Perceptual Models of Machine-Edited Text

Elizabeth M. Merkhofer,[1] Monica-Ann Mendoza,[1]
Rebecca Marvin[2] and **John C. Henderson[1]**
[1] MITRE
7525 Colshire Dr, McLean, VA 22102
[2] Independent
{emerkhofer,mamendoza,jhndrsn}@mitre.org,rebecca.c.marvin@gmail.com

Abstract

We introduce a novel dataset of human judgments of machine-edited text and initial models of those perceptions. Six machine-editing methods ranging from character swapping to variational autoencoders are applied to collections of English-language social media text and scientific abstracts. The edits are judged in context for detectability and the extent to which they preserve the meaning of the original. Automated measures of semantic similarity and fluency are evaluated individually and combined to produce composite models of human perception. Both meaning preservation and detectability are predicted within 6% of the upper bound of human consensus labeling.

1 Introduction

Machine-editing systems produce new versions of text using text as input. They contribute to tasks such as automatic summarization, simplification, natural language generation and generative adversarial NLP systems. These tasks have communicative goals, for example shorter, more accessible, or more appropriate text, and system developers are encouraged to improve their correlation with human performance on these tasks. While the measured task performance of machine-editing systems continues to improve, one might consider how humans perceive machine-edited text compared to human-produced text. One-off human evaluation of editing systems is expensive, incomparable, and must be constantly repeated. In this work we make first attempts at direct, general-purpose modeling of human perception of these texts and develop a model of human perception as it relates to two goals: to maximally maintain the meaning of an original and be minimally perceptible as machine output.

We present a dataset of human judgments about detectability and meaning preservation for machine-edited text.[1] This dataset consists of 14,400 judgments about contextualized pairs of machine-edited sentences. The original texts are English-language and come from two domains: scientific papers and social media. The edits are created by six different algorithms using a variety of techniques. By comparing trivial editors to more subtle approaches under the same evaluation framework, we move toward generic models of perception of edited text.

Our analysis finds high interannotator agreement and examines human preference among the six machine editors that generated the candidates. Existing measures of similarity and fluency are evaluated as models of perception. We find that reference-informed models come close to human consensus of meaning preservation and detectability. However, language models that don't have access to the reference text have less success as generic models of detection. This dataset and analysis constitute a step toward modeling meaning preservation and detectability under a variety of machine-editing conditions representative of the state of the practice.

2 Background

Machine editing is a component of multiple tasks that balance meaning preservation and fluency differently.

2.1 Machine-Editing Tasks

Text simplification (Saggion, 2017) and summarization (Narayan et al., 2018) produce new versions of text that are simpler or shorter, intended to be useful to a human reader. Evaluation measures informativeness relative to a reference. Abstractive techniques that fully rewrite the text have recently

[1] https://github.com/mitre/hpmet

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 3909–3920
August 1–6, 2021.

become viable alternatives to extractive techniques that build new texts from portions of the original text.

Paraphrase generation is the task of producing semantically equivalent variants of a sentence and can underlie applications like question answering and data augmentation. Recent approaches include a component that generates alternatives and a component that estimates their quality as paraphrase (Li et al., 2018; Kumar et al., 2020).

Natural language watermarking of text (Venugopal et al., 2011) and text steganography (Wilson et al., 2014) are conditional text generation practices that require both a meaningful surface form and the hidden encoding of additional information. In this case, it's essential that the text appear plausible as readers should not suspect the encoded information (Wilson et al., 2015).

Edited texts are used in adversarial learning and attacks for text processing systems. Adversarial inputs change a system output without altering some relevant aspect of human perception of the text, e.g. sentiment when attacking a sentiment analysis system (Alzantot et al., 2018). In cases of adversarial learning, where edited texts are used only to promote system robustness, human perception is not a concern (Jia and Liang, 2017). In contrast, adversarial attack vectors rely on human perception of the attack, whether it be communicating meaning regardless of detectability (Eger et al., 2019) or guaranteeing fluency (Zhang et al., 2019a). While authors have quantified the effect of adversarial perturbations on metrics of text quality like word modification rate and count of grammatical errors (Zeng et al., 2020), the relation of these automatic metrics to human perception is not yet studied.

2.2 Meaning Preservation

Machine editing often aims to guarantee semantic similarity or *meaning preservation* between input and output. Meaning preservation can be insensitive to surface forms such as tokenization, case-folding, stylistic variation in punctuation, spacing, font choice, and tense. Compact text representations (e.g. Morse code) tend to regularize all potential surface forms.

Semantic textual similarity and paraphrase identification are active areas of investigation in the NLP community (Cer et al., 2017). Natural language inference (NLI) also relies on notions of semantic similarity to recognize a larger set of relations between texts (Bowman et al., 2015). These subfields of NLP investigate semantic relatedness between *human-authored* texts.

