Chapter 1: See previous updates

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 **crowd-sourcing** and automated methods can be used to generate data at scale

(Section 2.3.3). Data can be gathered and annotated exclusively using **experts** (Section 2.3.4). Last, **hybrid** approaches combine anonymous crowd users with experts that verify the results (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 (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 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.

A statistical approach to language—a departure from the linguistics paradigm—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 with the insights of J.R. Firth, who declared that, "you shall know a word by the company

[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!

it keeps" (Firth, 1957). This insight serves as the foundation of embedding-based representations of language in modern-day NLP.

The language model has created the dependence on training data, with which this proposal is concerned. Statistical language modeling evolved over the 20th century from the Markov chain (Markov, 1906; Shannon, 1948; 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 is a limit to how much a language model can learn from statistics without understanding language, but that limit has not yet been ascertained.

2.2 Tasks

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

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

2.2.1 Machine Translation

Machine translation needs text from multiple languages, which requires 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:

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

where e is the English sentence and f is the French sentence. Hence, $p(e \mid f)$ calculates the highest corresponding English sentence for the French one. This has the same intuition as Equation 2.1, since an existing word predicts an unseen 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 existing corpus that 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 Translation facilitates model-building for machine translation (Koehn and Monz, 2006), which would be impossible without standardized datasets for the community collaboration. Statistical Machine Translation has been supplanted by neural machine translation (NMT) (Wu et al., 2016). Chapter **??** evaluates limitations of NMT for coreference resolution (Soon et al., 2001), the task of disambiguating the appropriate pronoun for each named entity. Our research introduces a new machine translation task, cultural adaptation (Chapter **??**), that requires collecting translations from cultural experts for gold standard evaluation.

Machine translation can be used for downstream tasks, such as question answering. At a linguistic level, pronouns must be resolved in multiple languages (Müller et al., 2018) to answer a question. But entire questions are desirable for machine learning, and choosing how to translate entire sentences is nontrivial. MLQA and

[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
XQuAD	1,190	Expert
MLAQ	12,000	Hybrid

Table 2.1: A tabular summary of machine translation datasets.

XQuAD automatically generate paired questions through machine translation (Lewis et al., 2019; Artetxe et al., 2019). As an alternative, TyDi (Clark et al., 2020a) gives crowd-sourced users prompts from Wikipedia articles to create questions in a wide range of languages. The following section discusses question answering, independently of machine translation.

2.2.2 Question Answering

Question answering (QA) is another task heavily dependent on training data. 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.

Questions
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.

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. Unlike Table 2.2, these questions are done through crowd-sourcing and Wikipedia and are not carefully planned.

The Text Retrieval Conference established QA 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 QA datasets, with SQuAD (Rajpurkar et al., 2016, 2018) 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., 2018), MS Marco (Bajaj et al., 2016), TriviaQA (Joshi

et al., 2017) QuAC (Choi et al., 2018), Quizbowl (Rodriguez 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.

Computers can read a question and select the answer from a passage of text. This format of QA is called machine reading comprehension (Rajpurkar et al., 2016, MRC), and has been a popular choice for dataset design. 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).

Recent work acknowledges that certain community practices, such as crowdsourcing for questions, may not be optimal for QA. Wallace et al. (2019) work with 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 speak-

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 dialog datasets. The datasets described as hybrid all scrape or use naturally-occurring language and then supplement it with crowd-sourced annotation.

ers of typologically diverse languages. Boyd-Graber (2020) questions the paradigm of using crowd-sourced workers as the measure for human baselines, rather than evaluating through a play test. Rodriguez et al. (2021) questions the paradigm of using quantitative leaderboards, given the disparity of question difficulties. van der Goot (2021) questions the paradigm of using a development set for model tuning. Kummerfeld (2021) questions the qualification requirements for Mechanical Turk workers. Last, Karpinska et al. (2021) questions the output of Mechanical Turk workers for evaluation.

2.2.3 Dialogs

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).

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 dialog datasets.

