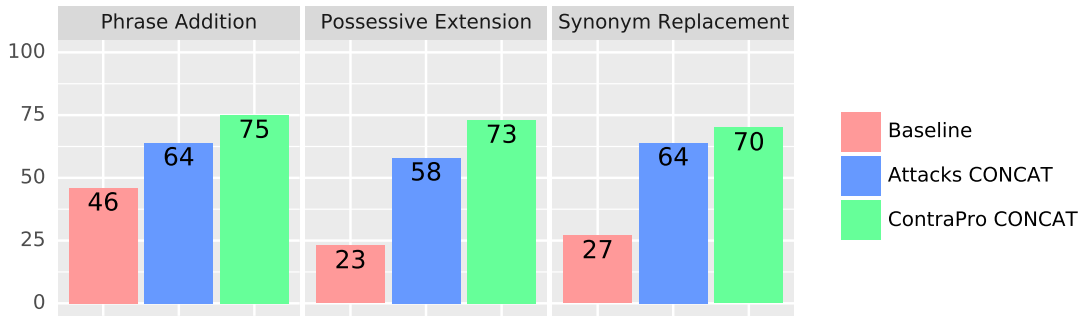

Figure 5.1: Results with the sentence-level Baseline and CONCAT on ContraPro and three adversarial attacks. The adversarial attacks modify the context, therefore the Baseline model's results on the attacks are unchanged and we omit them. **Phrase**: prepending "it is true: ...". **Possessive**: replacing original antecedent A with "Maria's A". **Synonym**: replacing the original antecedent with different-gender synonyms. Results for Phrase Addition are computed based on all 12,000 ContraPro examples, while for Possessive Extension and Synonym Replacement we only use the suitable subsets of 3,838 and 1,531 ContraPro examples.

that target each of the three steps: i) Markable Detection, ii) Coreference Resolution and iii) Language Translation. Our minimalistic templates draw entities from sets of animals, human professions (McCoy et al., 2019), foods, and drinks, along with associated verbs and attributes. We use these sets to fill slots in our templates. Animals and foods are natural choices for subject and object slots referenced by it. Restricting our sets to interrelated concepts with generically applicable verbs—all animals eat and drink—ensures semantic plausibility. Other object sets, such as buildings, would cause semantic implausibility with certain verbs.

5.5.2.5 Template Generation

Our templates consist of a previous sentence that introduces at least one entity and a main sentence containing the pronoun it. We use contrastive evaluation to judge anaphoric pronoun translation accuracy for each template; we create three translated versions for each German gender corresponding to an English sentence, e.g., *"The cat ate the egg. It rained."* and the corresponding *"Die Katze hat das Ei gegessen. Er/Sie/Es regnete"*. To fill a template, we only draw pairs of entities with two different genders, i.e., for animal a and food f: gender(a) $\neq$ gender(f). This way we can determine whether the model has picked the right antecedent.

First, we will create templates that analyze priors of the model for choosing a pronoun when no correct translation is obvious. Then, we will create templates with correct translations, guided by the three broad coreference steps. Table 5.3 provides examples for our templates.

5.5.2.6 Priors

Prior templates do not have a correct answer, but help to understand the model's biases. We will expose three priors with our templates: i) grammatical roles prior (e.g., subject) ii) position prior (e.g., first antecedent) and iii) a general prior if no antecedent and only a verb is present.

For i), we will create a Grammatical Role template where both subject and object are valid antecedents.

For ii), we will create a Position template where two objects are enumerated as shown in Table 5.3. We will create an additional example where the entities order is reversed and test if there are priors for specific nouns or alternatively positions in the sentence.

For iii), we will create a Verb template, expecting that certain transitive verbs trigger certain object gender choice. We will use 100 frequent transitive verbs and create sentences such as the example in Table 5.3.

5.5.2.7 Markable Detection with a Humanness Filter

Before doing the actual CR, the model will need to identify all possible entities that *it* can refer to. We will construct a template that contains a human and animal which are in principle plausible antecedents, if not for the condition that it does not refer to people. For instance, the model should always choose cat in *" The actress and the cat are hungry. However it is hungrier."*.

5.5.2.8 Coreference Resolution

Having determined all possible antecedents, the model will have to choose the correct one, relying on semantics, syntax, and discourse. The pronoun *it* can in principle be used as an anaphoric (referring to entities), event reference or pleonastic pronoun (Loáiciga et al., 2017). For the anaphoric *it*, we identify two major ways of identifying the antecedent: lexical overlap and world knowledge. Our templates for these categories are meant to be simple and solvable.

Overlap: Broadly speaking the subject, verb, or object can overlap from the previous sentence to the main sentence, as well as combinations of them. This gives us five templates: i) subject-overlap ii) verb-overlap iii) object-overlap iv) subject-verb-overlap and v) object-verb-overlap. We always use the same template for the context sentence, e.g., *"The **cat** ate the apple and the **owl** drank the water."*. For the object-verb-overlap we would then create the main sentence *"It ate the apple quickly."* and expect the model to choose cat as antecedent. To keep our overlap templates order-agnostic, we vary the order in the previous sentence by also creating *"The **owl** drank the water and the **cat** ate the apple."*

World Knowledge: CR has been traditionally seen as challenging as it requires world knowledge. Our templates will test simple forms of world knowledge by using attributes that either apply to animal or food entities, such as *cooked* for food or *hungry* for animals. We then evaluate whether the model chooses e.g., cat in *"The*

cat *ate the cookie. It was hungry."* The model occasionally predicts answers that require world knowledge, but most predictions are guided by a prior for choosing the neuter es or a prior for the subject.

Pleonastic and Event Templates: For the other two ways of using it, event reference and pleonastic-it, we again create a default previous sentence (*"The* ***cat*** *ate the apple.")*. For the main sentence, we used four typical pleonastic and event reference phrases such as *"It is a shame"* and *"It came as a surprise"*. We expect the model to correctly choose the neuter es as a translation every time.

5.5.2.9 Translation to German

After CR, the decoder has to translate from English to German. In our contrastive scoring approach the translation of the English antecedent to German is already given. However the decoder is still required to know the gender of the German noun to select between er, sie or, es. We will test this with a list of concrete nouns selected from Brysbaert et al. (2014), which we filter for nouns that occur more than 30 times in the training data. This selects 2051 nouns that are plugged into: *"I saw a N. It was {big, small}."*.

5.5.3 Results

The model performs poorly when actual CR is required. It frequently falls back to choosing the neuter es or preferring a position (e.g., first of two entities) for determining the gender. For Markable Detection the model always predicts the neuter es regardless of the actual genders of the entities.

In the Overlap template, the model fails to recognize the overlap and has a general preference for one of the two clauses. For instance in the case of verb-overlap, the model had a solid accuracy of 64.1% if the verb overlapped from the first clause (*"The cat ate and the dog drank. It ate a lot."*) but a weak accuracy of 39.0% when the verb overlapped from the second clause (*"The cat ate and the dog drank. It drank a lot."*.) The overall accuracy for the overlap templates is 47.2%, with little variation across the types of overlap. Adding more overlap, e.g., by overlapping both the verb and object (*"It ate the apple happily"*), yields no improvement. Overall, the model pays very little attention to overlaps when resolving pronouns.

We also see weak performance for world knowledge. An accuracy of 55.7% is slightly above the heuristic of randomly choosing an entity (= 50.0%). With a strong bias for the neuter es, the model has a high accuracy of 96.2% for event reference and pleonastic templates, where es is always the correct answer. Based on the strong performance on the Gender template in **??**, we conclude the model consistently memorized the gender of concrete nouns. Hence, CR mistakes stem from Step 1 or Step 2, suggesting that the model failed to learn proper CR.

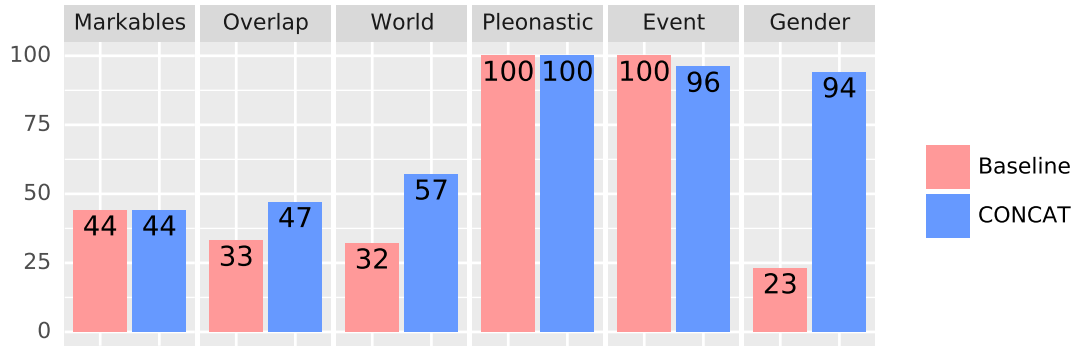

Figure 5.2: Results comparing the sentence-level baseline to CONCAT on ContraCAT. Pronoun translation pertaining to World Knowledge and language-specific Gender Knowledge benefits the most from additional context.

Antecedent-free augmentation	
Source	You let me worry about that. <SEP> How much you take for it?
Reference	Lassen Sie das meine Sorge sein. <SEP> Wie viel kostet er?
Augmentation 1	Lassen Sie das meine Sorge sein. <SEP> Wie viel kostet sie?
Augmentation 2	Lassen Sie das meine Sorge sein. <SEP> Wie viel kostet es?

Table 5.2: Examples of training data augmentations. The source side of the augmented examples remains the same.

5.6 Augmentation

We present an approach for augmenting the training data. While challenging for NLP, we focus on a narrow problem which lends itself to easier data manipulation. Our previous analyses show that our model is capable of modeling the gender of nouns. However, they also show a strong prior for translating *it* to *es* and very little CR capability. Our goal with the augmentation is to break off the strong prior and test if this can improve CR in the model.

We augment our training data and call it Antecedent-free augmentation (AFA). We identify candidates for augmentation as sentences where a coreferential *it* refers to an antecedent not present in the current or previous sentence (e.g., *I told you before. <SEP> It is red.* → *Ich habe dir schonmal gesagt. <SEP> Es ist rot.*). We create augmentations by adding two new training examples where the gender of the German translation of "it" is modified (e.g., the two new targets are "*Ich habe dir schonmal gesagt. <SEP> Er ist rot.*" and "*Ich habe dir schonmal gesagt. <SEP> Sie ist rot.*"). The source side remains the same. Table 5.2 provides an additional example. Antecedents and coreferential pronouns are identified using a CR tool (Clark and Manning, 2016a,b). We fine-tune our already trained concatenation model on a dataset consisting of the candidates and the augmented samples. As a baseline, we fine-tune on the candidates to confidently say that any potential improvements come from the augmentations.

5.6.1 Results

Results for adversarial attacks on ContraPro and on our templates are independent and are discussed separately.

5.6.1.1 Adversarial Attacks

AFA provides large improvements, scoring 85.3% on ContraPro. Results are in Figure **??**. The AFA baseline (fine-tuning on the augmentation candidates only) improves by 1.94%, presumably because many candidates consist of coreference chains of "it" and the model learns they are important for coreferential pronouns. However, the improvement is small compared to AFA.

Results on ContraPro for each gender (see Appendix) show that performance on *er* and *sie* is substantially increased, suggesting that the augmentation successfully removes the strong bias towards *es*. Templates provide further evidence about this. Although, the adversarial attacks lower AFA scores, in contrast to CONCAT, the model is more robust and the performance degradation is substantially lower (except on the synonym attack). We experimented with different learning rates during fine-tuning and present results with the LR that obtained the best baseline ContraPro score. Detailed scores in the Appendix show how LR can balance the scores across the three different genders. Furthermore, CONCAT and AFA obtain 31.5 and 32.2 BLEU on ContraPro, showing that this fine-tuning procedure, which is tailored to pronoun translation, does not lead to any degradation in general translation quality.

5.6.1.2 Templates

From the prior templates, the prior over gender pronouns is more evenly spread and not concentrated on *es*. This also provides for a more even distribution on the Position and Role Prior template.

The augmented model is also substantially better on markable detection, improving by 27.6%. Results for templates are in Figure 5.3.

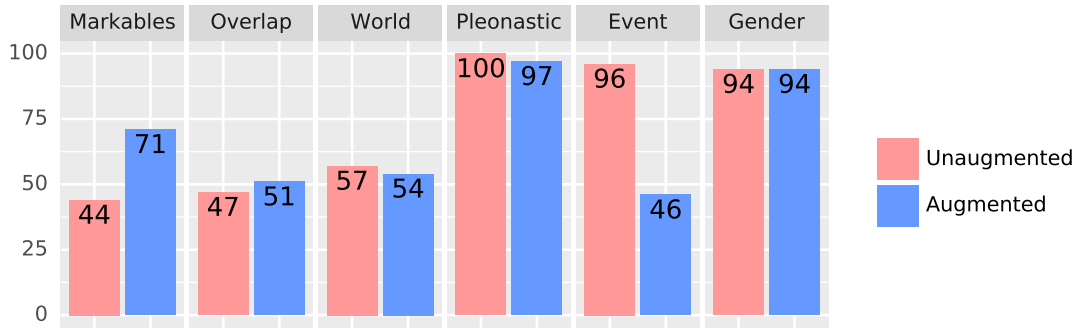

Figure 5.3: ContraCAT results with unaugmented and augmented CONCAT. We speculate that readjusting the prior over genders in augmented CONCAT explains the improvements on Markable and Overlap.

Template Target	Example
Priors	
Grammatical Role	The **cat** ate the **egg**. It (*cat*/*egg*) was big.
Order	I stood in front of the **cat** and the **dog**. It (*cat*/*dog*) was big.
Verb	Wow! She unlocked it.
Markable Detection	
Filter Humans	The ***cat*** and the *actress* were happy. However it (*cat*) was happier.
Coreference Resolution	
Lexical Overlap	The ***cat*** ate the apple and the *owl* drank the water. It (*cat*/*dogFir*) ate the apple quickly.
World Knowledge	The ***cat*** ate the *cookie*. It (*cat*) was hungry.
Pleonastic it	The cat ate the sausage. It was raining.
Event Reference	The cat ate the carrot. It came as a surprise.
Language Translation	
Antecedent Gender	I saw a ***cat***. It(*cat*) was big. → Ich habe eine Katze gesehen. Sie (*cat*) war groß.