Meaning preservation is related to the concept of *informativeness* used in automatic summarization and *adequacy* for machine translation. Summarization metrics tend to lean toward recall to make sure the central concepts of reference summaries are produced and MT metrics tend to lean toward precision to penalize systems that generate something outside of the references.

Many adversarial text editors don't require strict paraphrase, but simply that their perturbations not change the input's classification to a human reader (Ren et al., 2019; Lei et al., 2019; Alzantot et al., 2018; Ebrahimi et al., 2018). Other authors ask judges about similarity to the unperturbed original (Zhao et al., 2018; Alzantot et al., 2018; Ribeiro et al., 2018; Jin et al., 2020). New work correlates automatic metrics with human judgments capturing both semantic similarity and fluency about three word- and character-swapping algorithms (Michel et al., 2019).

2.3 Detectability

Language models were introduced early in both automatic speech recognition (Bahl et al., 1983) and statistical machine translation (Brown et al., 1990) to make output text more readable. They aimed to avoid decoding results that appeared computer-generated.

Recent work in several natural language generation tasks augments automatic evaluation, which approximates informativeness, with one-off human evaluations that estimate text quality. Authors elicit judgment for abstractive summaries about readability (Paulus et al., 2018), fluency (Hardy and Vlachos, 2018), and preference between human and machine-written abstracts (Fan et al., 2018). Desai et al. (2020) elicit human judgements of grammaticality for a compressive summarization system that deletes plausible spans. In image captioning and dialogue systems, several learned metrics judge system output to be higher quality when it is less distinguishable from human text (Cui et al., 2018; Lowe et al., 2017)

Several methods of generating adversarial text have been evaluated through surveys of human perception, for example by asking humans to detect the location of machine edits (Liang et al., 2018), or to judge the likelihood that a sentence is mod-

	arxiv	reddit
item count	7200	7200
total sentence lengths	177 078	140 453
mean length	24.7	19.5
mean context length	96.8	265.9

Table 1: Dataset size in words.

ified by a machine (Ren et al., 2019) or written by a human (Lei et al., 2019). Other authors ask human annotators about proxies like grammaticality (Jin et al., 2020), fluency (Zhang et al., 2019b) or readability (Hsieh et al., 2019) as a proxy for detectability.

Far more work asks whether computers can detect machine-edited text. Research on text generated with large language models finds that the output is easy to detect automatically because of the probabilities of the particular language model itself (Adelani et al., 2020; Gehrmann et al., 2019; Zellers et al., 2019). In fact, the generation setting that best fools humans produces output that is easy to detect automatically (Ippolito et al., 2020). This suggests human perception of such edits is different from machine detection.

Detectability and meaning preservation are not independent variables, but they represent different aspects of human perception. Destroying the fluency of a text can make it detectable as an edit in a high quality research document, but rewriting a section of chat in standard English can make it detectable in context. One can often transpose digits in scientific measurements to indetectably destroy meaning, and one could rewrite an abstract in randomized case patterns to raise suspicion without altering meaning.

3 Methods

3.1 Dataset Construction

We present a dataset of human judgments about two tasks, meaning preservation and detection, in each of two domains, social media and science writing. For each task and domain, we distributed packets of 600 multiple-choice questions to six judges. Each question was an AB test for a pair of editing systems both operating on a sentence in context. The first 105 questions of each packet were the same for all judges and are used to measure interannotator agreement. The remaining 495 sentences were the same, but the pairs of systems compared by judges varied. The judges were all native English speakers who work in AI research and were unfamiliar with

Which better preserves the meaning of the reference?

Reference: Later on that day I emailed the company that I purchased my order from and they confirmed it was delivered to that address.
A. Later that day I emailed the company I bought my order, and they confirmed that was delivered to that address.
B. Later in that time i received the website and i sent my email from what it said it was delivered for customer address.

Location: Florida I didn't know what to flair. About a month ago a package I ordered was delivered to my old apartment complex. When I went to the front office to ask if a package with my name was turned in they said no such thing had occurred. _____ I don't know how to move forward from this.

Which sentence reads more like it was altered by a machine?

A. When thi applied voltage is ifcreased to a few mV we find a strong declease of the spin injection efficiency.
B. While the required voltage is required to a tunnel voltage to obtain a lower amount of the joule injection injection.

Semiconductor spintronics will need to control spin injection phenomena in the non-linear regime. In order to study these effects we have performed spin injection measurements from a dilute magnetic semiconductor [(Zn,Be,Mn)Se] into nonmagnetic (Zn,Be)Se at elevated bias. _____ The observed behavior is modelled by extending the charge-imbalance model for spin injection to include band bending and charge accumulation at the interface of the two compounds. We find that the observed effects can be attributed to repopulation of the minority spin level in the magnetic semiconductor.