2.3 Data Collection Type

Data for machine learning can come from one of four sources: automation, crowd-sourcing, experts, and a hybrid mix of the crowd with 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 information 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 parliamentary proceedings (Koehn, 2005). Literature comes from a verified author (Iyyer et al., 2016), as does journalism (Lewis et al., 2004). The United Nations maintains detailed datasets about global populations. 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

Data **generation** is necessary as the data necessary for NLP cannot always be found. 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 **??** describes our project in which text-to-speech creates a dataset of 500,000 audio files. While large, our dataset is limited to a single female voice and read in a notably different cadence than 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 ser-

vices, 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) build 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. The average psychology experiment is conducted using university students that require hourly compensation and usually come from a concentrated geographic area and socio-economic background. 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

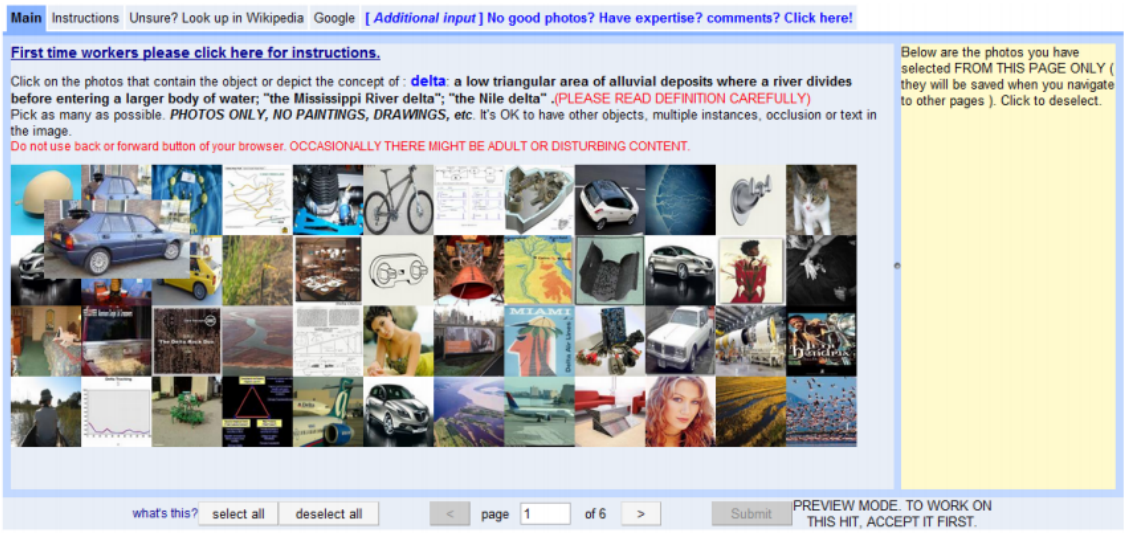

Figure 2.1: Deng et al. (2009) pioneers Mechanical Turk use for Computer Science. Simple **annotation** tasks can be completed reliably with crowd-sourcing since selecting if an image belongs to a WordNet category (e.g., car, bicycle, delta) is a relatively objective and straightforward task. However,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

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. This level of oversight is unlikely to occur 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 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). Snow et al. (2008) posit that (on average) four non-expert workers can emulate an expert for five NLP tasks: affect recognition, word similarity, textual entailment, temporal event recognition, and word sense disambiguation. 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., 2017).

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 for crowd-sourcing NLP data. 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. For classification tasks, 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. As a tangential consideration, legal regulation may ulti-

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

mately limit the effectiveness of this technique, since it is completely unregulated by current employment practices (Wolfson and Lease, 2011).

Chapter **??** 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. Experts are accountable in ways the crowd-user is not and do not require the same level of post-collection quality control.

2.3.4 Expert

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

> *"a person with a high level of knowledge or skill relating to a particular subject or activity."* The Cambridge Dictionary

Defining expertise is a tricky and subjective goal; for example, "high level" is highly subjective in this definition. Bourne et al. (2014) conclude that psychology is the appropriate framework for evaluating expertise, which "results from practice and experience, built on a foundation of talent, or innate ability". For NLP, we require that the person has both the incentive and skill to *accurately*, as opposed to quickly, complete their task. A degree of accountability, rather than full anonymity, is important as it prevents intentional fraud (Teitcher et al., 2015). Therefore, we require that experts be identifiable, in at least some capacity during the data

collection process. Such 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. Alternatively, skilled citizen scientists may generate high-quality language in the pursuit of a hobby such as journalism, writing, or debate. 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 Trade Organization are all organizations that release reliable large-scale data, as discussed in Section 2.3.1. These organizations hire professionals such as translators and lawyers to generate language.