Table 5.3: Template examples targeting different CR steps and substeps. } for Animals. For German, we create three versions with er, sie, or es as different translations of it.

No improvements are observed on the World Knowledge template. Pleonastic cases are still accurate, although not perfect as with CONCAT. The Event template identifies a systematic issue with our augmentation. We presume this is due to the CR tool marking cases where *it* refers to events. We do not apply any filtering and augment these cases as well, thus create wrong examples (an event reference *it* cannot be translated to *er* or *sie*). As a result, the scores are significantly lower compared to CONCAT. This issue with our model is not visible on ContraPro and the adversarial attacks results. In contrast, the Event template easily identifies this problem.

AFA performs on par with the unaugmented baseline on the Gender template. However, despite increasing by 3.8%, results on Overlap are still underwhelming.

AFA performs on par with the unaugmented baseline on the Gender template. However, despite increasing by 3.8%, results on Overlap are still underwhelming. Our analysis shows that augmentation helps in changing the prior. We believe this provides for improved CR heuristics which in turn provide for an improvement in coreferential pronoun translation. Nevertheless, the Overlap template shows that augmented models still do not solve CR in a fundamental way.

5.7 Recap

In this work, we will study how and to what extent CR is handled in context-aware NMT. This work aims to show that that standard challenge sets can easily be manipulated with adversarial attacks that cause dramatic drops in performance, suggesting that NMT uses a set of heuristics to solve the complex task of CR. Attempting to diagnose the underlying reasons, we propose targeted templates which systematically test the different aspects necessary for CR. This analysis will show that while some type of CR such as pleonastic and event CR are handled well, NMT does not solve the task in an abstract sense. We also propose a data augmentation approach to see if simple data modifications can improve model accuracy. This methodology will illustrate the dependence on data by models, and strengthen our claims that low-cost data generation techniques are approximating rather than *solving* NLP tasks. Having identified limitations in existing models, we will then be able to argue for concrete data extensions for coreference resolution. This methodology—creating an adverserial dataset which tests the understanding of a model—can be applied to most NLP tasks.

Chapter 6: Expert Participation

Expert-driven datasets require a large investment of time, relationship-building, and money. As a contrast to the earlier work, we create a deception dataset using only experts. Participants both *generate* and *annotate* data in the span of a game that usually lasts over a month. And they are handsomely compensated for their effort. The resulting product is a gold standard of conversational NLP data in terms of quality of language, diversity, and naturalness.[1] The conversations and annotations thereof would not be possible without experts from the community.

6.1 Where Does One Find Long-Term Deception?

A functioning society is impossible without trust. In online text interactions, users are typically trusting (Shneiderman, 2000), but this trust can be betrayed through false identities on dating sites (Toma and Hancock, 2012), spearphishing attacks (Dhamija et al., 2006), sockpuppetry (Kumar et al., 2017) and, more broadly, disinformation campaigns (Kumar and Shah, 2018). Beyond such one-off antisocial acts directed at strangers, deception can also occur in sustained relationships, where it can be strategically combined with truthfulness to advance a long-term objective (Cornwell and Lundgren, 2001; Kaplar and Gordon, 2004).

We introduce a dataset to study the strategic use of deception in long-lasting relationships. To collect reliable ground truth in this complex scenario, we design an interface for players to naturally generate and annotate conversational data while playing a negotiation-based game called Diplomacy. These annotations are done in *real-time* as the players send and receive messages. While this game setup might not directly translate to real-world situations, it enables computational frameworks for studying deception in a complex social context while avoiding privacy issues.

After providing background on the game of Diplomacy and our intended deception annotations (Section 6.2), we discuss our study (Section 6.3). To probe the value of the resulting dataset, we develop lie prediction models (Section 6.4) and analyze their results (Section 6.5).

[1] Denis Peskov, Benny Chang, Ahmed Elgohary, Joe Barrow, Cristian Danescu-Niculescu-Mizil, and Jordan Boyd-Graber. 2020. It Takes Two to Lie: One to Lie and One to Listen. In Proceedings of The Association for Computational Linguistics.
Peskov was responsible for designing the task, gathering the participants, running the games, building half the models, part of the data analysis, the visualizations, and the paper writing.

Message	Sender's intention	Receiver's percep.
If I were lying to you, I'd smile and say "that sounds great." I'm honest with you because I sincerely thought of us as partners.	Lie	Truth
You agreed to warn me of unexpected moves, then didn't . . . You've revealed things to England without my permission, and then made up a story about it after the fact!	Truth	Truth
. . . I have a reputation in this hobby for being sincere. Not being duplicitous. It has always served me well. . . . If you don't want to work with me, then I can understand that . . .	Lie	Truth
(Germany attacks Italy)		
Well this game just got less fun	Truth	Truth
For you, maybe	Truth	Truth

Table 6.1: An annotated conversation between Italy (white) and Germany (gray) at a moment when their relationship breaks down. Each message is annotated by the sender (and receiver) with its intended or perceived truthfulness; Italy is lying about . . . lying. A full transcript of this dialog is available in Appendix, Table **??**.

6.2 Diplomacy

The Diplomacy board game places a player in the role of one of seven European powers on the eve of World War I. The goal is to conquer a simplified map of Europe by ordering armies in the field against rivals. Victory points determine the success of a player and allow them to build additional armies; the player who can gain and maintain the highest number of points wins.[2] The mechanics of the game are simple and deterministic: armies, represented as figures on a given territory, can only move to adjacent spots and the side with the most armies always wins in a disputed move. The game movements become publicly available to all players after the end of a turn.

Because the game is deterministic and everyone begins with an equal amount of armies, a player cannot win the game without forming alliances with other players—hence the name of the game: Diplomacy. Conquering neighboring territories depends on support from another player's armies. After an alliance has outlived its usefulness, a player often dramatically breaks it to take advantage of their erstwhile ally's vulnerability. Table 6.1 shows the end of one such relationship. As in real life, to succeed a betrayal must be a surprise to the victim. Thus, players pride themselves on being able to lie and detect lies. Our study uses their skill and passion to build a dataset of deception created by battle-hardened diplomats. Senders

[2]In the parlance of Diplomacy games, points are "supply centers" in specific territories (e.g., London). Having more supply centers allows a player to build more armies and win the game by capturing more than half of the 34 supply centers on the board.

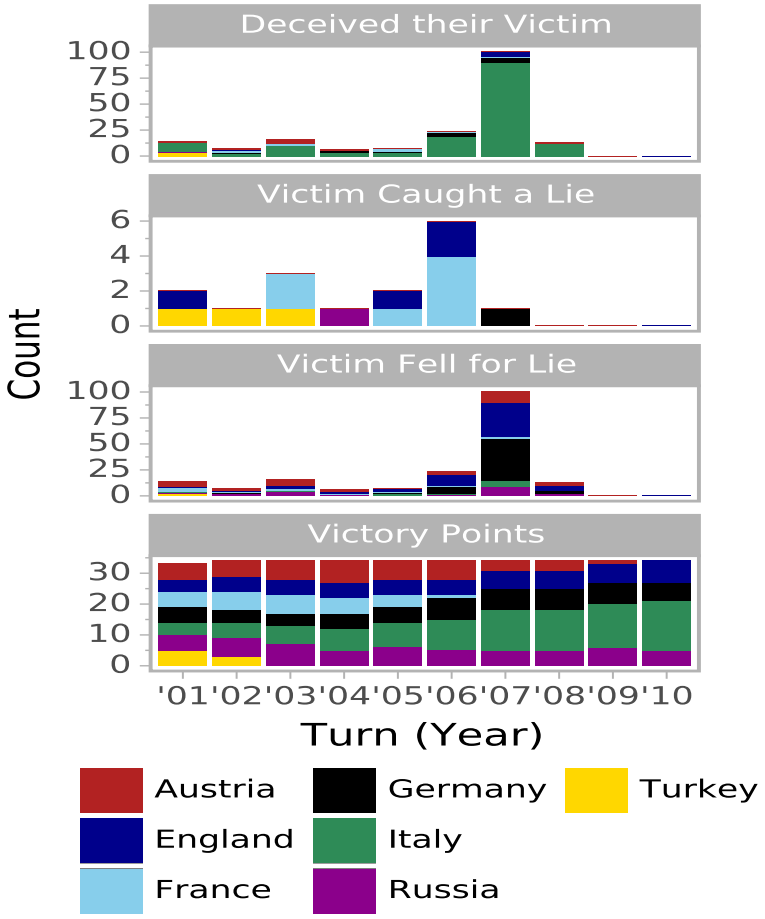

Figure 6.1: Counts from one game featuring an Italy (green) adept at lying but who does not fall for others' lies. The player's successful lies allow them to gain an advantage in points over the duration of the game. In 1906, Italy lies to England before breaking their relationship. In 1907, Italy lies to everybody else about wanting to agree to a draw, leading to the large spike in successful lies.

annotate whether each message they write is an ACTUAL LIE and recipients annotate whether each message received is a SUSPECTED LIE. Further details on the annotation process are in Section 6.3.1.

6.2.1 A game walk-through

Figure 6.1 shows the raw counts of one game in our dataset. But numbers do not tell the whole story. We analyze this case study using rhetorical tactics (Cialdini and Goldstein, 2004), which Oliveira et al. (2017) use to dissect spear phishing e-mails and Anand et al. (2011) apply to persuasive blogs. Mentions of tactics are in italic (e.g., *authority*. For the rest of the paper, we will refer to players via the name of their assigned country.

Through two lie-intense strategies—convincing England to betray Germany and convincing all remaining countries to agree to a draw—Italy gains control of the board. Italy's first deception is a plan with Austria to dismantle Turkey. Turkey believes Italy's initial assurance of non-aggression in 1901. Italy begins by excusing his initial silence due to a rough day at work, evoking empathy and *likability*. While they do not fall for subsequent lies, Turkey's initial gullibility cements Italy's first-strike advantage. Meanwhile, Italy proposes a long-term alliance with England against France, packaging several small truths with a big lie. The strategy succeeds, eliminating Italy's greatest threat.

Local threats eliminated, Italy turns to rivals on the other end of the map. Italy persuades England to double-cross its long-time ally Germany in a moment of *scarcity*: if you do not act now, there will be nowhere to expand. England accepts help from ascendant Italy, expecting *reciprocity*. However, Italy aggressively and successfully moves against England. The last year features a meta-game deception. After Italy becomes too powerful to contain, the remaining four players team up. Ingeniously, Italy feigns acquiescence to a five-way draw, individually lying to each player and establishing *authority* while brokering the deal. Despite Italy's record of deception, the other players believe the proposal (annotating received messages from Italy as truthful) and expect a 1907 endgame, the year with the most lies. Italy goes on the offensive and knocks out Austria.

Each game has relationships that are forged and then riven. In another game, an honest attempt by a strong Austria to woo an ascendant Germany backfires, knocking Austria from the game. Germany builds trust with Austria through a believed fictional experience as a Boy Scout in Maine (*likability*). In a third game, two consecutive unfulfilled promises by an ambitious Russia leads to a quick demise, as their subsequent excuses and apologies are perceived as lies (failed *consistency*). In another game, England, France, and Russia simultaneously attack Germany after offering duplicitous assurances. Game outcomes vary despite the identical, balanced starting board, as different players use unique strategies to persuade, and occasionally deceive, their opponents.

6.2.2 Defining a lie

Statements can be incorrect for a host of reasons: ignorance, misunderstanding, omission, exaggeration. (Gokhman et al., 2012) highlight the difficulty of finding willful, honest, and skilled deception outside of short-term, artificial contexts (DePaulo et al., 2003). Crowdsourced and automatic datasets rely on simple negations (Pérez-Rosas et al., 2017) or completely implausible claims (e.g., "Tipper Gore was created in 1048" from (Thorne et al., 2018b)). While lawyers in depositions and users of dating sites will not willingly admit to their lies, the players of online games are more willing to revel in their deception.

We must first define what we mean by deception. Lying is a mischaracterization; it's thus no surprise that a definition may be divisive or the subject of academic debate (Gettier, 1963). We provide this definition to our users: "Typically, when [someone] lies [they] say what [they] know to be false in an attempt to deceive the

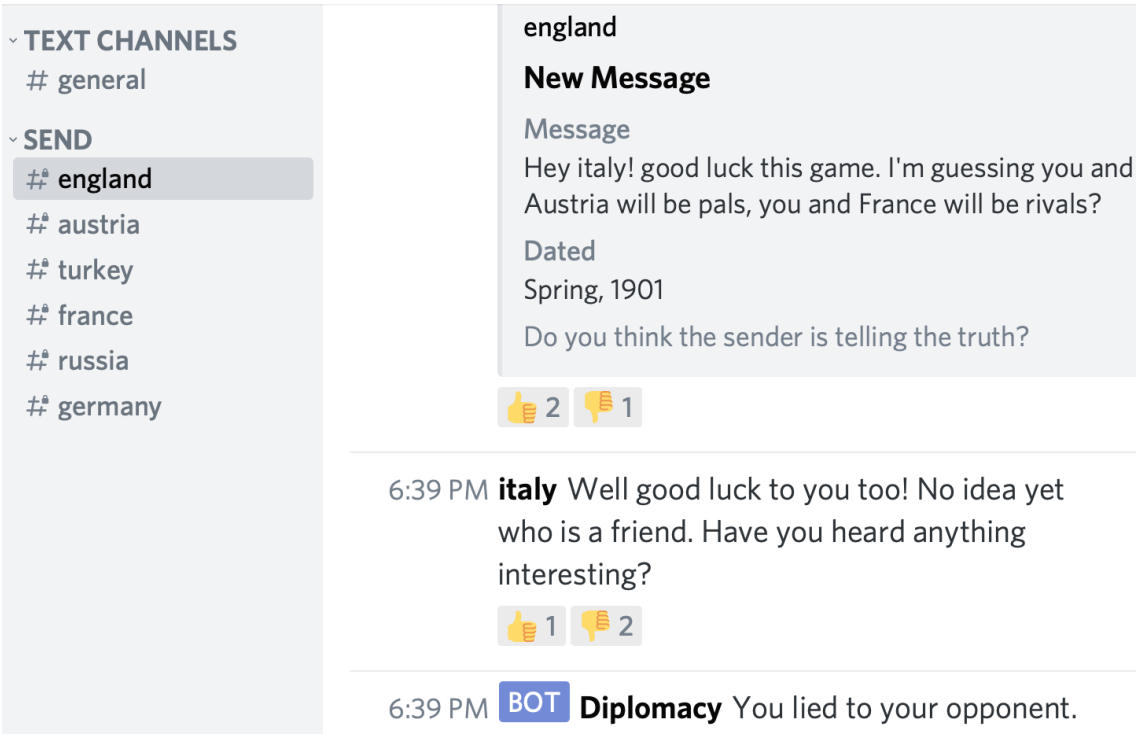

Figure 6.2: Every time they send a message, players say whether the message is truthful or intended to deceive. The receiver then labels whether incoming messages are a lie or not. Here Italy indicates they believe a message from England is truthful but that their reply is not.

listener" (Siegler, 1966). An orthodox definition requires the speaker to utter an explicit falsehood (Mahon, 2016); skilled liars can deceive with a patina of veracity. A similar definition is required for prosecution of perjury, leading to a paucity of convictions (Bogner et al., 1974). Indeed, when we ask participants what a lie looks like, they mention evasiveness, shorter messages, over-qualification, and creating false hypothetical scenarios (DePaulo et al., 2003).