Figure 1: Sample prompts from the meaning preservation task on Reddit (top) and the detection task on ArXiv (bottom.)

the processes used to edit the original text.

Source sentences for the ArXiv dataset were randomly selected from all sentences in ArXiv abstracts submitted between its start in 1991 and the end of January, 2018. The Reddit sentences were randomly selected from all sentences in Reddit posts made in January, 2018. The two source collections were roughly the same size. Sentences less than 10 tokens or longer than 40 tokens were avoided in both collections to ensure judge productivity. To satisfy IRB and to minimize the likelihood of negative effects on judges, we excluded all posts from the subreddits listed on the official *nsfw* list, and any that were no longer reachable by September 2019. Table 1 describes statistics about the sentences selected for editing and the contexts provided for judges.

The meaning preservation task involved AB judgments on six different editing systems. For the detection task, we included the original texts among the edited variants for a total of seven sys-

tems. We refer to this as the `null` editor. An all-pairs design of six systems requires 15 pairs and an all-pairs design of seven systems requires 21 pairs. Both designs were iterated to yield 600 pairs of system variants, truncating the final seven system pattern early. The first 105 example editing pairs (seven full all-pairs sets for meaning and five for detection) were identical for all judges and the remaining 495 in each packet were chosen from the possible pairs according to *independent permutations* to encourage balance.

This can be described again for more clarity. $C(6, 2) = 15$ (meaning preservation) and $C(7, 2) = 21$ (detectability). 600 examples lines up perfectly with a 15-item boundary but not with a 210 item boundary. Thus, there are 12 examples left over from a complete set of all-pairs of 7 systems in detectability when truncating to 600 items per source per judge. The first 105 system pair assignments come from 7 cycles through $C(6, 2)$ pairs or 5 cycles through $C(7, 2)$. The remaining sequences of pairs for each judge are all-pairs cycles through independently randomized permutations of the systems. Machine edit assignment to positions A and B were independently shuffled for each judge and the questions were presented to each judge in randomly shuffled order. Judges were instructed to choose between the two alternatives.

Each item was presented as a choice between two edited versions of the same sentence, presented with the rest of the Reddit post or ArXiv abstract as context. Figure 1 shows examples and the specific prompts used to elicit judgments. In the meaning preservation example, the candidates were produced by round-trip machine translation and the VAE. In the detection example, the candidates were produced by charswap and the VAE. Cases where detection paired the `null` system against a machine editor were collected to determine how often each editor was preferred to the original.

Less than half of one percent of detectability items are automatically marked as ties. These include cases where the edited text is the same string as the original, disregarding casing and punctuation, or where the VIPER editor (described below) produced an alternative that rendered identically in packets. These are included in the analysis to capture the intuition that a perceptual model should score ties the same.

3.2 Machine-Editing Systems

We employ six editing systems to capture the effect that varied systems have on human perception. Each takes just the sentence to be edited, without context.

3.2.1 Swapping editors

Simple word- and character-swapping editors are prevalent in literature about adversarial attacks and data augmentation (Michel et al., 2019). Our **charswap** editor is inspired by several works in adversarial NLP that examine character swapping as a minimal change to text inputs that can degrade system performance (Belinkov and Bisk, 2018; Ebrahimi et al., 2018). Our implementation randomly swaps 1 to 3 lower-case ASCII characters per input for other ASCII characters, selecting the least likely of 100 alternatives under the GPT-2 language model (Radford et al., 2019).

VIPER is a character-swapping algorithm informed by visual closeness, inspired by a common strategy used to avoid keyword filters, for example in online forums Eger et al. (2019). The VIPER algorithm replaces random characters with their nearest neighbors among embeddings based on their glyph e.g. l→1 and 0→O. We further bias the open source implementation toward visual closeness by randomly swapping between 1 and 3 characters, with the probability of each swap weighted by its visual similarity.

The **AddCos** system uses word embedding distance to replace a single word with a paraphrase. The algorithm is adapted from a machine translation metric that measures the fit of words that are not in a reference, using the cosine similarity of the proposed replacement and the sum of vectors for sentence context (Apidianaki et al., 2018). We adapt the open source implementation as a machine editor, obtaining candidate replacements from the Penn Paraphrase Database (Ganitkevitch et al., 2013) and selecting the one best replacement.

3.2.2 Rewriting editors

Machine translation (MT) has recently become reliable and on-par with human translation capabilities in some cases (Bojar et al., 2018). We utilized **round trip MT** (from source English text to another language and then back) as a type of text editor. Three of the authors performed a blind assessment of approximately one hundred candidate languages available from an online MT provider

and determined that $en \rightarrow pt \rightarrow en$ is a high-quality round trip route.