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, 1995).

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 **??** 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 2.6.

2.3.5 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, slot filling. 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., 2018) 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).

2.4 Models & Metrics

Data does not exist in a vacuum and tasks cannot be solved without a formalization. Therefore, we summarize popular models used with the data to solve machine translation, question answering, and dialog. Additionally, we discuss the metrics used to evaluate these models. This emphasis on model, and not data, evaluation is a key limitation in NLP.

2.4.1 Logistic Regression

According to Ng and Jordan (2002), 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, $\vec{w}$, and a bias term, b according to:

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

The variable 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 $\vec{w}$ and b. Cross-entropy loss calculates the difference between the predicted $\hat{y}$ and the true y. The gradient descent algorithm (Bottou, 2010; Ruder, 2016) finds the minimum loss.

At test time, for each example the highest probability label is predicted in y. 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. A feature weight of close to zero would indicate that the feature is not essential for the model; conversely the highest weighted feature is important in the task. 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 a more powerful classifier than logistic regressions and can be shown to learn any function due to a hidden layer. 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.

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 and Pitts, 1943; Rosenblatt, 1958). . However, it was not until the 21st century that computing infrastructure allowed neural networks to be effectively applied.

All neural networks depend on a **loss function** and **backpropagation** The **loss function** tells the neural network how quantitatively wrong a prediction is. Popular loss functions include Cross Entropy Loss—often used for logistic regres-

sion and classification tasks— and Mean Squared Error (Sammut and Webb, 2010).

Backpropagation 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., 2013) 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.

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 architectures more applicable to NLP: Deep Averaging Networks (Section 2.4.3) and Recurrent Neural Networks (Section 2.4.4).

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. Unlike Logistic Regression, the DAN adapts to linguistic versatility by using embeddings in lieu of specific word features. 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 **??**.

2.4.4 Sequence Models

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 **??** and **??**, as the current state of the apart for NLP.

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 **??**, 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.

2.4.5 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., 2019); 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**. Machine translation uses metrics such as BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), and TERp (Snover et al., 2009) as an automatic approximation of *target* quality; however, the quality of the *source* data—which must be generated by human users—is never evaluated. 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. We explore two examples of large-scale data projects and the limitations of relying on model accuracy, without data verification, in Chapter **??**.