6.2.3 Annotating truthfulness

Previous work on the language of Diplomacy (Niculae et al., 2015) lacks access to players' internal state and was limited to *post-hoc* analysis. We improve on this by designing our own interface that gathers players' intentions and perceptions in real-time (Section 6.3.1). As with other highly subjective phenomena like sarcasm (González-Ibáñez et al., 2011; Bamman and Smith, 2015), sentiment (Pang et al., 2008) and framing (Greene and Resnik, 2009), the intention to deceive is reflective on someone's internal state. Having individuals provide their own labels for their internal state is essential as third party annotators could not accurately access it (Chang et al., 2020).

Most importantly, our gracious players have allowed this language data to be released in accordance with IRB authorized anonymization, encouraging further

work on the strategic use of deception in long-lasting relations.[3]

6.3 Engaging a Community of Liars

This dataset requires both a social and technical setup: finding a community that plays Diplomacy online and having them use a framework for annotating these messages.

6.3.1 Technical implementation

We need two technical components for our study: a game engine and a chat system. We choose Backstabbr[4] as an accessible game engine on desktop and mobile platforms: players input their moves and the site adjudicates game mechanics (Chiodini, 2020). Our communication framework is atypical. Thus, we create a server on Discord,[5] the group messaging platform most used for online gaming and by the online Diplomacy community (Coberly, 2019). The app is reliable on both desktop and mobile devices, free, and does not limit access to messages. Instead of direct communication, players communicate with a bot; the bot does not forward messages to the recipient until the player annotates the messages (Figure 6.2). In addition, the bot scrapes the game state from Backstabbr to sync game and language data.

Annotation of lies is a forced binary choice in our experiment. Explicitly calling a statement a lie is difficult, and people would prefer degrees of deception (Bavelas et al., 1990; Bell and DePaulo, 1996). Thus, we follow previous work that views linguistic deception as binary (Buller et al., 1996; Braun and Van Swol, 2016). Some studies make a more fine-grained distinction; for example, Swol et al. (2012) separate strategic omissions from blatant lies (we consider both deception). However, because we are asking the speakers themselves (and not trained annotators) to make the decision, we follow the advice from crowdsourcing to simplify the task as much as possible (Snow et al., 2008; Sabou et al., 2014). Long messages can contain both truths and lies, and we ask players to categorize these as lies since the truth can be a shroud for their aims.

6.3.2 Building a player base

The Diplomacy players maintain an active, vibrant community through real-life meetups and online play (Hill, 2014; Chiodini, 2020). We recruit top players alongside inexperienced but committed players in the interest of having a diverse pool. Our experiments include top-ranked players and community leaders from online platforms, grizzled in-person tournament players with over 100 past games, and board game aficionados. These players serve as our foundation and during

[3]Data available at `http://go.umd.edu/diplomacy_data` and as part of ConvoKit `http://convokit.cornell.edu`.

[4]`https://www.backstabbr.com`

[5]`https://www.discord.com`

Category	Value
Message Count	13,132
ACTUAL LIE Count	591
SUSPECTED LIE Count	566
Average # of Words	20.79

Table 6.2: Summary statistics for our train data (nine of twelve games). Messages are long and only five percent are lies, creating a class imbalance.

initial design helped us to create a minimally annoying interface and a definition of a lie that would be consistent with Diplomacy play. Good players—as determined by active participation, annotation and game outcome—are asked to play in future games.

In traditional crowdsourcing tasks compensation is tied to piecework that takes seconds to complete (Buhrmester et al., 2011). Diplomacy games are different in that they can last a month. . . and people already play the game for free. Thus, we do not want compensation to interfere with what these players already do well: lying. Even the obituary of the game's inventor explains

> Diplomacy rewards all manner of mendacity: spying, lying, bribery, rumor mongering, psychological manipulation, outright intimidation, betrayal, vengeance and backstabbing (the use of actual cutlery is discouraged)" (Fox, 2013).

Thus, our goal is to have compensation mechanisms that get people to play this game as they normally would, finish their games, and put up with our (slightly) cumbersome interface. Part of the compensation is non-monetary: a game experience with players that are more engaged than the average online player.

To encourage complete games, most of the payment is conditioned on finishing a game, with rewards for doing well in the game. Players get at least $40 upon finishing a game. Additionally, we provide bonuses for specific outcomes: $24 for winning the game (an evenly divisible amount that can be split among remaining players) and $10 for having the most successful lies, i.e., statements they marked as a lie that others believed.[6] Diplomacy usually ends with a handful of players dividing the board among themselves and agreeing to a tie. In the game described in Section 6.2.1, the remaining four players shared the winner's pool with Italy after 10 in-game years, and Italy won the prize for most successful lies.

6.3.3 Data overview

Table 6.2 quantitatively summarizes our data. Messages vary in length and can be paragraphs long (Figure 6.3). Close to five percent of all messages in the

[6]The lie incentive is relatively small (compared to incentives for participation and winning) to discourage an opportunistic player from marking everything as a lie. Games were monitored in real-time and no player was found abusing the system (marking more than ∼20% lies).

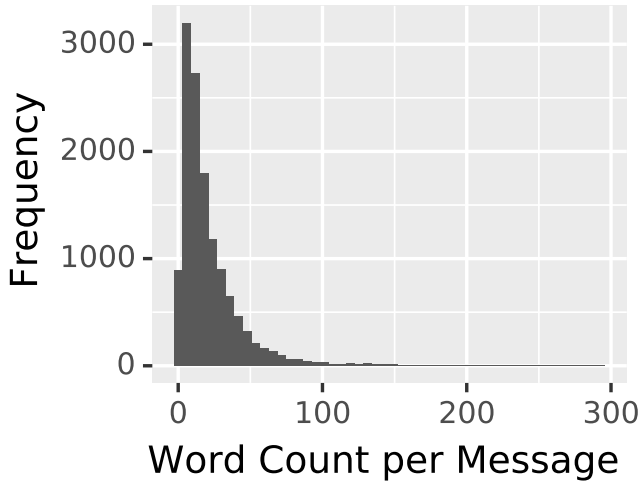

Figure 6.3: Individual messages can be quite long, wrapping deception in pleasantries and obfuscation.

dataset are marked as lies and almost the same percentage (but not necessarily the same messages) are perceived as lies, consistent with the "veracity effect" (Levine et al., 1999). In the game discussed above, eight percent of messages are marked as lies by the sender and three percent of messages are perceived as lies by the recipient; however, the messages perceived as lies are rarely lies (Figure 6.4).

6.3.4 Demographics and self-assessment

We collect anonymous demographic information from our study participants: the average player identifies as male, between 20 and 35 years old, speaks English as their primary language, and has played over fifty Diplomacy games.[7] Players self-assess their lying ability before the study. The average player views themselves as better than average at lying and average or better than average at perceiving lies.

In a post-game survey, players provide information on whom *they* betrayed and who betrayed *them* in a given game. This is a finer-grained determination than the *post hoc* analysis used in past work on Diplomacy (Niculae et al., 2015). We ask players to optionally provide linguistic cues to their lying and to summarize the game from their perspective.

6.3.5 An ontology of deception

Four possible combinations of deception and perception can arise from our data. The sender can be lying or telling the truth. Additionally, the receiver can perceive the message as deceptive or truthful. We name the possible outcomes for lies

[7] Our data skews 80% male and 95% of the players speak English as a primary language. Ages range from eighteen and sixty-four. Game experience is distributed across beginner, intermediate, and expert levels.

		Receiver's perception	
		Truth	**Lie**
Sender's intention	**Truth**	**Straightforward** Salut! Just checking in, letting you know the embassy is open, and if you decide to move in a direction I might be able to get involved in, we can probably come to a reasonable arrangement on cooperation. Bonne journee!	**Cassandra** I don't care if we target T first or A first. I'll let you decide. But I want to work as your partner. ...I literally will not message anyone else until you and I have a plan. I want it to be clear to you that you're the ally I want.
	Lie	**Deceived** You, sir, are a terrific ally. This was more than you needed to do, but makes me feel like this is really a long term thing! Thank you.	**Caught** So, is it worth us having a discussion this turn? I sincerely wanted to work something out with you last turn, but I took silence to be an ominous sign.

Table 6.3: Examples of messages that were intended to be truthful or deceptive by the sender or receiver. Most messages occur in the top left quadrant (Straightforward). Figure 6.4 shows the full distribution. Both the intended and perceived properties of lies are of interest in our study.

as Deceived or Caught, and the outcomes for truthful messages as Straightforward or Cassandra,[8] based on the receiver's annotation (examples in Table 6.3, distribution in Figure 6.4).

6.4 Detecting Lies

We build computational models both to detect lies to better understand our dataset. The data from the user study provide a training corpus that maps language to annotations of truthfulness and deception. Our models progressively integrate information—conversational context and in-game power dynamics—to approach human parity in deception detection.

6.4.1 Metric and data splits

We investigate two phenomena: detecting what is *intended* as a lie and what is *perceived* as a lie. However, this is complicated because most statements are not lies: less than five percent of the messages are labeled as lies in both the ACTUAL LIE and the SUSPECTED LIE tasks (Table 6.2). Our results use a weighted F_1 feature across truth and lie prediction, as accuracy is an inflated metric given the class imbalance (Japkowicz and Stephen, 2002). We thus adopt an in-training approach (Zhou and Liu, 2005) where incorrect predictions of lies are penalized more than truth-

[8] In myth, Cassandra was cursed to utter true prophecies but never be believed. For a discussion of Cassandra's curse *vis a vis* personal and political oaths, see Torrance (2015).

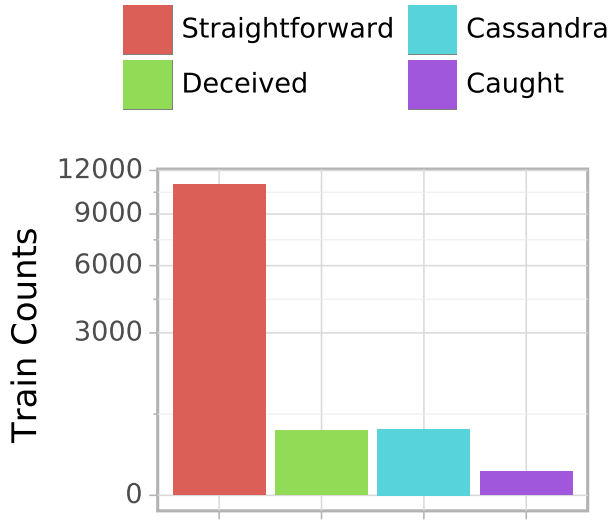

Figure 6.4: Most messages are truthful messages identified as the truth. Lies are often not caught. Table 6.3 provides an example from each quadrant.

ful statements. The relative penalty between the two classes is a hyper-parameter tuned on F_1.

Before we move to computational models for lie detection, we first establish the *human* baseline. We know when senders were lying and when receivers spotted a lie. Humans spot 88.3% of lies. However, given the class imbalance, this sounds better than it is. Following the suggestion of (Levine et al., 1999), we focus on the detection of lies, where humans have a 22.5 Lie F_1.

To prevent overfitting to specific games, nine games are used as training data, one is used for validation for tuning parameters, and two games are test data. Some players repeat between games.

6.4.2 Logistic regression

Logistic regression models, described in Background Section 2.4.1, have interpretable coefficients which show linguistic phenomena that correlate with lies. A *word* that occurs infrequently overall but often in lies, such as 'honest' and 'candidly', helps identify which messages are lies.

(Niculae et al., 2015) propose linguistic **Harbingers** that can predict deception. These are word lists that cover topics often used in interpersonal communication—*claims*, *subjectivity*, *premises*, *contingency*, *comparisons*, *expansion*, *temporal language associated with the future*, and *all other temporal language* (complete word list in Appendix, Table **??**). The Harbingers word lists do not provide full coverage, as they focus on specific rhetorical areas. A logistic regression model with all word types as features further improves F_1.