A **variational autoencoder (VAE)** learns a semantically meaningful latent space. We use an implementation[2] based on the model of Zhang et al. (2017) to train a VAE for each domain with 200,000 sequences of up to 40 tokens. Edits are obtained by encoding an original sentence and sampling from the latent distribution.

Syntactically controlled paraphrase networks (SCPNs) encode sentences and decode them according to a target constituency parse (Iyyer et al., 2018). Unlike swapping editors, this system introduces syntactic variation. Using the open source code and default templates, we generate ten paraphrases per sentence. We select the paraphrase with the best GPT-2 language model score.

3.3 Modeling Human Perception

We evaluate a set of automatic metrics as models of human perception. To test a metric as a model of the collected judgments, the metric scores each edited sentence and chooses the item in the pair with the better score. The choice is compared to the judge's preference.

In addition, we learn a combination system that scores sentences by weighting each component metric. One combination is learned for each task, using the data from both domains. The data is split into a training set of 80% used for fitting the combination, a validation set of 10% and a final test set of 10% of examples. For items repeated among judges, all six instances are assigned to the same partition.

Our objective function, maximizing agreement on AB tests, is neither continuous, smooth, nor particularly amenable to a logistic transform. We search for our mixture parameters using the Dlib `MaxLIPO+TR` Lipschitz function and trust region search algorithm (King, 2009). The model is optimized to minimize the errors in the training set with an L1 regularization term. A best combination is selected using forward feature selection and validation set accuracy.

4 Experiments

We examine text similarity and fluency metrics that originate from several tasks in NLP as possible models of human perception. We first present the portfolio of metrics we use.

4.1 Measures of Meaning Preservation

Levenshtein edit distance measures the minimum number of character operations needed to change one string into another (Levenshtein, 1966). We compute both the classical Levenshtein distance over *character* edits and *word* edits (WER).

NLP Task Metrics. We evaluated several metrics used to measure the quality of NLP system output compared to a human reference for tasks including machine translation, summarization, and image captioning. **BLEU** is a machine translation evaluation method based on word n-gram precision, with a brevity penalty (Papineni et al., 2002). The **METEOR** metric uses stemming and WordNet synsets to characterize acceptable synonymy in translation (Banerjee and Lavie, 2005). **CIDEr** also uses stemming and incorporates importance weighting for ngrams based on corpus frequency (Vedantam et al., 2015). The **ROUGE-L** metric, used in summarization and image captioning, is based on longest common subsequence between a reference and hypothesis (Lin, 2004). **ChrF** and variants like chrF++ compare bags of character and ngram substrings to capture sub-word similarity without language-specific resources (Popović, 2016, 2017). The **BEER** metric is trained to correlate with human judgment at a sentence level using features like character n-grams and permutation trees that are less sparse at that level (Stanojević and Sima'an, 2014).

Neural Network-based Similarities. Recent work uses trained, neural-network vector representations to quantify semantic similarity. We experiment with three based on BERT, a neural network trained on Wikipedia and the Google Books Corpus (Devlin et al., 2019). **BERTScore** computes an F1-based similarity score between the contextual embeddings for subword tokens in a candidate and reference sentence (Zhang et al., 2020). The metric can also be computed as **RoBERTaScore** using weights from RoBERTa pretraining (Liu et al., 2019). **BLEURT** fine tunes BERT to predict sentence-level machine translation quality scores (Sellam et al., 2020). **Sentence-BERT** measures similarity using a model finetuned with a paraphrase objective to create semantically meaningful sentence vectors that can be directly compared (Reimers and Gurevych, 2019).

[2] https://github.com/mitre/tmnt

	detect		meaning		preferred	
system	arxiv	reddit	arxiv	reddit	arxiv	reddit
addcos	**0.60**	0.48	0.74	0.64	0.04	0.03
mt	0.55	**0.59**	0.55	0.57	**0.21**	**0.10**
viper	0.54	0.56	**0.79**	**0.88**	0.00	0.01
charswap	0.46	0.46	0.67	0.67	0.01	0.01
scpn	0.21	0.35	0.17	0.18	0.02	0.06
vae	0.20	0.11	0.07	0.06	0.08	0.02

Table 2: Probability that system wins in A/B test or is preferred to original. Rows are sorted by first column.

4.2 Measures of Detectability

Detectability is a property of text in context, without regard for a reference. We evaluate language model scores, which measure fluency, as proxies for detectability.

