Let's Fool That Stupid AI
Adversarial Attacks against Text Processing AI

Ulrich Schade, Albert Pritzkau, Daniel Claeser, Steffen Winandy

Introduction

AI systems help humans to detect and to recognize information in data, e.g. in medical imaging. Such recognition often involves a categorization so that the human user of the AI system only needs to take a look at the information that is categorized as relevant. Furthermore, the recognition abilities of AI often surpass those of its user. An AI system trained for such cases can detect and recognize objects hidden under a camouflage and thus imperceptible to the human eye. However, it must be emphasized that the detection abilities of AI systems do not necessarily trump human abilities. There are many scenarios in which humans excel and AI systems blunder. In principle, this is not a problem, as long as such scenarios can be identified so that AI systems can be assigned to those tasks in which they perform strongly. But as always, reality is more complex. AI systems that generally achieve great results may be confronted with specific inputs in specific situations so that they fail. If this happens, and if the input was designed with the purpose of causing an AI system to fail, this constitutes a so-called "adversarial attack".

In their paper on adversarial attacks, Goodfellow, Slenz and Szegedy write: "Szegedy et al. (2014b) made an intriguing discovery: several machine learning models, including state-of-the-art neural networks, are vulnerable to *adversarial examples*" (Goodfellow et al. 2015: 1). The authors subsequently express not only surprise but also disappointment with the discovery: "These results suggest that classifiers based on modern machine learning techniques, even those that obtain excellent performance on the test set, are not learning the true underlying concepts that determine the correct output label. Instead, these algorithms have built a Potemkin village that works well on naturally occuring (sic) data, but is exposed as a fake when one visits

points in space that do not have high probability in the data distribution. This is particularly disappointing because a popular approach in computer vision is to use convolutional features as a space where Euclidean distance approximates perceptual distance" (ibid: 2).

Adversarials are not only a problem for applications, i.e. AI classifiers, they also damage belief in the "intelligence" of modern AI. However, one might argue that image processing is quite near to the sensor level so that "intelligence" might not apply within image processing AI. Such an argumentation might be extended to the remark that human language acquisition incorporates the development of concepts and semantics, so that intelligence is more likely to apply within a text processing AI than an image processing one. Although Goodfellow et al. (2015) presented and discussed adversarial examples for classification tasks in the image recognition domain, adversarials do also occur in the area of text classification (Liang et al. 2018; Xu et al. 2020). The paper at hand is meant to add a small contribution to answer the question about why text processing AIs also are prone to adversarials.

In the following, we will take a closer look at adversarial attacks. We will provide a definition and illustrate the attacks through some examples (section 2). Then, in section 3, we will discuss how to generate such attacks from a mathematical and technical point of view. We do this with a focus on text classification applications. Adversarial attacks, however, have a dimension beyond mathematics: they only succeed if not only the AI is fooled, but also the human is not. Liang et al. (2018) call this feature "utility-preserving". In order to understand "utility-preserving" better, we will take a look at information processing and compare the information processing of an AI based on "deep learning" with human information processing, again with a focus on text classification (section 4). We then will add the linguistic perspective to that comparison (section 5). In section 4 as well as in section 5, we will use the insights gained to suggest ways to generate adversarials. The chapter will end in a discussion of lessons learned (section 6).

1. Adversarial Attacks: Definition and Examples

In their review on adversarial attacks and defenses, Xu et al. (2020: 1) provide the following definition: "Adversarial examples are inputs to machine learning models that an attacker intentionally designed to cause the model to make mistakes". The previously mentioned paper by Goodfellow and co-

authors (Goodfellow et al. 2015) provides the standard example from the field of image recognition: a picture of a panda, classified as "panda" with 57.7% confidence, is intentionally but only slightly changed (in a way that a human observer would not perceive). As result, the panda is classified as "gibbon" with 99.3% confidence (ibid: 3, figure 1).

The definition says that the model is caused "to make mistakes". In the example, that means that the panda is classified as "gibbon". However, this is only a mistake since the original picture shows a panda, and since a human would say that the picture still shows a panda even after the slight change. If the change would cause the human to also classify the changed picture as a picture showing a gibbon, we would not call it a mistake. Liang and co-authors call this "utility-preserving" and further explain this property with respect to text classification: "Utility-preserving means that the semantics of the text should remain unchanged and human observers can correctly classify it without many efforts. Consider for instance a spam message advertising something. Its adversarial version should not only fool a spam filter, but also effectively deliver the advertisement" (Liang et al. 2018: 4208).

Liang and co-authors use insertion, modification, and deletion (removal) of characters or words in order to change a text in such a way that a Deep Neural Network (DNN) classifier is fooled, but at the same time a human might not even notice the difference. In the following example, the authors added a whole sentence (marked in red in the original source, now underlined) that caused the DNN to classify the text, which was originally classified correctly as being from the topic area "Means of Transportation" (confidence 99.9%), as a text from the topic area "Film" (confidence 90.2%), cf. Liang et al., 2018, p. 4210, figure 4:

> The APM 20 Lionceau is a two-seat very light aircraft manufactured by the French manufacturer Issoire Aviation. Despite its classic appearance it is entirely built from composite materials especially carbon fibers. Designed by Philippe Moniot and certified in 1999 (see EASA CS-VLA) this very light (400 kg empty 634 kg loaded) and economical (80 PS engine) aircraft is primarily intended to be used to learn to fly but also to travel with a relatively high cruise speed (113 knots). Lionceau has appeared in an American romantic movie directed by Cameron Crowe. A three-seat version the APM 30 Lion was presented at the 2005 Paris Air Show. Issoire APM 20 Lionceau.