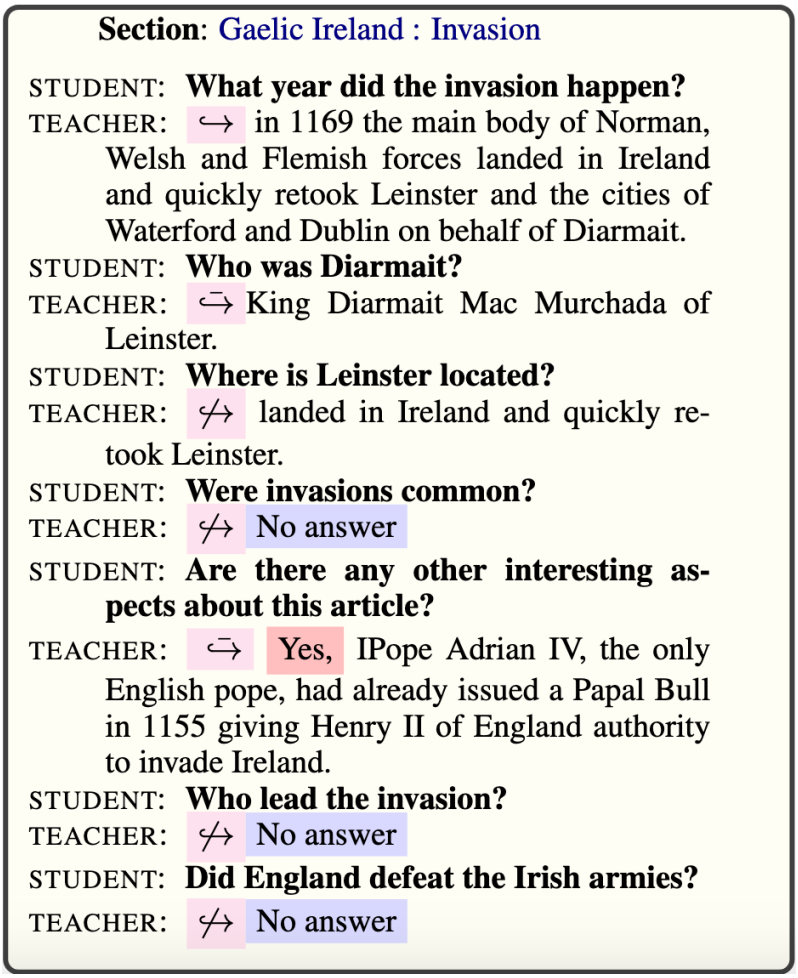

Figure 2.2: Crowd-sourcing can also be used to generate large-scale NLP data. However, **generation** creates a quality issue not present in **annotation**. In this particular example, Choi et al. (2018) highlight that the teacher does not provide quality responses. However, the student's conversation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

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 **??**.

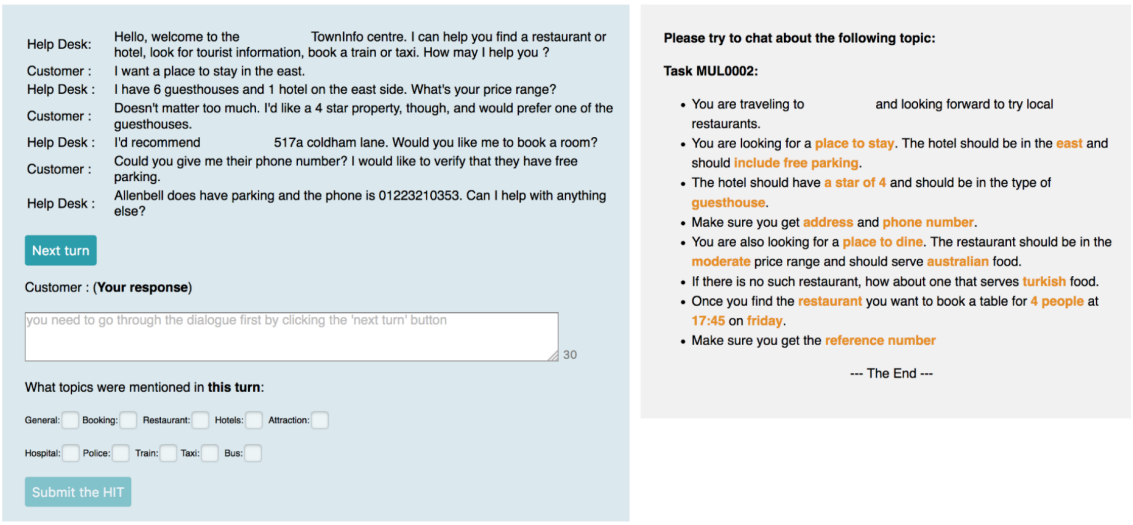

Figure 2.3: Hybrid approaches try to control the quality of language **generated** by the crowd. MultiWoz (Budzianowski et al., 2018), creates a rigid template for the user conversation, avoiding the worst quality issues at the expense of user creativity.