We evaluate a **Kneser-Ney 5-gram** language model trained on a full English Wikipedia dump (Wikipedia contributors, 2020) with KenLM (Heafield, 2011). We estimate the model using modified Kneser-Ney smoothing without pruning. We also evaluate the language model score given by **GPT-2**, a large neural transformer-based language model trained on 8 million web pages (Radford et al., 2019). We use the technique described in Salazar et al. (2020) to obtain a **BERT Masked Language Model (MLM)** that accounts for the model's self-attention. We compute each language model score under two conditions: using only the edited sentence, and including one sentence before and after the edited sentence (+*context*).

Predictions from **BERT's Next Sentence Prediction (NSP)** task estimate the likelihood for a sequence of sentences. This classifier is trained to discriminate sequences of two sentences found in the pretraining corpus from sequences drawn using negative sampling (Devlin et al., 2019).

5 Results

Table 2 illustrates the relative success of the machine-editing systems. Success is measured using the number of A/B tests where an edit by the system was selected (for meaning) or the other item selected (for detectability), divided by the number of prompts involving the editor. *Preference* refers to only the portion of the detectability dataset where the edited text is compared to the original. The swapping algorithms are most often chosen as preserving meaning. The visual perturbations of VIPER have little effect on perception of meaning. The preference for these editors is not as strong on detectability items. For both tasks, the round-trip machine translation model is preferred in slightly over half of comparisons, while the VAE and SCPN perform quite poorly. One substantial difference in these conditional generation algorithms may be that MT is trained on web-scale data, while the others are trained in-house with relatively small datasets.

Among detectability items, human judges prefer an edited version over the original (`null` system) 4.9% of the time, 101 of 2054 relevant judgments. These prompts most commonly include the round trip machine translation editor, but all editing systems were preferred over the original at least once. Round-trip machine translation is picked over the original reference 20% of the time in ArXiv and 10% of the time in Reddit, suggesting that these outputs are more fluent or more typical for the domain than the original. For these items, the character swapping algorithms are most detectable.

5.1 Annotator Consistency

One hundred five prompts per task were presented to all judges to measure interannotator agreement. As judgments are made between constructed, randomized flips and pairwise tests, we compare to the raw prior of 50% agreement. For ArXiv, the probability of agreement among pairs of judges was 82.2% for meaning and 75.6% for detection. For Reddit, the probabilities were 86.7% and 75.9% respectively. The lower interannotator agreement in the science and technology domain may reflect lower familiarity with the subjects of the abstracts.

A consensus vote is determined by a plurality of the six judges, or randomly in cases of ties. The probability of agreement of a random judge with the consensus is reported in Table 3 as an upper bound for the performance of automatic systems.

5.2 Correspondence of Metrics with Human Judgment

Table 3 shows the correspondence of the best metrics with the 630 multiply-annotated prompts and includes the upper bound of human consensus. The table shows only metrics with accuracy within five items of the best. Table 4 shows agreement with the entire dataset and over the full set of systems tested. The meaning metrics are also evaluated as measures of detectability. At editing time, they can be used to estimate the detectability of a candidate edit. However, they are not practical as generic

	detect			meaning			tot
	arxiv	reddit	tot	arxiv	reddit	tot	
annotator consensus upper bound	0.838	0.848	0.843	0.884	0.917	0.900	0.871
max per source	0.781	0.794	0.788	0.843	0.870	0.857	0.822
RoBERTaScore/R	0.765	0.781	0.773	**0.837**	**0.870**	**0.853**	**0.813**
chrF	0.759	0.784	0.772	**0.841**	0.830	0.835	0.803
chrF/R	0.759	0.781	0.770	**0.835**	0.830	0.833	0.801
chrf++	0.759	**0.787**	0.773	0.824	0.830	0.827	0.800
BEER	0.733	**0.787**	0.760	0.808	**0.863**	0.836	0.798
METEOR	**0.775**	0.778	**0.776**	**0.843**	0.792	0.818	0.797
BERTScore/R	**0.775**	0.765	0.770	0.830	0.813	0.821	0.795
BERTScore/F	**0.775**	0.775	**0.775**	0.811	0.806	0.808	0.792
CIDEr	0.759	**0.787**	0.773	0.824	0.781	0.802	0.788
RoBERTaScore/F	0.762	**0.794**	**0.778**	0.795	0.781	0.788	0.783
BERTScore/P	**0.781**	0.756	0.768	0.798	0.768	0.783	0.776
RoBERTa MLM	**0.683**	**0.673**	**0.678**				
RoBERTa MLM +context	0.663	**0.676**	0.669				
GPT-2 +context	0.654	**0.679**	0.667				

Table 3: Probability of agreement of metrics with sets of 630 multiply-annotated human judgments. Consensus is an *upper bound*. Results differing from the best by five items or fewer are shown in bold.