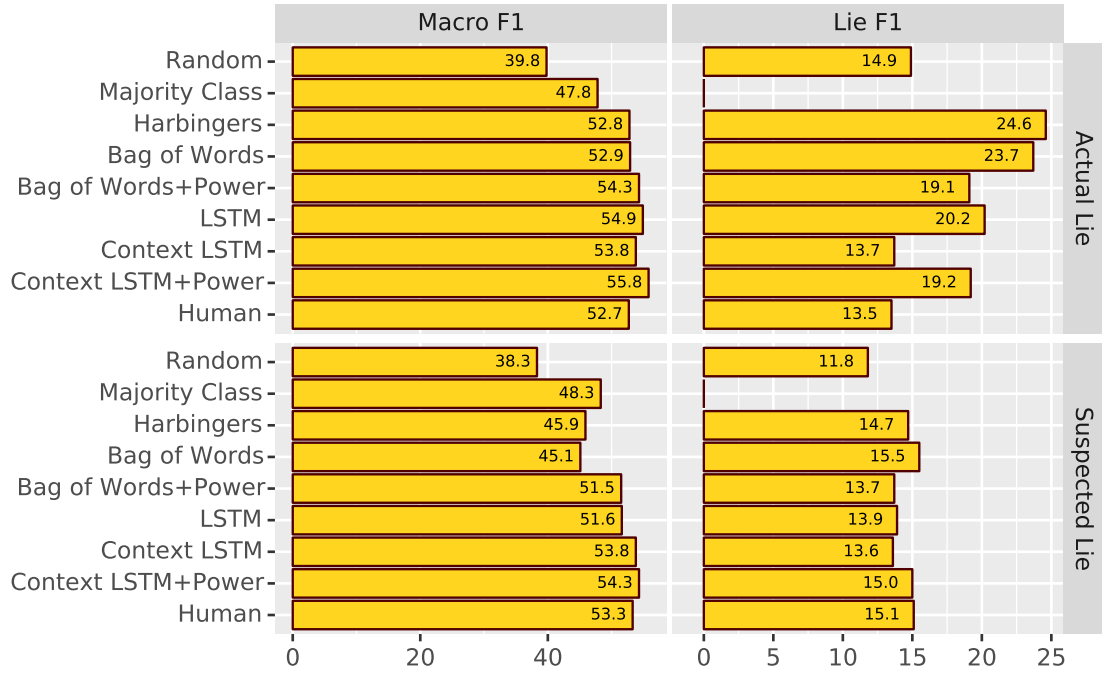

Figure 6.5: Test set results for both our ACTUAL LIE and SUSPECTED LIE tasks. We provide baseline (Random, Majority Class), logistic (language features, bag of words), and neural (combinations of a LSTM with BERT) models. The neural model that integrates past messages and power dynamics approaches human F_1 for ACTUAL LIE (top). For ACTUAL LIE, the human baseline is how often the receiver correctly detects senders' lies. The SUSPECTED LIE lacks such a baseline.

Power dynamics influence the language and flow of conversation (Danescu-Niculescu-Mizil et al., 2012, 2013; Prabhakaran et al., 2013). These dynamics may influence the likeliness of lying; a stronger player may feel empowered to lie to their neighbor. Recall that victory points (Section 6.2) encode how well a player is doing (more is better). We represent the power differential as the difference between the two players. Peers will have a zero differential, while more powerful players will have a positive differential with their interlocutor. The differential changes throughout the game, so this feature encodes the difference in the season the message was sent. For example, a message sent by an Italy with seven points to a Germany with two points in a given season would have a value of five.

6.4.3 Neural

While less interpretable, neural models are often more accurate than logistic regression ones (Ribeiro et al., 2016; Belinkov and Glass, 2019). We build a standard long short-term memory network (Hochreiter and Schmidhuber, 1997, LSTM), described in Background Section 2.4.4, to investigate if word sequences—ignored by logistic regression—can reveal lies.

		Model Prediction	
		Correct	**Wrong**
Player Prediction	**Correct**	**Both Correct** Not sure what your plan is, but I might be able to support you to Munich.	**Player Correct** Don't believe Turkey, I said nothing of the sort. I imagine he's just trying to cause an upset between us.
	Wrong	**Model Correct** Long time no see. Sorry for the stab earlier. I think we should try to work together to stop france from winning; if we work together we can stop france from getting 3 more centers, and then we will all win in a 3, 4, or 5 way draw when the game is hard-capped at 1910.	**Both Wrong** I'm considering playing fairly aggressive against England and cutting them off at the pass in 1901, your support for that would be very helpful.

Table 6.4: An example of an ACTUAL LIE detected (or not) by both players and our best computational model (Context LSTM + Power) from each quadrant. Both the model and the human recipient are mostly correct overall (Both Correct), but they are both mostly wrong when it comes to specifically predicting lies (Both Wrong).

Integrating message context and power dynamics improves on the neural baseline. A Hierarchical LSTM can help focus attention on specific phrases in long conversational contexts. In the same way it would be difficult for a human to determine *prima facie* if a statement is a lie without previous context, we posit that methods that operate at the level of a single message are limited in the types of cues they can extract. The hierarchical LSTM is given the context of previous messages when determining if a given message is a lie, which is akin to the labeling task humans do when annotating the data. The model does this by encoding a single message from the tokens, and then running a forward LSTM over all the messages. For each message, it looks at both the content and previous context to decide if the current message is a lie. Fine-tuning BERT (Devlin et al., 2019) embeddings, introduced in Background Section 2.4.5, to this model did not lead to notable improvement in F_1, likely due to the relative small size of our training data. Last, we incorporate information about power imbalance into this model. This model approaches human performance in terms of F_1 score by combining content with conversational context and power imbalance.

6.5 Qualitative Analysis

This section examines specific messages where both players and machines are correctly identifying lies and when they make mistakes on our test set. Most messages are correctly predicted by both the model and players (2055 of 2475 messages); but this is because of the veracity effect. The picture is less rosy if we only look at messages the sender marks as ACTUAL LIE: both players and models are generally

	Model Correct	Model Wrong
Player Correct	10	32
Player Wrong	28	137

Table 6.5: Conditioning on only lies, most messages are now identified incorrectly by both our best model (Context LSTM + Power) and players.

wrong (Table 6.5).

Both models and players can detect lies when liars get into specifics. In Diplomacy, users must agree to help one another through orders that stipulate "I will help another player move from X to Y". The in-game term for this is "support"; half the messages where players and computers correctly identify lies contain this word, but it rarely occurs in the other quadrants.

Models seem to be better at not falling for vague excuses or fantastical promises in the future. Players miss lies that promise long-term alliances, involve extensive apologies, or attribute motivation as coming from other countries' disinformation (*Model Correct*). Unlike our models, players have access to conversations with other players and accordingly players can detect lies that can easily be verified through conversations with other players (*Player Correct*).

However, ultimately most lies are believable and fool both models and players (*Both Wrong*). For example, all messages that contain the word "true" are predicted as truthful by both models and players. Many of these messages are relatively tame;[9] confirming the Pinocchio effect found by Swol et al. (2012). If liars can be detected when they wax prolix, perhaps the best way to avoid detection is to be terse and to the point.

Sometimes additional contextual information helps models improve over player predictions. For example, when France tells Austria "I am worried about a steamroller Russia Turkey alliance", the message is incorrectly perceived as truthful by both the player and the single-message model. However, once the model has context—a preceding question asking if Austria and Turkey were cooperating—it can detect the lie.

Finally, we investigate categories from the Harbingers (Niculae et al., 2015) word lists. Lies are more likely to contain *subjectivity* and *premises* while true messages include *expansion* phrases ("later", "additionally"). We also use specific words in the bag of words logistic regression model. The coefficient weights of words that express sincerity (e.g., "sincerely", "frankly") and apology (e.g., "accusation", "fallout", "alternatives") skew toward ACTUAL LIE prediction in the logistic regression model. More laid back appellations (e.g., "dude", "man") skew towards truthfulness, as do words associated with reconnaissance (e.g., "fyi","useful", "information") and time (e.g., "weekend", "morning"). Contested areas on the Diplomacy map, such

[9]Examples include "It's true—[Budapest] back to [Rumania] and [Serbia] on to [Albania] could position for more forward convoys without needing the rear fleet..." and "idk if it's true just letting u know since were allies".

as Budapest and Sevastopol, are more likely to be associated with lies, while more secure ones like Berlin, are more likely to be associated with truthful messages.

6.6 Related Work

Early computational deception work focuses on single utterances (Newman et al., 2003), especially for product reviews (Ott et al., 2012). But deception is intrinsically a discursive phenomenon and thus the context in which it appears is essential. Our platform provides an opportunity to observe deception in the context in which it arises: goal-oriented conversations around in-game objectives. Gathering data through an interactive game has a cheaper per-lie cost than hiring workers to write deceptive statements (Jurgens and Navigli, 2014).

Other conversational datasets are mostly based on games that involve deception including Werewolf (Girlea et al., 2016), Box of Lies (Soldner et al., 2019), and tailor-made games (Ho et al., 2017). However, these games assign individuals roles that they maintain throughout the game (i.e., in a role that is supposed to deceive or in a role that is deceived). Thus, deception labels are coarse: an individual always lies or always tells the truth. In contrast, our platform better captures a more multi-faceted reality about human nature: everyone can lie or be truthful with everyone else, and they use both strategically. Hence, players must think about *every* player lying at any moment: "given the evidence, do I think this person is lying to me now?"

Deception data with conversational labels is also available through interviews (Pérez-Rosas et al., 2016), some of which allow for finer-grained deception spans (Levitan et al., 2018). Compared with game-sourced data, however, interviews provide shorter conversational context (often only a single exchange with a few follow-ups) and lack a strategic incentive—individuals lie because they are instructed to do so, not to strategically accomplish a larger goal. In Diplomacy, users have an intrinsic motivation to lie; they have entertainment-based and financial motivations to win the game. This leads to higher-quality, creative lies.

Real-world examples of lying include perjury (Louwerse et al., 2010), calumny (Fornaciari and Poesio, 2013), emails from malicious hackers (Dhamija et al., 2006), and surreptitious user recordings. But real-world data comes with real-world complications and privacy concerns. The artifice of Diplomacy allows us to gather pertinent language data with minimal risk and to access both sides of deception: intention and perception. Other avenues for less secure research include analyzing dating profiles for accuracy in self-presentation (Toma and Hancock, 2012) and classifying deceptive online spam (Ott et al., 2011).

6.7 Conclusion

In Dante's *Inferno*, the ninth circle of Hell—a fate worse even than that reserved for murderers—is for betrayers. Dante asks Count Ugolino to name his betrayer, which leads him to say:

> but if my words can be the seed to bear
> the fruit of infamy for this betrayer
> who feeds my hunger, then I shall speak—in tears (Alighieri and Musa, 1995, Canto XXXIII)

Similarly, we ask victims to expose their betrayers in the game of Diplomacy. The seeds of players' negotiations and deceit could, we hope, yield fruit to help others: understanding multi-party negotiation and protecting Internet users.

While we ignore nuances of the game board to keep our work general, Diplomacy is also a rich, multi-agent strategic environment; (Paquette et al., 2019) ignore Diplomacy's rich language to build bots that only move pieces around the board. An exciting synthesis would incorporate deception and language generation into an agent's policy; our data would help train such agents. Beyond playing against humans, playing with a human in the loop (HITL) resembles designs for cyber-security threats (Cranor, 2008), annotation (Branson et al., 2010), and language alteration (Wallace et al., 2019b). Likewise, our lie-detection models can help a user in the moment better decide whether they are being deceived (Lai et al., 2020). Computers can meld their attention to detail and nigh infinite memory to humans' grasp of social interactions and nuance to forge a more discerning player.

Beyond a silly board game, humans often need help verifying claims are true when evaluating health information (Xie and Bugg, 2009), knowing when to take an e-mail at face value (Jagatic et al., 2007), or evaluating breaking news (Hassan et al., 2017). Building systems to help information consumers become more discerning and suspicious in low-stakes settings like online Diplomacy are the seeds that will bear the fruits of interfaces and machine learning tools necessary for a safer and more robust Internet ecosystem.

In contrast to Chapter 3 and Chapter 4, this dataset is created exclusively with expert users, in this case Diplomacy players. While there are quality differences even within a verified pool of community-of-interest, only one out of 80 users did not actively participate in the experiment. In contrast over 10% of the data was duplicated by crowd-sourced workers in Chapter 4. Additionally, we find the *generated* data to be thoughtful, clever, and sometimes even funny, which are adjectives that seldom apply to large-scale NLP datasets. Both the *generation* and *annotation* for this task would not be possible without experts.

In Chapter 7, we propose to create an expert-dependent task for another subfield of NLP, machine translation. In addition, we will see if a large crowd-sourced dataset, WikiData, provides higher quality predictions than automatically created embeddings for this task.

Chapter 7: Proposed Work

Our past work has established that experts can solve tasks not possible by generalists. We propose a new task where the gold standard is integral for evaluation, thereby requiring experts. Machine translation usually translates words literally; however, this does not necessarily apply in a cultural context. Certain Named Entities may be relevant in one culture but not another. One can find applicable Named Entity modulations by referencing WikiData, a human-interpretable and human-verified representation of Wikipedia. We will want to investigate if this method generates better candidates than an embedding-based approach, such as word2vec. And a genuine evaluation of this approach requires specialized users, specifically German nationals that would understand the language and culture.

7.1 Using Cultural Experts for Translation

Chapter 4 proposes a method to evaluate machine translation models and in turn data. If we can establish that neural models are shallow in their understanding of a task, we should be able to establish that current auto-generated or crowd-sourced datasets are insufficient in quality. How then can we generate data at scale, but with a level of reliability? Modulation is a task that combines our past work in Question Answering, with our proposed work in Machine Translation and is a good, difficult test-bed. This project posits two questions about experts. Can relying on *human-verified* datasets, specifically WikiData, set a higher standard for machine translation of question answering than is now possible? Additionally, how do you verify that a generative task with many possible options is providing a reasonable answer?

A challenge for modern data-hungry natural language processing (NLP) techniques is to replicate the impressive results for standard English tasks and datasets to other languages. Literally translating text into the target language is the most obvious solution. This can be the best option for tasks such as sentiment analysis (Araujo et al., 2016), but for other tasks such as question answering (QA), literal translations might miss cultural nuance if you directly translate questions from English to German to provide additional training data. While this might allow QA systems to answer questions about baseball and Tom Hanks in German, it does not fulfill the promise of a smart assistant answering a culturally-situated question about Oktoberfest.

This alternative is called cultural adaptation. If you put a German sentence into a translation system, you might get literal, correct translation like "Mr. Müller

grabbed a Berliner from Dietsch at the Hauptbahnhof before jumping on the ICE". The cultural context of Germany is necessary to understand this example.

An extreme adaptation could render the sentence as "Mr. Miller grabbed a Boston Cream from the Dunkin' Donuts in Grand Central before jumping on the Acela", elucidating that Müller literally means "Miller", that Dietsch (like Dunkin' Donuts) is a mid-range purveyor of baked goods, both Berliners and Boston Creams are filled sweet pastries named after a city, and that the ICE is the (slightly) ritzier inter-city train. Humans translators use this type of adaptation frequently when it is appropriate to the translation.