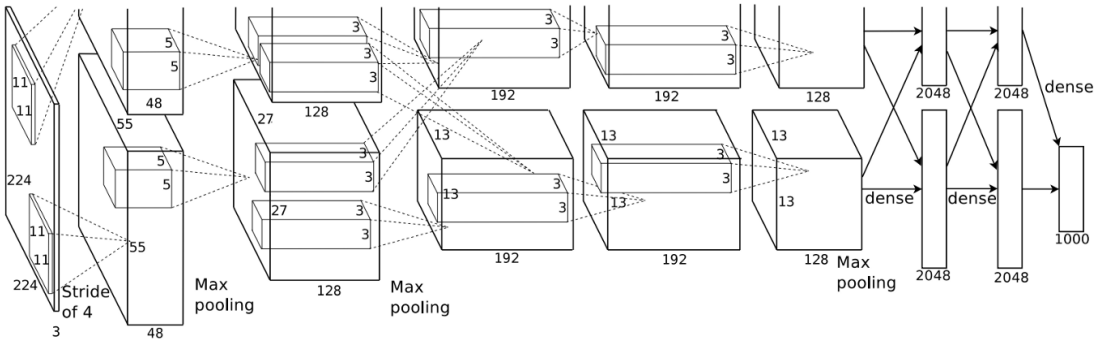

Figure 2.4: Krizhevsky et al. (2012)'s CNN architecture.

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).

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

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

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*.

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*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

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.

Léon Bottou. 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer.

L. E. Bourne, J. Kole, and A. Healy. 2014. Expertise: defined, described, explained. *Frontiers in Psychology*, 5.

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

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. In *Proceedings of Advances in Neural Information Processing Systems*.

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. In *Proceedings of Empirical Methods in Natural Language Processing*.

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.

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. In *Proceedings of Empirical Methods in Natural Language Processing*.

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.

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.

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*.

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*.

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*.

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.

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*.

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.

Matthew Dunn, Levent Sagun, Mike Higgins, V. Ugur Güney, Volkan Cirik, and Kyunghyun Cho. 2017. 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.

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 *Conference of the North American Chapter of the Association for Computational Linguistics*.

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*.

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

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

Rob van der Goot. 2021. We need to talk about train-dev-test splits. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 4485–4494, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

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.

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.

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.

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.

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.

Marzena Karpinska, Nader Akoury, and Mohit Iyyer. 2021. The perils of using mechanical turk to evaluate open-ended text generation. In *Proceedings of Empirical Methods in Natural Language Processing*.

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 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.

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.

Jonathan K. Kummerfeld. 2021. Quantifying and avoiding unfair qualification labour in crowdsourcing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 343–349, Online. Association for Computational Linguistics.

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.

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.

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*.

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

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*.

Andrey Andreyevich Markov. 1906. Extension of the law of large numbers to dependent quantities. *Izv. Fiz.-Matem. Obsch. Kazan Univ.(2nd Ser)*, 15:135–156.

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.

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*.

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

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

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. *arXiv preprint arXiv:1810.02268*.

Andrew Y Ng and Michael I Jordan. 2002. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in neural information processing systems*, pages 841–848.

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.

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.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the Association for Computational Linguistics*, pages 311–318.

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

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*.

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

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. 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*.

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.

Pedro Rodriguez, Joe Barrow, Alexander Miserlis Hoyle, John P. Lalor, Robin Jia, and Jordan Boyd-Graber. 2021. Evaluation examples are not equally informative: How should that change NLP leaderboards? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4486–4503, Online. Association for Computational Linguistics.

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*.

Frank Rosenblatt. 1958. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

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*.

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

Claude Elwood Shannon. 1948. A mathematical theory of communication. *Bell system technical journal*.

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.

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.

Matthew G Snover, Nitin Madnani, Bonnie Dorr, and Richard Schwartz. 2009. Terplus: paraphrase, semantic, and alignment enhancements to translation edit rate. *Machine Translation*, 23(2):117–127.

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*.

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational linguistics*, 27(4):521–544.

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.

Jennifer EF Teitcher, Walter O Bockting, José A Bauermeister, Chris J Hoefer, Michael H Miner, and Robert L Klitzman. 2015. Detecting, preventing, and responding to "fraudsters" in internet research: ethics and tradeoffs. *Journal of Law, Medicine & Ethics*, 43(1):116–133.

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

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

AM Turing. 1950. Computing machinery and intelligence.

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

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*.

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.

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

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Proceedings of Advances in Neural Information Processing Systems*.

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

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.

Ludwig Wittgenstein. 1953. *Philosophical Investigations*.

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*.

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.

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