	detect			meaning			tot
	arxiv	reddit	tot	arxiv	reddit	tot	
max per source	0.788	0.790	0.789	0.845	0.841	0.843	0.816
RoBERTaScore/R	0.781	0.773	0.777	0.826	0.835	0.831	**0.804**
chrF	0.763	0.754	0.758	0.832	**0.840**	**0.836**	0.797
BERTScore/R	0.773	0.758	0.766	**0.845**	0.813	0.829	0.797
chrF++	0.763	0.758	0.760	0.823	**0.841**	0.832	0.796
chrF/R	0.762	0.751	0.756	0.830	0.838	0.834	0.795
BERTscore/F	0.773	0.765	0.769	0.839	0.804	0.822	0.795
chrF++/P	0.763	0.764	0.764	0.815	0.833	0.824	0.794
ROUGE	0.775	0.763	0.769	0.825	0.813	0.819	0.794
chrF++/R	0.760	0.754	0.757	0.821	0.838	0.829	0.793
BLEU	0.765	0.765	0.765	0.817	0.824	0.821	0.793
chrF/P	0.767	0.765	0.766	0.814	0.826	0.820	0.793
BEER	0.753	0.758	0.756	0.810	**0.841**	0.826	0.791
Sentence-BERT	0.763	0.770	0.766	0.813	0.821	0.817	0.791
METEOR	0.779	0.766	0.772	0.828	0.791	0.809	0.790
edit distance	0.755	0.745	0.750	0.815	**0.840**	0.827	0.788
word error rate	0.766	0.757	0.762	0.810	0.817	0.814	0.788
BERTScore/P	0.775	0.766	0.770	0.822	0.779	0.800	0.785
CIDEr	0.769	0.758	0.764	0.817	0.793	0.805	0.784
RoBERTaScore/F	**0.788**	**0.790**	**0.789**	0.792	0.767	0.780	0.784
BLEURT	0.746	0.746	0.746	0.788	0.751	0.770	0.758
RoBERTaScore/P	0.773	0.781	0.777	0.730	0.700	0.715	0.746
RoBERTa MLM +context	0.663	**0.712**	**0.688**				
RoBERTa MLM	0.642	0.700	0.671				
GPT-2 +context	0.644	0.685	0.665				
BERT NSP $logit0(s_{i-1}, s_i)$	0.672	0.650	0.661				
BERT NSP $logit1(s_{i-1}, s_i)$	**0.676**	0.639	0.657				
BERT NSP $p(s_{i-1}, s_i)$	0.666	0.640	0.653				
BERT NSP $p(s_{i-1}, s_i)p(s_i, s_{i+1})$	0.651	0.636	0.643				
BERT NSP $logit0(s_i, s_{i+1})$	0.652	0.631	0.641				
BERT NSP $logit1(s_i, s_{i+1})$	0.654	0.628	0.641				
BERT MLM +context	0.640	0.638	0.639				
BERT NSP $p(s_i, s_{i+1})$	0.631	0.626	0.629				
BERT MLM	0.617	0.627	0.622				
GPT-2	0.586	0.642	0.614				
KenLM	0.565	0.568	0.567				

Table 4: Probability of agreement of metrics with 3600 human judgments. Results differing from the best by five items or fewer are shown in bold.

	arxiv		reddit		mean	
metric	accuracy	F1	accuracy	F1	accuracy	F1
max per source	0.902	0.252	0.895	0.229	0.899	0.241
RoBERTa MLM	0.870	0.231	0.862	0.220	0.866	0.226
GPT-2 +context	**0.902**	**0.252**	0.874	0.168	0.888	0.210
BERT MLM +context	0.867	0.243	0.802	0.158	0.835	0.201
BERT MLM	0.849	0.236	0.799	0.156	0.824	0.196
KenLM	0.767	0.216	0.691	0.150	0.729	0.183
GPT-2	0.806	0.213	0.812	0.150	0.809	0.181
BERT NSP $p(s_i, s_{i+1})$	0.828	0.234	0.764	0.123	0.796	0.178
BERT NSP $logit0(s_i, s_{i+1})$	0.749	0.194	0.722	0.112	0.736	0.153
BERT NSP $logit1(s_i, s_{i+1})$	0.754	0.202	0.714	0.104	0.734	0.153
RoBERTa MLM +context	0.862	0.078	**0.895**	**0.229**	0.879	0.153
BERT NSP $p(s_{i-1}, s_i)p(s_i, s_{i+1})$	0.793	0.196	0.714	0.104	0.754	0.150
BERT NSP $p(s_{i-1}, s_i)$	0.853	0.209	0.763	0.083	0.808	0.146
BERT NSP $logit1(s_{i-1}, s_i)$	0.762	0.164	0.718	0.088	0.740	0.126
BERT NSP $logit0(s_{i-1}, s_i)$	0.749	0.151	0.735	0.087	0.742	0.119
baseline	0.941	0.000	0.961	0.000	0.951	0.000

Table 5: Performance predicting which edits humans prefer over originals.

	train	dev	test
meaning			
combo	0.857	0.844	0.830
ChrF	0.839	0.831	0.785
detect			
combo	0.689	0.752	0.724
RoBERTa MLM +context	0.678	0.731	0.687

Table 6: Combo performance results.