Because adaptation is understudied, we leave the full translation task to future work. Instead, we focus on the task of cultural adaptation of entities: given an entity in English, what is the corresponding entity in a target language. For example, the German Anthony Fauci is Christian Drosten. Can machines reliably find these analogs with minimal supervision?

7.2 Was ist *George Washington*?

This section defines cultural adaptation and motivates it application for tasks like creating culturally-centered training data for QA. Vinay and Darbelnet (1995) define adaptation as translation in which the relationship, and not the literal meaning, between the receiver and the content is recreated.

Work on analogy is close to our interest, but the standard analogy set-up lacks the cross-cultural and cross-lingual dimensions (Turney, 2008; Gladkova et al., 2016). Additionally, recent methods for identifying entities or cross-lingual translation could be repurposed for adaptation (Duh et al., 2011; Schnabel et al., 2015; Kasai et al., 2019; Arora et al., 2019; Kim et al., 2019; Hangya and Fraser, 2019)

Adaptation is most applicable when machine translation is combined with other tasks. Non-literal translation would be harmful for certain tasks such as the information retrieval of news stories. In contrast, question answering is one domain where adaptation seems crucial. There has been an explosion of English-language QA data, but not in other languages. Several approaches try to transfer English's bounty to other languages. MLQA and XQuAD generate questions through machine translation (Lewis et al., 2019; Artetxe et al., 2019). TyDi (Clark et al., 2020a) gives users prompts from Wikipedia articles; other datasets like SQuAD recapitulate the problematic distribution of encyclopedias (Reagle and Rhue, 2011).

Most of the entities asked about in major QA datasets—SQuAD, TriviaQA, Quizbowl—are American. The coverage of the question remains the same across languages.

Given that we already have professionally-written questions, can we adapt, rather than literally generate, them to another culture and language?

7.3 Adaptation from a Knowledge Base

We first adapt entities using a knowledge base. We use WikiData (Vrandečić and Krötzsch, 2014), a structured, human-annotated representation of Wikipedia that is actively developed. This resource is well-suited to the task, particularly as features are standardized both within and across languages.

Many knowledge bases explicitly encode the nationality of individuals, places, and creative works.[1] Entities are represented in knowledge bases as discrete sparse vectors, where most dimensions are unknown or not applicable (e.g., a building do not have a spouse). For example, Angela Merkel is a human (instance of), German (country of citizenship), politician (occupation), Rotarian (member of), Lutheran (religion), 1.65 meters tall (height), and has a PhD (academic degree). How would we find the "most similar" American adaptation to Angela Merkel? Intuitively, we should find someone whose nationality is American.

Some issues immediately present themselves; contemporary entities will have more non-zero entries than older entities. Some characteristics are more important than others: matching unique attributes like "worked as journalist" is more important than matching "is human".

The items can be grouped by *property* and by *value*, the WikiData equivalent of intents and slots. *Properties* in WikiData are the abstract intents: Merkel has an "occupation", a "academic degree". *Values* are the slots: her "occupation" is "politician", her "academic degree" is a "doctorate". The former works for macro-entity classification since a building, a person and a song have different properties. Additionally, more popular items have more properties. The latter are useful *within* a culture as Merkel will belong to a *value* like the Christian Democratic Union, unlike an American politician.

First, we bifurcate the WikiData into two sets: an American set $\mathcal{A}$ for items which contain the *value* "United States of America" and a German set $\mathcal{D}$ for those with German values.[2] This is a liberal approximation, but it successfully excludes roughly seven out of the eight million items in WikiData. Then we explore the *properties* and the *values* from the WikiData. *Properties* are limited and centrally organized. *Values* are more numerous and varying in quality. We select the highest frequency features.[3] Values exist in all types of dimensions and the structure of WikiData is occasionally inconsistent. For example, you will not find Goethe under any expected variations of Germany; he is only annotated under Saxe-Weimar-Eisenach. Including additional values does not lead to qualitatively better predictions with 20,000 values than with 1,000 values. We use *properties* for our final results.

[1] Like with language, nationality is often correlated with culture, but is not synonymous. Large countries contain multitudes, while some nationalities (e.g., Kurds) lack a *de jure* nation but span many nations. We elide this detail and focus on information often available in knowledge bases.

[2] While the geopolitical definition of American is straightforward, the German nation state is more nuanced (Schulze, 1991). Following Green (2003), we adopt members of the Zollverein or the German Confederation as "German" as well as their prececessor and sucessor states.

[3] Including a maximum and a minimum cap did not obviously generate better candidates than the most frequent items

The *properties* are discrete and categorical; Merkel either has an "occupation" or she does not. Each entity then has a sparse vector. We calculate the similarity of the vectors with Faiss's (Johnson et al., 2017) L2 distance. Specifically, we search for each of the source German adaptation entities in the pre-selected 1,000,000 item American matrix. Conversely we search for each of the American entities in the pre-selected 180,000 item German matrix. This division is crucial as the most similar candidates are from the same cultural background.

Formally we calculate the vector as:

$$d' = \arg\min_{a \in \mathcal{A}} \|a - d\|^2 \tag{7.1}$$

where d' is the optimal German vector and $a \in \mathcal{A}$ are the items in the American matrix.

For both WikiData and the embedding-based approach, we select 100 candidates per item.

So who is the American Angela Merkel? One possible answer is Woodrow Wilson, a blue-eyed protestant who had a PhD, served as head of state, and was also nominated for a Nobel Peace Prize. This answer may be unsatisfying as it was Barack Obama who sat across from Merkel for nearly a decade. To capture these more nuanced similarities, we turn to large text corpora in Section **??**.

While the classic NLP vector example (Mikolov et al., 2013c) isn't as magical as initially claimed (Rogers et al., 2017), it provides useful intuition. We can use the intuitions of the cliché:

$$\overrightarrow{\text{King}} - \overrightarrow{\text{Man}} + \overrightarrow{\text{Woman}} = \overrightarrow{\text{Queen}} \tag{7.2}$$

to adapt between languages. We follow the word analogy approach of 3CosAdd (Levy and Goldberg, 2014; Köper et al., 2016) to adapt the source word by solving:

$$x - \overrightarrow{\text{American}} + \overrightarrow{\text{German}} = \overrightarrow{\text{Merkel}} \tag{7.3}$$

to find the closest entity, Obama, to x.[4]

Towards this end, we will need to create relevant embeddings. First, we use Wikipedia dumps in the English and German language, processed using Moses' preprocessing pipeline (Koehn et al., 2007). However, by default, the dumps are separated as unigrams, whereas Named Entities such as people are often phrases. We follow Mikolov et al. (2013b) and use co-occurrence statistics to build bigrams and trigrams, limiting the vocabulary to the 1M most frequent tokens. We use word2vec (Mikolov et al., 2013b), rather than FastText (Bojanowski et al., 2016), as we do not want orthography to influence the similarity of entities. Merkel in English and in German have quite different neighbors, and we intend to keep it that way.

However, the standard word2vec model assumes a single monolingual embedding space. To align the two monolingual spaces we use unsupervised Vecmap (Artetxe et al., 2018), a leading tool for cross-lingual word embeddings. American→German

[4]We experimented with 3CosMul as well but found 3CosAdd generally more robust.

can be thought of as representing the source embedding in the American space and the target embedding in the German space. Hence, the source (American) becomes x in this equation, meaning that x-a+b represents its adapted vector and the closest target words (German) based on cosine similarity its word adaptations. a and b represent the American and German culture and are used as anchors for the adaptation. We average the vector of United States in the English space and that of USA in the German space for robustness. Similarly we average Germany and Deutschland for vector b. In standard analogy the a and b vectors are different for each test pair. In our case, the vectors are the same because the relation is identical for each x-y pair.

Summarizing, we take the German (or American) embedding of the Named Entity, adapt it with 3CosAdd and look for the most similar words to the adapted embeddings in the American (or German) model. In the case where the phrase is not found as an embedding, we back off to the last name of the named entity (e.g., Barack Obama → Obama).

7.4 Evaluation by Experts

The difficulty of the task merits skilled users. Since quality control is difficult for generation (Peskov et al., 2019), we need users who will answer the task accurately and without annotation artifacts. We select five American citizens educated at American universities and five German citizens educated at German university. These human annotations serve as a gold standard against which we can compare our automated approaches. To improve the user experience, we create a custom interface that:

1. describes the task and provides examples
2. tracks the user inputting the annotation
3. provides a brief summary from Wikipedia
4. pre-populates from an autocomplete box *a la* answer selection in Wallace et al. (2019c)

The annotation task requires roughly two hours for our users to complete. Our entities come from two sources: the top 500 most visited Wikipedia pages and the Veale NOC List (Veale, 2016). Wikipedia has a heavy skew towards pop culture; the top 500 pages had to be preemptively filtered to avoid being dependent on pop music and films. The Veale NOC list is human-verified and contains a historically broader sweep of people. We conduct this exercise in both directions; while Berlin is the German Washington, DC, there is less consensus on what is the American Berlin, as Berlin is both the capital, a tech hub, and a film hub. A full list of our items and their suggested adaptations are in the Appendix. We expect this dataset to show how prototypical particular examples are within a culture.

7.4.1 Summary

We propose entity adaptation as a task. Word2vec embeddings and WikiData can be used to figuratively—not just literally—translate entities into a different culture. We are interested in knowing if both methods generate reasonable candidates. WikiData is largely human-verified and will test if crowd-sourced information is more similar to expert decision-making than automatic embeddings. Additionally, we will see how interpretable our predictions are. For our experiments we will create and release the first adaptation dataset for which citizens of the respective countries provide annotations for popular items from English and German Wikipedia, and a part of the Veale Non-Official Characterization list.

Chapter 8: Conclusion

In this proposal, we have covered past work that creates datasets using three types of data pools: unspecialized, hybrid, and expert. We argue that improving data quality with reliable data generators and annotators is paramount towards establishing new NLP tasks. We propose a new task, cultural adaptation, that both passively evaluates a crowd-sourced data source, WikiData, while using verified cultural experts to create the gold standard.

8.1 Timeline

- **Late January/Early February** Thesis Proposal
- **February 2021** Modulation Paper Updates
- **June 2021** Modulation Paper presented at NAACL
- **January till June 2021** Assistance with other projects on Question Answering and Diplomacy
- **August 2021** Thesis Defense
- **September 2021** Transition to academic postdoc

Chapter 9: Reading List

9.1 Crowd-Sourcing

1. Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y Ng. 2008. Cheap and fast—but is it good?: Evaluating non-expert annotations for natural language tasks. In Proceedings of Empirical Methods in Natural Language Processing

2. J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In 2009 IEEE Conference on Computer Vision and Pattern Recognition, pages 248–255

3. Timothy W. Finin, William Murnane, Anand Karandikar, Nicholas Keller, Justin Martineau, and Mark Dredze. 2010. Annotating named entities in twitter data with crowdsourcing. In Mturk@HLT-NAACL

4. Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. arXiv preprint arXiv:1810.00278

5. Michael Buhrmester, Tracy Kwang, and Samuel D Gosling. 2011. Amazon's mechanical Turk: A new source of inexpensive, yet high-quality data? Perspectives on psychological science: a journal of the Association for Psychological Science, 6 1:3–5

6. Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019a. Universal adversarial triggers for attacking and analyzing nlp. arXiv preprint arXiv:1908.07125

7. Bill Byrne, Karthik Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Daniel Duckworth, Semih Yavuz, Ben Goodrich, Amit Dubey, Kyu-Young Kim, and Andy Cedilnik. 2019. Taskmaster-1: Toward a realistic and diverse dialog dataset

8. Sue Atkins, Jeremy Clear, and Nicholas Ostler. 1992. Corpus design criteria. Literary and linguistic computing, 7(1):1–16

9. Eric Schenk and Claude Guittard. 2011. Towards a characterization of crowdsourcing practices. Journal of Innovation Economics Management, (1):93–107

10. Djellel Difallah, Elena Filatova, and Panos Ipeirotis. 2018. Demographics and dynamics of mechanical turk workers. In Proceedings of the eleventh ACM international conference on web search and data mining, pages 135–143

9.2 Question Answering

1. Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In Proceedings of Empirical Methods in Natural Language Processing

2. Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018b. Know what you don't know: Unanswerable questions for SQuAD. In Proceedings of the Association for Computational Linguistics

3. Matthew Dunn, Levent Sagun, Mike Higgins, V. Ugur Güney, Volkan Cirik, and Kyunghyun Cho. 2017b. Searchqa: A new Q&A dataset augmented with context from a search engine. CoRR, abs/1704.05179

4. Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke S. Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Proceedings of the Association for Computational Linguistics

5. Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. arXiv preprint arXiv:1611.09268

6. Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. In Proceedings of Empirical Methods in Natural Language Processing

7. Eric Wallace, Pedro Rodriguez, Shi Feng, Ikuya Yamada, and Jordan Boyd-Graber. 2019b. 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

8. Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020a. Tydi qa: A benchmark for information-seeking question answering in typologically diverse languages. In Transactions of the Association for Computational Linguistics

9. Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. Coqa: A conversational question answering challenge. Transactions of the Association for Computational Linguistics, 7:249–266

10. Jordan Boyd-Graber. 2020. What question answering can learn from trivia nerds. In Proceedings of the Association for Computational Linguistics

9.3 Model Interpretability

1. Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners

2. Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist

3. Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence, 1(5):206–215

4. Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems. In Advances in neural information processing systems, pages 3266–3280

5. Tal Linzen. 2020. How can we accelerate progress towards human-like linguistic generalization? In Proceedings of the Association for Computational Linguistics

6.

7. Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, et al. 2020. Evaluating models' local decision boundaries via contrast sets. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, pages 1307–1323

Bibliography

Abejide Olu Ade-Ibijola, Ibiba Wakama, and Juliet Chioma Amadi. 2012. An expert system for automated essay scoring (aes) in computing using shallow nlp techniques for inferencing. International Journal of Computer Applications, 51(10).

Dante Alighieri and Mark Musa. 1995. Dante's Inferno: The Indiana Critical Edition. Indiana masterpiece editions. Indiana University Press.