	$\sigma\|w\|$	w	component
meaning	0.25	0.999	edit distance
	0.15	0.912	BERTScore/R
	0.013	0.0871	Sentence-Bert
	0.0044	0.0307	BERTScore/P
	0.0035	0.0046	BLEURT
	0.0018	0.0354	RoBERTaScore/R
	0.0012	0.0265	RoBERTaScore/F
	0.0012	0.0263	RoBERTaScore/P
detect	0.83	0.0161	RoBERTa MLM +context
	0.13	0.577	BERT NSP $p(s_i, s_{i+1})$
	0.11	0.462	BERT NSP $p(s_{i-1}, s_i)$
	0.0069	0.0106	GPT-2 +context

Table 7: Components in the detect and meaning combo systems ranked by influence ($\sigma|w|$).

detection models sniffing out machine edits in the wild where no original is available.

Several automatic metrics show good correspondence with meaning. The best systems include large, neural models intended to capture subtle synonymy as well as simple metrics like chrF. In general, the recall component of BERTScore-based metrics correlates better than the precision component. Though the BLEURT metric is trained to predict human judgements of translation quality, it seems a poor fit for perceptions of meaning preservation in our dataset.

Applied to the detection task, the reference-informed metrics also approach the upper bound of human consensus. Using RoBERTaScore as a single model of both meaning preservation and detectability reaches over 81% agreement with consensus.

The language model metrics fall behind in performance on detection but still perform well above the level of chance. We find that including additional context improves performance for the same system and the large, neural models greatly outperform the traditional 5-gram model. Across the board, models with RoBERTa training perform better than their BERT-based counterparts.

Table 5 shows performance for predicting the detection items for which the judge preferred the edited text to the original. A baseline system that always selects the original gets around 95% accuracy, but cannot identify an edited text that a human accepts as a substitute. All of the detection systems tested were able to identify some substitutable edits. The best overall are large language models with context, reaching 0.241 F1.

As shown in Table 6, learned combinations of metrics are able to achieve better performance than the single best metric for each task. The components of those systems are specified in Table 7, sorted by their importance in the combination as calculated by the product of the standard deviation of their values (σ) and the magnitude of their weights (w).

6 Conclusion

We introduced a novel dataset of human judgments of machine-edited texts and initial models of those perceptions. A portfolio of automated metrics was

assessed for the ability to predict judges' preferences on meaning preservation and detectability. Automated measures of semantic similarity and fluency were evaluated individually and combined to produce factored models of human perception. Both meaning preservation and detectability are modeled within 6% accuracy of the upper bound of human consensus labeling. However, we observe that existing metrics poorly predict whether humans find an edited text to appear more human-like than the original.

Future work could explore deeper models and other factors of human perception not modeled by the metrics present here. For example, humans are sensitive to capitalization and correct spacing but many automatic metrics perform tokenization and normalization. Direct modeling of human perception drives understanding of human factors involving text variation. Adaptive models of human text perception would enable text editing to target understanding by individual readers.

References

David Ifeoluwa Adelani, Haotian Mai, Fuming Fang, Huy H Nguyen, Junichi Yamagishi, and Isao Echizen. 2020. Generating sentiment-preserving fake online reviews using neural language models and their human- and machine-based detection. In *International Conference on Advanced Information Networking and Applications*, pages 1341–1354. Springer.

Moustafa Alzantot, Yash Sharma Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Marianna Apidianaki, Guillaume Wisniewski, Anne Cocos, and Chris Callison-Burch. 2018. Automated paraphrase lattice creation for hyter machine translation evaluation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 480–485. Code found at https://github.com/acocos/pp-lexsub-hytera/.

L. R. Bahl, F. Jelinek, and R. L. Mercer. 1983. A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(2):179–190.

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, Ann Arbor, Michigan. Association for Computational Linguistics.

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

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 272–303, Belgium, Brussels. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Yin Cui, Guandao Yang, Andreas Veit, Xun Huang, and Serge Belongie. 2018. Learning to evaluate image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5804–5812.

Shrey Desai, Jiacheng Xu, and Greg Durrett. 2020. Compressive summarization with plausibility and salience modeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6259–6274, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia. Association for Computational Linguistics. Code available at https://github.com/AnyiRao/WordAdver.

Steffen Eger, Gözde Gül Şahin, Andreas Rücklé, Ji-Ung Lee, Claudia Schulz, Mohsen Mesgar, Krishnkant Swarnkar, Edwin Simpson, and Iryna Gurevych. 2019. Text processing like humans do: Visually attacking and shielding NLP systems. 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 1634–1647, Minneapolis, Minnesota. Association for Computational Linguistics.

Angela Fan, David Grangier, and Michael Auli. 2018. Controllable abstractive summarization. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 45–54, Melbourne, Australia. Association for Computational Linguistics.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 111–116, Florence, Italy. Association for Computational Linguistics.

Hardy Hardy and Andreas Vlachos. 2018. Guided neural language generation for abstractive summarization using Abstract Meaning Representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 768–773, Brussels, Belgium. Association for Computational Linguistics.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

Yu-Lun Hsieh, Minhao Cheng, Da-Cheng Juan, Wei Wei, Wen-Lian Hsu, and Cho-Jui Hsieh. 2019. On the robustness of self-attentive models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1520–1529.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1808–1822, Online. Association for Computational Linguistics.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. 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 1875–1885. Code found at https://github.com/miyyer/scpn.

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

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8018–8025.

Davis E. King. 2009. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758. Global search described at http://blog.dlib.net/2017/12/a-global-optimization-algorithm-worth.html.

Ashutosh Kumar, Kabir Ahuja, Raghuram Vadapalli, and Partha Talukdar. 2020. Syntax-guided controlled generation of paraphrases. *Transactions of the Association for Computational Linguistics*, 8:329–345.

Qi Lei, Lingfei Wu, Pin-Yu Chen, Alexandros G Dimakis, Inderjit S Dhillon, and Michael Witbrock. 2019. Discrete adversarial attacks and submodular optimization with applications to text classification.

Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics doklady*, 10(8):707–710.

Zichao Li, Xin Jiang, Lifeng Shang, and Hang Li. 2018. Paraphrase generation with deep reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3865–3878, Brussels, Belgium. Association for Computational Linguistics.

Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2018. Deep text classification can be fooled. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4208–4215.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ryan Lowe, Michael Noseworthy, Iulian Vlad Serban, Nicolas Angelard-Gontier, Yoshua Bengio, and Joelle Pineau. 2017. Towards an automatic turing test: Learning to evaluate dialogue responses. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1116–1126.

Paul Michel, Xian Li, Graham Neubig, and Juan Pino. 2019. On evaluation of adversarial perturbations for sequence-to-sequence models. 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 3103–3114, Minneapolis, Minnesota. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*.

Maja Popović. 2016. chrF deconstructed: beta parameters and n-gram weights. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 499–504, Berlin, Germany. Association for Computational Linguistics.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

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

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1085–1097.

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.

Horacio Saggion. 2017. Automatic text simplification. *Synthesis Lectures on Human Language Technologies*, 10(1):1–137.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Miloš Stanojević and Khalil Sima'an. 2014. BEER: BEtter evaluation as ranking. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 414–419, Baltimore, Maryland, USA. Association for Computational Linguistics.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. CIDEr: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

Ashish Venugopal, Jakob Uszkoreit, David Talbot, Franz Och, and Juri Ganitkevitch. 2011. Watermarking the outputs of structured prediction with an application in statistical machine translation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1363–1372, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Wikipedia contributors. 2020. Wikipedia, the free encyclopedia. [Online; 20200720 enwiki dump, accessed 11-August-2020].

Alex Wilson, Phil Blunsom, and Andrew Ker. 2015. Detection of steganographic techniques on twitter.

In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2564–2569.

Alex Wilson, Phil Blunsom, and Andrew D Ker. 2014. Linguistic steganography on twitter: hierarchical language modeling with manual interaction. In *Media Watermarking, Security, and Forensics 2014*, volume 9028, page 902803. International Society for Optics and Photonics.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *arXiv preprint arXiv:1905.12616*.

Guoyang Zeng, Fanchao Qi, Qianrui Zhou, Tingji Zhang, Bairu Hou, Yuan Zang, Zhiyuan Liu, and Maosong Sun. 2020. Openattack: An open-source textual adversarial attack toolkit. *arXiv preprint arXiv:2009.09191*.

Huangzhao Zhang, Hao Zhou, Ning Miao, and Lei Li. 2019a. Generating fluent adversarial examples for natural languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5564–5569, Florence, Italy. Association for Computational Linguistics.

Huangzhao Zhang, Hao Zhou, Ning Miao, and Lei Li. 2019b. Generating fluent adversarial examples for natural languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5564–5569.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yizhe Zhang, Dinghan Shen, Guoyin Wang, Zhe Gan, Ricardo Henao, and Lawrence Carin. 2017. Deconvolutional paragraph representation learning. In *Advances in Neural Information Processing Systems*, pages 4169–4179.

Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2018. Generating natural adversarial examples. In *International Conference on Learning Representations*.