Amazon. 2021. Amazon Mechanical Turk. `http://www.mturk.com/`. [Online; accessed 03-January-2021].

Pranav Anand, Joseph King, Jordan Boyd-Graber, Earl Wagner, Craig Martell, Douglas W. Oard, and Philip Resnik. 2011. Believe me: We can do this! In The AAAI 2011 workshop on Computational Models of Natural Argument.

Matheus Araujo, Julio Reis, Adriano Pereira, and Fabricio Benevenuto. 2016. An evaluation of machine translation for multilingual sentence-level sentiment analysis. In Proceedings of the 31st Annual ACM Symposium on Applied Computing, SAC '16, page 1140–1145, New York, NY, USA. Association for Computing Machinery.

Ravneet Arora, Chen-Tse Tsai, Ketevan Tsereteli, Prabhanjan Kambadur, and Yi Yang. 2019. A semi-Markov structured support vector machine model for high-precision named entity recognition. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5862–5866, Florence, Italy. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2019. On the cross-lingual transferability of monolingual representations. CoRR, abs/1910.11856.

Layla El Asri, Hannes Schulz, Shikhar Sharma, Jeremie Zumer, Justin Harris, Emery Fine, Rahul Mehrotra, and Kaheer Suleman. 2017. Frames: A corpus for adding memory to goal-oriented dialogue systems. arXiv preprint arXiv:1704.00057.

Sue Atkins, Jeremy Clear, and Nicholas Ostler. 1992. Corpus design criteria. Literary and linguistic computing, 7(1):1–16.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. arXiv preprint arXiv:1611.09268.

David Bamman and Noah A. Smith. 2015. Contextualized Sarcasm Detection on Twitter. In Proceedings of ICWSM.

Juan M Banda, Ramya Tekumalla, Guanyu Wang, Jingyuan Yu, Tuo Liu, Yuning Ding, and Gerardo Chowell. 2020. A twitter dataset of 150+ million tweets related to covid-19 for open research. Type: dataset.

Janet Beavin Bavelas, Alex Black, Nicole Chovil, and Jennifer Mullett. 1990. Truths, lies, and equivocations: The effects of conflicting goals on discourse. Journal of Language and Social Psychology, 9(1-2):135–161.

Rachel Bawden, Rico Sennrich, Alexandra Birch, and Barry Haddow. 2018. Evaluating discourse phenomena in neural machine translation. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 1304–1313, New Orleans, Louisiana. Association for Computational Linguistics.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In Proceedings of the International Conference on Learning Representations.

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. Transactions of the Association for Computational Linguistics.

Kathy L Bell and Bella M DePaulo. 1996. Liking and lying. Basic and Applied Social Psychology, 18(3):243–266.

Adam Berger, Peter F Brown, Stephen A Della Pietra, Vincent J Della Pietra, John R Gillett, John Lafferty, Robert L Mercer, Harry Printz, and Lubos Ures. 1994. The candide system for machine translation. In HUMAN LANGUAGE TECHNOLOGY: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994.

Jean Berko. 1958. The child's learning of english morphology. Word, 14(2-3):150–177.

William E. Bogner, Margaret Edwards, Leon Zelechowski, Kevin J. Egan, William J. Rogers, Eloy Burciaga, and John Scott Arthur. 1974. Perjury: The forgotten offense. The Journal of Criminal Law and Criminology, 65(3):361–372.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606.

Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2016. Learning end-to-end goal-oriented dialog. arXiv preprint arXiv:1605.07683.

Jordan Boyd-Graber. 2020. What question answering can learn from trivia nerds. In Proceedings of the Association for Computational Linguistics.

Jordan Boyd-Graber, Shi Feng, and Pedro Rodriguez. 2018. Human-Computer Question Answering: The Case for Quizbowl. Springer Verlag.

Steve Branson, Catherine Wah, Florian Schroff, Boris Babenko, Peter Welinder, Pietro Perona, and Serge Belongie. 2010. Visual recognition with humans in the loop. In European Conference on Computer Vision.

Michael T. Braun and Lyn M. Van Swol. 2016. Justifications offered, questions asked, and linguistic patterns in deceptive and truthful monetary interactions. Group Decision and Negotiation, 25(3):641–661.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014. Concreteness ratings for 40 thousand generally known english word lemmas. Behavior Research Methods, 46:904–911.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. arXiv preprint arXiv:1810.00278.

Michael Buhrmester, Tracy Kwang, and Samuel D Gosling. 2011. Amazon's mechanical Turk: A new source of inexpensive, yet high-quality data? Perspectives on psychological science: a journal of the Association for Psychological Science, 6 1:3–5.

David B. Buller, Judee K. Burgoon, Aileen Buslig, and James Roiger. 1996. Testing interpersonal deception theory: The language of interpersonal deception. Communication Theory, 6(3):268–289.

Bill Byrne, Karthik Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Daniel Duckworth, Semih Yavuz, Ben Goodrich, Amit Dubey, Kyu-Young Kim, and Andy Cedilnik. 2019. Taskmaster-1: Toward a realistic and diverse dialog dataset.

Chris Callison-Burch, Lyle Ungar, and Ellie Pavlick. 2015. Crowdsourcing for nlp. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorial Abstracts, pages 2–3.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. arXiv preprint arXiv:1708.00055.

Jesse J Chandler and Gabriele Paolacci. 2017. Lie for a dime: When most prescreening responses are honest but most study participants are impostors. Social Psychological and Personality Science, 8(5):500–508.

Jonathan P. Chang, Justin Cheng, and Cristian Danescu-Niculescu-Mizil. 2020. Don't let me be misunderstood: Comparing intentions and perceptions in online discussions. In Proceedings of the World Wide Web Conference.

Wendy W Chapman, Prakash M Nadkarni, Lynette Hirschman, Leonard W D'avolio, Guergana K Savova, and Ozlem Uzuner. 2011. Overcoming barriers to nlp for clinical text: the role of shared tasks and the need for additional creative solutions.

James Cheng, Monisha Manoharan, Yan Zhang, and Matthew Lease. 2015. Is there a doctor in the crowd? diagnosis needed! (for less than $5). iConference 2015 Proceedings.

Johnny Chiodini. 2020. Playing Diplomacy online transformed the infamously brutal board game from unbearable to brilliant. Dicebreaker.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. In Proceedings of Empirical Methods in Natural Language Processing.

Noam Chomsky. 1986. Knowledge of language: Its nature, origin, and use. Greenwood Publishing Group.

Robert B Cialdini and Noah J Goldstein. 2004. Social influence: Compliance and conformity. Annual Review of Psychology, 55:591–621.

Christopher Cieri, David Miller, and Kevin Walker. 2004. The fisher corpus: a resource for the next generations of speech-to-text. In Proceedings of the Language Resources and Evaluation Conference.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020a. Tydi qa: A benchmark for information-seeking question answering in typologically diverse languages. In Transactions of the Association for Computational Linguistics.

Jonathan H Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020b. Tydi qa: A benchmark for information-seeking question answering in typologically diverse languages. In Transactions of the Association for Computational Linguistics.

Kevin Clark and Christopher D. Manning. 2016a. Deep reinforcement learning for mention-ranking coreference models. In Empirical Methods on Natural Language Processing.

Kevin Clark and Christopher D. Manning. 2016b. Improving coreference resolution by learning entity-level distributed representations. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 643–653, Berlin, Germany. Association for Computational Linguistics.

Cohen Coberly. 2019. Discord has surpassed 250 million registered users. Techspot.

B. Cornwell and D. C. Lundgren. 2001. Love on the internet: involvement and misrepresentation in romantic relationships in cyberspace vs. realspace. Computational Human Behavior, 17:197–211.

Lorrie F Cranor. 2008. A framework for reasoning about the human in the loop. In UPSEC.

Cristian Danescu-Niculescu-Mizil, Lillian Lee, Bo Pang, and Jon Kleinberg. 2012. Echoes of power: Language effects and power differences in social interaction. In Proceedings of the World Wide Web Conference.

Cristian Danescu-Niculescu-Mizil, Moritz Sudhof, Dan Jurafsky, Jure Leskovec, and Christopher Potts. 2013. A computational approach to politeness with application to social factors. In Proceedings of the Association for Computational Linguistics.

Louise Deléger, Magnus Merkel, and Pierre Zweigenbaum. 2009. Translating medical terminologies through word alignment in parallel text corpora. Journal of Biomedical Informatics, 42(4):692–701.

J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In 2009 IEEE Conference on Computer Vision and Pattern Recognition, pages 248–255.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In Computer Vision and Pattern Recognition.

Bella M DePaulo, James J Lindsay, Brian E Malone, Laura Muhlenbruck, Kelly Charlton, and Harris Cooper. 2003. Cues to deception. Psychological bulletin, 129(1):74.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Conference of the North American Chapter of the Association for Computational Linguistics.

Rachna Dhamija, J. Doug Tygar, and Marti A. Hearst. 2006. Why phishing works. In International Conference on Human Factors in Computing Systems.

Djellel Difallah, Elena Filatova, and Panos Ipeirotis. 2018. Demographics and dynamics of mechanical turk workers. In Proceedings of the eleventh ACM international conference on web search and data mining, pages 135–143.

Chris Donahue, Bo Li, and Rohit Prabhavalkar. 2018. Exploring speech enhancement with generative adversarial networks for robust speech recognition. In IEEE International Conference on Acoustics, Speech and Signal Processing.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 1723–1732.

Kevin Duh, Akinori Fujino, and Masaaki Nagata. 2011. Is machine translation ripe for cross-lingual sentiment classification? In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pages 429–433.

Matthew Dunn, Levent Sagun, Mike Higgins, V. Ugur Güney, Volkan Cirik, and Kyunghyun Cho. 2017a. SearchQA: A new Q&A dataset augmented with context from a search engine. CoRR, abs/1704.05179.

Matthew Dunn, Levent Sagun, Mike Higgins, V. Ugur Güney, Volkan Cirik, and Kyunghyun Cho. 2017b. Searchqa: A new Q&A dataset augmented with context from a search engine. CoRR, abs/1704.05179.

Alfred Dürr. 2005. The cantatas of JS Bach: with their librettos in German-English parallel text. OUP Oxford.

Jeffrey L Elman. 1990. Finding structure in time. Cognitive science, 14(2):179–211.

David A. Ferrucci. 2010. Build Watson: an overview of DeepQA for the Jeopardy! challenge. In 19th International Conference on Parallel Architecture and Compilation Techniques, pages 1–2.

Elena Filatova, Vasileios Hatzivassiloglou, and Kathleen McKeown. 2006. Automatic creation of domain templates. In Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions, pages 207–214.

Timothy W. Finin, William Murnane, Anand Karandikar, Nicholas Keller, Justin Martineau, and Mark Dredze. 2010. Annotating named entities in twitter data with crowdsourcing. In Mturk@HLT-NAACL.

Ailbhe Finnerty, Pavel Kucherbaev, Stefano Tranquillini, and Gregorio Convertino. 2013. Keep it simple: Reward and task design in crowdsourcing. In Proceedings of the Biannual Conference of the Italian Chapter of SIGCHI, pages 1–4.

John R Firth. 1957. A synopsis of linguistic theory, 1930-1955. Studies in linguistic analysis.

Tommaso Fornaciari and Massimo Poesio. 2013. Automatic deception detection in Italian court cases. Artificial intelligence and law, 21(3):303–340.

Margalit Fox. 2013. Allan Calhamer dies at 81; invented Diplomacy game. New York Times.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, et al. 2020. Evaluating models' local decision boundaries via contrast sets. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, pages 1307–1323.

Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. 1999. Learning to forget: Continual prediction with lstm.

Edmund Gettier. 1963. Is justified true belief knowledge? Analysis, 23(6):121–123.

Codruta Girlea, Roxana Girju, and Eyal Amir. 2016. Psycholinguistic features for deceptive role detection in Werewolf. In Conference of the North American Chapter of the Association for Computational Linguistics.

Anna Gladkova, Aleksandr Drozd, and Satoshi Matsuoka. 2016. Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. In Proceedings of the NAACL Student Research Workshop, pages 8–15, San Diego, California. Association for Computational Linguistics.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking NLI systems with sentences that require simple lexical inferences. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 650–655, Melbourne, Australia. Association for Computational Linguistics.

Stephanie Gokhman, Jeff Hancock, Poornima Prabhu, Myle Ott, and Claire Cardie. 2012. In search of a gold standard in studies of deception. In Proceedings of the Workshop on Computational Approaches to Deception Detection.

Yoav Goldberg. 2017. Neural network methods for natural language processing. Synthesis Lectures on Human Language Technologies, 10(1):1–309.

Roberto González-Ibáñez, Smaranda Muresan, and Nina Wacholder. 2011. Identifying sarcasm in Twitter: A closer look. In Proceedings of the Association for Computational Linguistics.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. Advances in neural information processing systems, 27:2672–2680.

Abigail Green. 2003. Representing germany? the zollverein at the world exhibitions, 1851–1862. The Journal of Modern History, 75(4):836–863.

Stephan Greene and Philip Resnik. 2009. More than words: Syntactic packaging and implicit sentiment. In Conference of the North American Chapter of the Association for Computational Linguistics.

Liane Guillou and Christian Hardmeier. 2016. PROTEST: A test suite for evaluating pronouns in machine translation. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), pages 636–643, Portorož, Slovenia. European Language Resources Association (ELRA).

Liane Guillou and Christian Hardmeier. 2018. Automatic reference-based evaluation of pronoun translation misses the point. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 4797–4802, Brussels, Belgium. Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In NAACL-HLT.

Birgit Hamp and Helmut Feldweg. 1997. GermaNet - a lexical-semantic net for German. In Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications.

Viktor Hangya and Alexander Fraser. 2019. Unsupervised parallel sentence extraction with parallel segment detection helps machine translation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1224–1234, Florence, Italy. Association for Computational Linguistics.

Christian Hardmeier. 2012. Discourse in statistical machine translation. a survey and a case study. Discours. Revue de linguistique, psycholinguistique et informatique. A journal of linguistics, psycholinguistics and computational linguistics, (11).

Christian Hardmeier and Marcello Federico. 2010. Modelling pronominal anaphora in statistical machine translation. In IWSLT (International Workshop on Spoken Language Translation); Paris, France; December 2nd and 3rd, 2010., pages 283–289.

Naeemul Hassan, Fatma Arslan, Chengkai Li, and Mark Tremayne. 2017. Toward automated fact-checking: Detecting check-worthy factual claims by claimbuster. In Knowledge Discovery and Data Mining.

Charles T Hemphill, John J Godfrey, and George R Doddington. 1990. The atis spoken language systems pilot corpus. In Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990.

Matthew Henderson, Pawel Budzianowski, Iñigo Casanueva, Sam Coope, Daniela Gerz, Girish Kumar, Nikola Mrksic, Georgios Spithourakis, Pei-Hao Su, Ivan Vulic, and Tsung-Hsien Wen. 2019. A repository of conversational datasets. CoRR, abs/1904.06472.

Matthew Henderson, Blaise Thomson, and Jason D. Williams. 2014. The second dialog state tracking challenge. In Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL), pages 263–272, Philadelphia, PA, U.S.A. Association for Computational Linguistics.

David Hill. 2014. Got your back. This American Life Podcast.

Geoffrey E. Hinton and Ruslan Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. Science, 313:504 – 507.

Shuyuan Mary Ho, Jeffrey T Hancock, and Cheryl Booth. 2017. Ethical dilemma: Deception dynamics in computer-mediated group communication. Journal of the Association for Information Science and Technology.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. Neural computation, 9(8):1735–1780.

Na Hong, Andrew Wen, Majid Rastegar Mojarad, Sunghwan Sohn, Hongfang Liu, and Guoqian Jiang. 2018. Standardizing heterogeneous annotation corpora using hl7 fhir for facilitating their reuse and integration in clinical nlp. In AMIA Annual Symposium Proceedings, volume 2018, page 574. American Medical Informatics Association.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In Proceedings of the Association for Computational Linguistics.

W John Hutchins. 2004. The georgetown-ibm experiment demonstrated in january 1954. In Conference of the Association for Machine Translation in the Americas, pages 102–114. Springer.

Mohit Iyyer, Anupam Guha, Snigdha Chaturvedi, Jordan Boyd-Graber, and Hal Daumé III. 2016. Feuding families and former friends: Unsupervised learning for dynamic fictional relationships. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1534–1544.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In Proceedings of the Association for Computational Linguistics.

Tom N Jagatic, Nathaniel A Johnson, Markus Jakobsson, and Filippo Menczer. 2007. Social phishing. Communications of the ACM, 50(10):94–100.

Nathalie Japkowicz and Shaju Stephen. 2002. The class imbalance problem: A systematic study. Intelligent data analysis, 6(5):429–449.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In Proceedings of Empirical Methods in Natural Language Processing.

Qiqi Jiang, Chuan-Hoo Tan, Chee Wei Phang, Juliana Sutanto, and Kwok-Kee Wei. 2013. Understanding chinese online users and their visits to websites: Application of zipf's law. International journal of information management, 33(5):752–763.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with gpus. arXiv preprint arXiv:1702.08734.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke S. Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Proceedings of the Association for Computational Linguistics.

Daniel Jurafsky and James H Martin. 2000. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Prentice Hall PTR.

David Jurgens and Roberto Navigli. 2014. It's all fun and games until someone annotates: Video games with a purpose for linguistic annotation. In Transactions of the Association for Computational Linguistics.

Prathyusha Jwalapuram, Shafiq Joty, Irina Temnikova, and Preslav Nakov. 2019. Evaluating pronominal anaphora in machine translation: An evaluation measure and a test suite. 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 2964–2975, Hong Kong, China. Association for Computational Linguistics.

Kushal Kafle, Mohammed Yousefhussien, and Christopher Kanan. 2017. Data augmentation for visual question answering. In Proceedings of the 10th International Conference on Natural Language Generation, pages 198–202.

Mary E Kaplar and Anne K Gordon. 2004. The enigma of altruistic lying: Perspective differences in what motivates and justifies lie telling within romantic relationships. Personal Relationships, 11(4):489–507.

Jungo Kasai, Kun Qian, Sairam Gurajada, Yunyao Li, and Lucian Popa. 2019. Low-resource deep entity resolution with transfer and active learning. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5851–5861, Florence, Italy. Association for Computational Linguistics.

John F Kelley. 1984. An iterative design methodology for user-friendly natural language office information applications. ACM Transactions on Information Systems (TOIS), 2(1):26–41.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. Proceedings of Empirical Methods in Natural Language Processing.

Yunsu Kim, Yingbo Gao, and Hermann Ney. 2019. Effective cross-lingual transfer of neural machine translation models without shared vocabularies. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1246–1257, Florence, Italy. Association for Computational Linguistics.

Bryan Klimt and Yiming Yang. 2004. The enron corpus: A new dataset for email classification research. In European Conference on Machine Learning, pages 217–226. Springer.

Shailesh Kochhar, Stefano Mazzocchi, and Praveen Paritosh. 2010. The anatomy of a large-scale human computation engine. In Proceedings of the acm sigkdd workshop on human computation, pages 10–17.

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

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions, pages 177–180. Association for Computational Linguistics.

Philipp Koehn and Christof Monz. 2006. Manual and automatic evaluation of machine translation between european languages. In Proceedings on the Workshop on Statistical Machine Translation, pages 102–121.

Maximilian Köper, Sabine Schulte im Walde, Max Kisselew, and Sebastian Padó. 2016. Improving zero-shot-learning for german particle verbs by using training-space restrictions and local scaling. In Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics, pages 91–96.

Ana Kozomara and Sam Griffiths-Jones. 2014. mirbase: annotating high confidence micrornas using deep sequencing data. Nucleic acids research, 42(D1):D68–D73.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105.

Abhimanu Kumar and Matthew Lease. 2011. Learning to rank from a noisy crowd. In Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, pages 1221–1222.

Srijan Kumar, Justin Cheng, Jure Leskovec, and V.S. Subrahmanian. 2017. An Army of Me: Sockpuppets in Online Discussion Communities. In Proceedings of the World Wide Web Conference, Republic and Canton of Geneva, Switzerland.

Srijan Kumar and Neil Shah. 2018. False information on web and social media: A survey. In Social Media Analytics: Advances and Applications. CRC.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. Transactions of the Association for Computational Linguistics, 7:453–466.

Vivian Lai, Han Liu, and Chenhao Tan. 2020. "why is 'chicago' deceptive?" Towards building model-driven tutorials for humans. In International Conference on Human Factors in Computing Systems.

Chia-Hsuan Lee, Shang-Ming Wang, Huan-Cheng Chang, and Hung-Yi Lee. 2018. Odsqa: Open-domain spoken question answering dataset. In 2018 IEEE Spoken Language Technology Workshop (SLT).

Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2011. Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. In Proceedings of the 15th conference on computational natural language learning: Shared task, pages 28–34. Association for Computational Linguistics.

Honglak Lee, Peter Pham, Yan Largman, and Andrew Y Ng. 2009. Unsupervised feature learning for audio classification using convolutional deep belief networks. In Proceedings of Advances in Neural Information Processing Systems, pages 1096–1104.

Oliver Lemon, Kallirroi Georgila, James Henderson, and Matthew Stuttle. 2006. An isu dialogue system exhibiting reinforcement learning of dialogue policies: generic slot-filling in the talk in-car system. In Demonstrations.

Gondy Leroy and James E Endicott. 2012. Combining nlp with evidence-based methods to find text metrics related to perceived and actual text difficulty. In Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium, pages 749–754.

Anton Leuski, Ronakkumar Patel, David Traum, and Brandon Kennedy. 2009. Building effective question answering characters. In Proceedings of the Annual SIGDIAL Meeting on Discourse and Dialogue.

Timothy R. Levine, Hee Sun Park, and Steven A. McCornack. 1999. Accuracy in detecting truths and lies: Documenting the "veracity effect". Communication Monographs, 66(2):125–144.

Sarah Ita Levitan, Angel Maredia, and Julia Hirschberg. 2018. Linguistic cues to deception and perceived deception in interview dialogues. In Conference of the North American Chapter of the Association for Computational Linguistics.

Omer Levy and Yoav Goldberg. 2014. Linguistic regularities in sparse and explicit word representations. In Proceedings of the eighteenth conference on computational natural language learning, pages 171–180.

David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. 2004. Rcv1: A new benchmark collection for text categorization research. Journal of machine learning research, 5(Apr):361–397.

Patrick Lewis, Barlas Oğuz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2019. Mlqa: Evaluating cross-lingual extractive question answering. arXiv preprint arXiv:1910.07475.

Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. 2020. Question and answer test-train overlap in open-domain question answering datasets. arXiv preprint arXiv:2008.02637.

Tal Linzen. 2020. How can we accelerate progress towards human-like linguistic generalization? In Proceedings of the Association for Computational Linguistics.

Sharid Loáiciga, Liane Guillou, and Christian Hardmeier. 2017. What is it? disambiguating the different readings of the pronoun 'it'. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1325–1331, Copenhagen, Denmark. Association for Computational Linguistics.

Max Louwerse, David Lin, Amanda Drescher, and Gun Semin. 2010. Linguistic cues predict fraudulent events in a corporate social network. In Proceedings of the Annual Meeting of the Cognitive Science Society.

Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. arXiv preprint arXiv:1506.08909.

James Edwin Mahon. 2016. The definition of lying and deception. In The Stanford Encyclopedia of Philosophy, winter 2016 edition. Metaphysics Research Lab, Stanford University.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. Computational linguistics.

Sameen Maruf, André F. T. Martins, and Gholamreza Haffari. 2019. Selective Attention for Context-aware Neural Machine Translation. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 3092–3102, Minneapolis, Minnesota. Association for Computational Linguistics.

Winter Mason and Siddharth Suri. 2012. Conducting behavioral research on amazon's mechanical turk. Behavior research methods, 44(1):1–23.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval, pages 43–52.

R Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In Proceedings of the Association for Computational Linguistics.

Warren S McCulloch and Walter Pitts. 1943. A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 5(4):115–133.

Merriam-Webster. Crowdsourcing. In Merriam-Webster.com dictionary.

Paul Michel and Graham Neubig. 2018. Mtnt: A testbed for machine translation of noisy text. In Proceedings of Empirical Methods in Natural Language Processing.

Lesly Miculicich, Dhananjay Ram, Nikolaos Pappas, and James Henderson. 2018. Document-Level Neural Machine Translation with Hierarchical Attention Networks. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2947–2954. Association for Computational Linguistics.

Lesly Miculicich Werlen and Andrei Popescu-Belis. 2017. Validation of an automatic metric for the accuracy of pronoun translation (APT). In Proceedings of the Third Workshop on Discourse in Machine Translation, pages 17–25, Copenhagen, Denmark. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013a. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems, pages 3111–3119.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In Proceedings of Advances in Neural Information Processing Systems.

Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies, pages 746–751.

George A. Miller. 1995a. Wordnet: A lexical database for english. COMMUNICATIONS OF THE ACM, 38:39–41.

George A Miller. 1995b. Wordnet: a lexical database for english. Communications of the ACM, 38(11):39–41.

Taniya Mishra and Srinivas Bangalore. 2010. Qme!: A speech-based question-answering system on mobile devices. In Conference of the North American Chapter of the Association for Computational Linguistics.

Tom Mitchell. 1997. Introduction to machine learning. Machine Learning, 7:2–5.

Ethan Mollick and Ramana Nanda. 2016. Wisdom or madness? comparing crowds with expert evaluation in funding the arts. Manag. Sci., 62:1533–1553.

Barzan Mozafari, Purnamrita Sarkar, Michael J. Franklin, Michael I. Jordan, and Samuel Madden. 2014. Scaling up crowd-sourcing to very large datasets: A case for active learning. Proc. VLDB Endow., 8:125–136.

Mathias Müller, Annette Rios, Elena Voita, and Rico Sennrich. 2018. A Large-Scale Test Set for the Evaluation of Context-Aware Pronoun Translation in Neural Machine Translation. In WMT 2018, Brussels, Belgium. Association for Computational Linguistics.

Matthew L Newman, James W Pennebaker, Diane S Berry, and Jane M Richards. 2003. Lying words: Predicting deception from linguistic styles. Personality and social psychology bulletin, 29(5):665–675.

An T Nguyen, Matthew Lease, and Byron C Wallace. 2019. Explainable modeling of annotations in crowdsourcing. In Proceedings of the 24th International Conference on Intelligent User Interfaces, pages 575–579.

Vlad Niculae, Srijan Kumar, Jordan Boyd-Graber, and Cristian Danescu-Niculescu-Mizil. 2015. Linguistic harbingers of betrayal: A case study on an online strategy game. In Proceedings of the Association for Computational Linguistics.

Stefanie Nowak and Stefan Rüger. 2010. How reliable are annotations via crowdsourcing: a study about inter-annotator agreement for multi-label image annotation. In Proceedings of the international conference on Multimedia information retrieval, pages 557–566.

Daniela Oliveira, Harold Rocha, Huizi Yang, Donovan Ellis, Sandeep Dommaraju, Melis Muradoglu, Devon Weir, Adam Soliman, Tian Lin, and Natalie Ebner. 2017. Dissecting spear phishing emails for older vs young adults: On the interplay of

weapons of influence and life domains in predicting susceptibility to phishing. In International Conference on Human Factors in Computing Systems.

Myle Ott, Claire Cardie, and Jeff Hancock. 2012. Estimating the prevalence of deception in online review communities. In Proceedings of the World Wide Web Conference.

Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T Hancock. 2011. Finding deceptive opinion spam by any stretch of the imagination. In Proceedings of the Association for Computational Linguistics.

Bo Pang, Lillian Lee, et al. 2008. Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval, 2(1–2):1–135.

Philip Paquette, Yuchen Lu, Seton Steven Bocco, Max Smith, Satya Ortiz-Gagné, Jonathan K. Kummerfeld, Joelle Pineau, Satinder Singh, and Aaron C Courville. 2019. No-press diplomacy: Modeling multi-agent gameplay. In Proceedings of Advances in Neural Information Processing Systems. Curran Associates, Inc.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In Conference on Neural Information Processing Systems: Autodiff Workshop: The Future of Gradient-based Machine Learning Software and Techniques.

Vijayaditya Peddinti, Guoguo Chen, Vimal Manohar, Tom Ko, Daniel Povey, and Sanjeev Khudanpur. 2015. Jhu aspire system: Robust lvcsr with tdnns, ivector adaptation and rnn-lms. In Automatic Speech Recognition and Understanding (ASRU), IEEE Workshop on, pages 539–546.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In Proceedings of Empirical Methods in Natural Language Processing.

Verónica Pérez-Rosas, Mohamed Abouelenien, Rada Mihalcea, Yao Xiao, C. J. Linton, and Mihai Burzo. 2016. Verbal and nonverbal clues for real-life deception detection. In Proceedings of Empirical Methods in Natural Language Processing.

Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. 2017. Automatic detection of fake news. Proceedings of International Conference on Computational Linguistics.

Denis Peskov, Nancy Clarke, Jason Krone, Brigi Fodor, Yi Zhang, Adel Youssef, and Mona Diab. 2019. Multi-domain goal-oriented dialogues (multidogo): Strategies toward curating and annotating large scale dialogue data. 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 4518–4528.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In Proc. of NAACL.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Nagendra Goel, Mirko Hannemann, Yanmin Qian, Petr Schwarz, and Georg Stemmer. 2011. The Kaldi speech recognition toolkit. In IEEE Workshop on Automatic Speech Recognition and Understanding.

David MW Powers. 1998. Applications and explanations of zipf's law. In New methods in language processing and computational natural language learning.

Vinodkumar Prabhakaran, Ajita John, and Dorée D Seligmann. 2013. Power dynamics in spoken interactions: a case study on 2012 Republican primary debates. In Proceedings of the World Wide Web Conference.

Raimon H. R. Pruim, Maarten Mennes, Daan van Rooij, Alberto Llera, Jan K. Buitelaar, and Christian F. Beckmann. 2015. Ica-aroma: A robust ica-based strategy for removing motion artifacts from fmri data. NeuroImage, 112:267–277.

Junfei Qiu, Qihui Wu, Guoru Ding, Yuhua Xu, and Shuo Feng. 2016. A survey of machine learning for big data processing. EURASIP Journal on Advances in Signal Processing, 2016(1):67.

Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher D Manning. 2010. A multi-pass sieve for coreference resolution. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pages 492–501.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018a. Know what you don't know: Unanswerable questions for SQuAD. In Proceedings of the Association for Computational Linguistics.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018b. Know what you don't know: Unanswerable questions for SQuAD. In Proceedings of the Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In Proceedings of Empirical Methods in Natural Language Processing.

Juan Ramos. 2003. Using tf-idf to determine word relevance in document queries. In Proceedings of the International Conference of Machine Learning.

Joseph Reagle and Lauren Rhue. 2011. Gender bias in Wikipedia and Britannica. International Journal of Communication, 5(0).

Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. Coqa: A conversational question answering challenge. Transactions of the Association for Computational Linguistics, 7:249–266.

Philip Resnik, Mari Broman Olsen, and Mona Diab. 1999. The bible as a parallel corpus: Annotating the 'book of 2000 tongues'. Computers and the Humanities, 33(1-2):129–153.

Philip Resnik and Noah A Smith. 2003. The web as a parallel corpus. Computational Linguistics, 29(3):349–380.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why should i trust you?" explaining the predictions of any classifier. In Knowledge Discovery and Data Mining.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging NLP models. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 856–865, Melbourne, Australia. Association for Computational Linguistics.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. Foundations and Trends in Information Retrieval, 3(4):333–389.

Pedro Rodriguez, Shi Feng, Mohit Iyyer, He He, and Jordan Boyd-Graber. 2019. Quizbowl: The case for incremental question answering. arXiv preprint arXiv:1904.04792.

Anna Rogers, Aleksandr Drozd, and Bofang Li. 2017. The (too many) problems of analogical reasoning with word vectors. In Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (* SEM 2017), pages 135–148.

Ronald Rosenfeld. 2000. Two decades of statistical language modeling: Where do we go from here? Proceedings of the IEEE, 88(8):1270–1278.

Sebastian Ruder. 2016. An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747.

Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence, 1(5):206–215.

Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. 2018. Gender bias in coreference resolution. In Conference of the North American Chapter of the Association for Computational Linguistics.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning representations by back-propagating errors. nature, 323(6088):533–536.

Marta Sabou, Kalina Bontcheva, Leon Derczynski, and Arno Scharl. 2014. Corpus annotation through crowdsourcing: Towards best practice guidelines. In Proceedings of the Language Resources and Evaluation Conference.

Claude Sammut and Geoffrey I. Webb, editors. 2010. Mean Squared Error, pages 653–653. Springer US, Boston, MA.

Eric Schenk and Claude Guittard. 2011. Towards a characterization of crowdsourcing practices. Journal of Innovation Economics Management, (1):93–107.

Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims. 2015. Evaluation methods for unsupervised word embeddings. In Proceedings of the 2015 conference on empirical methods in natural language processing, pages 298–307.

Hagen Schulze. 1991. The Course of German Nationalism: From Frederick the Great to Bismarck 1763–1867. Cambridge University Press.

Sebastian Schuster, Sonal Gupta, Rushin Shah, and Mike Lewis. 2018. Cross-lingual transfer learning for multilingual task oriented dialog. arXiv preprint arXiv:1810.13327.

Rico Sennrich. 2017. How grammatical is character-level neural machine translation? assessing MT quality with contrastive translation pairs. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 376–382, Valencia, Spain. Association for Computational Linguistics.

Claude Elwood Shannon, Warren Weaver, et al. 1949. mathematical theory of communication.

Kathryn Sharpe Wessling, Joel Huber, and Oded Netzer. 2017. MTurk Character Misrepresentation: Assessment and Solutions. Journal of Consumer Research, 44(1):211–230.

Elben Shira and Matthew Lease. 2010. Expert search on code repositories.

Ben Shneiderman. 2000. Designing trust into online experiences. Communications of the ACM, 43(12):57–59.

Frederick A Siegler. 1966. Lying. American Philosophical Quarterly, 3(2):128–136.

Jason Smith, Herve Saint-Amand, Magdalena Plamadă, Philipp Koehn, Chris Callison-Burch, and Adam Lopez. 2013. Dirt cheap web-scale parallel text from the common crawl. In Proceedings of the Association for Computational Linguistics, pages 1374–1383.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y Ng. 2008. Cheap and fast—but is it good?: Evaluating non-expert annotations for natural language tasks. In Proceedings of Empirical Methods in Natural Language Processing.

Felix Soldner, Verónica Pérez-Rosas, and Rada Mihalcea. 2019. Box of lies: Multimodal deception detection in dialogues. In Conference of the North American Chapter of the Association for Computational Linguistics.

Matthias Sperber, Graham Neubig, Jan Niehues, and Alex Waibel. 2017. Neural lattice-to-sequence models for uncertain inputs. In Proceedings of the Association for Computational Linguistics.

Gabriel Stanovsky, Noah A. Smith, and Luke Zettlemoyer. 2019. Evaluating gender bias in machine translation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1679–1684, Florence, Italy. Association for Computational Linguistics.

Dario Stojanovski and Alexander Fraser. 2018. Coreference and coherence in neural machine translation: A study using oracle experiments. In Proceedings of the Third Conference on Machine Translation: Research Papers, pages 49–60, Brussels, Belgium. Association for Computational Linguistics.

Siddharth Suri, Daniel G. Goldstein, and Winter A. Mason. 2011. Honesty in an online labor market. In Proceedings of the 11th AAAI Conference on Human Computation, AAAIWS'11-11, page 61–66. AAAI Press.

Lyn M. Van Swol, Deepak Malhotra, and Michael T. Braun. 2012. Deception and its detection: Effects of monetary incentives and personal relationship history. Communication Research, 39(2):217–238.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018a. Fever: a large-scale dataset for fact extraction and verification. arXiv preprint arXiv:1803.05355.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal, editors. 2018b. Proceedings of the First Workshop on Fact Extraction and VERification (FEVER). Association for Computational Linguistics.

Robert Tibshirani. 1996. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1):267–288.

Jörg Tiedemann and Yves Scherrer. 2017. Neural Machine Translation with Extended Context. In Proceedings of the Third Workshop on Discourse in Machine Translation, pages 82–92.

Catalina L Toma and Jeffrey T Hancock. 2012. What lies beneath: The linguistic traces of deception in online dating profiles. Journal of Communication, 62(1):78–97.

Isabelle Torrance. 2015. Distorted oaths in Aeschylus. Illinois Classical Studies, 40(2):281–295.

AM Turing. 1950. Computing machinery and intelligence.

Peter D Turney. 2008. A uniform approach to analogies, synonyms, antonyms, and associations. arXiv preprint arXiv:0809.0124.

Shyam Upadhyay, Manaal Faruqui, Gokhan Tür, Hakkani-Tür Dilek, and Larry Heck. 2018. (almost) zero-shot cross-lingual spoken language understanding. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 6034–6038. IEEE.

Donna Vakharia and Matthew Lease. Beyond mechanical turk: An analysis of paid crowd work platforms.

Vladimir Vapnik. 1995. The nature of statistical learning theory. Springer science & business media.

Dániel Varga, András Kornai, Viktor Nagy, László Németh, and Viktor Trón. 2007. Parallel corpora for medium density languages.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Proceedings of Advances in Neural Information Processing Systems.

Tony Veale. 2016. Round up the usual suspects: Knowledge-based metaphor generation. In Proceedings of the Fourth Workshop on Metaphor in NLP, pages 34–41.

Jean-Paul Vinay and Jean Darbelnet. 1995. Comparative stylistics of French and English: A methodology for translation, volume 11. John Benjamins Publishing.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019. When a Good Translation is Wrong in Context: Context-Aware Machine Translation Improves on Deixis, Ellipsis, and Lexical Cohesion. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1198–1212, Florence, Italy. Association for Computational Linguistics.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. Context-Aware Neural Machine Translation Learns Anaphora Resolution. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volu pages 1264–1274, Melbourne, Australia.

Ellen M Voorhees et al. 1999. The trec-8 question answering track report. In Trec, volume 99, pages 77–82.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. Communications of the ACM, 57(10):78–85.

Maja Vukovic and Claudio Bartolini. 2010. Towards a research agenda for enterprise crowdsourcing. In International Symposium On Leveraging Applications of Formal Methods, Verification and Validation, pages 425–434. Springer.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019a. Universal adversarial triggers for attacking and analyzing nlp. arXiv preprint arXiv:1908.07125.

Eric Wallace, Pedro Rodriguez, Shi Feng, Ikuya Yamada, and Jordan Boyd-Graber. 2019b. 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.

Eric Wallace, Pedro Rodriguez, Shi Feng, Ikuya Yamada, and Jordan Boyd-Graber. 2019c. Trick me if you can: Human-in-the-loop generation of adversarial question answering examples. Transactions of the Association of Computational Linguistics, 10.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems. In Advances in neural information processing systems, pages 3266–3280.

Xiaosen Wang, Hao Jin, and Kun He. 2019b. Natural language adversarial attacks and defenses in word level. arXiv preprint arXiv:1909.06723.

Zeerak Waseem. 2016. Are you a racist or am i seeing things? annotator influence on hate speech detection on twitter. In NLP+CSS@EMNLP.

Bonnie Webber. 1992. Question answering. In Stuart C. Shapiro, editor, Encyclopedia of Artificial Intelligence, 2nd edition, pages 814–822. John Wiley & Sons, Inc., New York, NY.

Wei Wei, Quoc Le, Andrew Dai, and Jia Li. 2018. Airdialogue: An environment for goal-oriented dialogue research. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3844–3854.

Frank Wessel and Hermann Ney. 2004. Unsupervised training of acoustic models for large vocabulary continuous speech recognition. IEEE Transactions on Speech and Audio Processing.

Mark E Whiting, Grant Hugh, and Michael S Bernstein. 2019. Fair work: Crowd work minimum wage with one line of code. In Proceedings of the AAAI Conference on Human Computation and Crowdsourcing, volume 7, pages 197–206.

Jake Ryland Williams, Paul R Lessard, Suma Desu, Eric M Clark, James P Bagrow, Christopher M Danforth, and Peter Sheridan Dodds. 2015. Zipf's law holds for phrases, not words. Scientific reports, 5:12209.

Jason Williams, Antoine Raux, and Matthew Henderson. 2016. The dialog state tracking challenge series: A review. Dialogue & Discourse, 7(3):4–33.

Ludwig Wittgenstein. 1953. Philosophical Investigations.

Marty J Wolf, Keith W Miller, and Frances S Grodzinsky. 2017. Why we should have seen that coming: comments on microsoft's tay "experiment," and wider implications. The ORBIT Journal, 1(2):1–12.

Stephen M Wolfson and Matthew Lease. 2011. Look before you leap: Legal pitfalls of crowdsourcing. Proceedings of the American Society for Information Science and Technology, 48(1):1–10.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144.

Bo Xie and Julie M. Bugg. 2009. Public library computer training for older adults to access high-quality internet health information. Library and Information Science Research, 31(3).

Ikuya Yamada, Ryuji Tamaki, Hiroyuki Shindo, and Yoshiyasu Takefuji. 2018. Studio Ousia's quiz bowl question answering system. In NIPS Competition: Building Intelligent Systems, pages 181–194.

Kaiyu Yang, Klint Qinami, Li Fei-Fei, Jia Deng, and Olga Russakovsky. 2020. Towards fairer datasets: Filtering and balancing the distribution of the people subtree in the imagenet hierarchy. In Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, pages 547–558.

Dani Yogatama, Cyprien de Masson d'Autume, Jerome Connor, Tomás Kociský, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, and Phil Blunsom. 2019. Learning and evaluating general linguistic intelligence. arXiv preprint arXiv:1901.11373.

Omar Zaidan and Chris Callison-Burch. 2011. Crowdsourcing translation: Professional quality from non-professionals. In Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies, pages 1220–1229.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.

Zhi-Hua Zhou and Xu-Ying Liu. 2005. Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Transactions on knowledge and data engineering, 18(1):63–77.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pages 1097–1100.

George Kingsley Zipf. 1935. The psycho-biology of language: An introduction to dynamic philology, volume 21. Psychology Press.