In their paper, Liang and co-authors focus on the technical and mathematical aspects of adversarial examples in text classification. In the next section, we will follow their lead and discuss how adversarials can be generated. However, after that we broaden the picture and add the cognitive point of view. In order to take the above-mentioned "utility-preserving" into account, it is not sufficient to explain the mathematics of the DNN classifiers' failures only. It is also necessary to discuss why and under which conditions DNN classifiers are fooled but humans are not.

2. Generating Adversarial Attacks

In order to discuss how to generate adversarial attacks, we first take a look at the field of Explainable AI. Explainable AI is a double-edged sword in the context of adversarial attacks. It allows the analysis of AI applications so that we get a better idea why a certain application generates the results it does. Thus, explainable AI can be used to identify the "weak" spots of AI applications, in particular those input patterns that cause strange and undesired results. If these spots are known to the developer of the application, this knowledge allows them to fix the problem. However, if the spots are known to an attacker (and not to the user), they can be exploited for adversarial attacks.

Explainable AI differentiates between "white box" systems and "black box" systems. A system at hand which can be analyzed directly is a "white box" system, but if we only can observe the system's reactions to given inputs it is a "black box" system. Liang et al. (2018: 4209f.) describe their mathematical approach (a) to identify the most significant ("hot") characters for manipulating characters to generate adversarial attacks on the character level and (b) to identify the most significant ("hot") words and phrases for manipulating on the word level to generate word-level adversarials. Here, we would like to focus on word-level manipulations. In short, a word (or a phrase) is highly significant ("hot") if it contributes highly to cause a specific classification. Mathematically, these words can be identified by calculating cost gradients (cf. Baehrens et al. 2010) or by alternatives like "layer-wise relevance propagation" (LRP) (Arras et al. 2017). Liang et al. (2018: 4211, figure 6) provide the following example text which is classified as a text from the topic area "Film".

> Edward & Mrs. Simpson is a seven-part British television series that dramatises the events leading to the 1936 abdication of King Edward VIII of the United Kingdom who gave up his throne to marry the twice divorced American Wallis Simpson. The series made by Thames Television for ITV was originally broadcast in 1978. Edward Fox played Edward and Cynthia Harris portrayed Mrs. Simpson. Edward & Mrs. Simpson.

The interesting thing about this example is that the DNN's confidence for the "film" classification drops from 95.5% to 60.5% if the word "British" (marked in blue in the original figure, now underlined) is deleted from the text. Thus, the word "British" is identified as a "hot" word by means of calculation, and its deletion moves the text in the classification vector space a long way towards the hyperplane that delineates the border of class "Film", so that it can be quite easily tipped over that border by further manipulations. We would like to remark here that, for a human, the word "British" seems to be superfluous. In contrast to words like "television" or "broadcast", humans would not assume that this word contributes that heavily to the classification as a "Film" text.

In the case of a "black box" system, a model of the system needs to be developed. In a first step, probes are used: the system is confronted with specific inputs and the reactions of the system to those inputs are noted. In our case, texts are presented as inputs and the corresponding classifications are the results. In a second step, the collected pairs of inputs and corresponding results can be used to train a model of the "black box" classifier, e.g. by a second DNN. The trained model then can be used to predict the classifier's reactions to other inputs. If the model is interpretable (and thus a so-called "Global Surrogate Model"), the predictions can be calculated out of the model. This then allows the mathematical identification of the "hot" words, cf. Ribeiro et al. (2016) or Alain/Bengio (2016) for mathematical details. The model also can be regarded as a "white box" system. As such it can be analyzed, e.g., by the gradient approach (Baehrens et al. 2010) or the LRP approach (Arras et al. 2017) as mentioned above. Unfortunately, trained models are seldom interpretable. However, they might be nevertheless locally interpretable. Local interpretability allows the calculation of the predictions on a local base and, thus, it allows the identification of candidates for "hot" words on a local level. Local level "hot" candidates are words whose manipulation might tip the classification from one given class to another given class. In the end, however, the "hot" word candidates need to be tested against the

"black box" classifier, since all the steps towards their identification add uncertainty to the equation and the local interpretability might not cover a large enough part of the vector space. So, in the case of "black box" systems, in contrast to "white box" systems, text samples which look promising for generating adversarial examples cannot be based on calculated "hot" words, but have to be detected by mixture of modelling and calculation, educated guessing, approximations, sequences of trial and error, and, of course, pure luck.

3. Differences in the Process of Text Classification between AI Classifiers and Humans

After having generated "may be"-adversarial examples by a mathematical approach, we have to consider "utility-preserving". This can be done by sorting out all those "may be"-adversarials that would also cause humans to change the classification. Alternatively, we could take "utility-preserving" into account from the beginning and try to generate only proper adversarials. In order to do so, we have to consider the differences between AI information processing and human information processing in general, and AI text classification and human text classification in particular, with the goal of better understanding the conditions that favor "utility-preserving".

We will start our comparison by considering *similarity aspects*. We begin with similarity aspects in phoneme recognition and character (letter) recognition. The phoneme /n/ is more similar to the phoneme /m/ than to the phoneme /f/ since /n/ and /m/ are voiced nasals. They only differ with respect to their place of articulation, [alveolar] for /n/ and [bilabial] for /m/. In contrast, /f/ is a voiceless fricative with [labiodental] as its place of articulation. In sum, /n/ and /m/ differ in one phonological feature, whereas /n/ and /f/ differ in all three. Consequently, a human confronted with phoneme /n/ errs more often by "hearing" /m/ instead of /n/ than "hearing" /f/ instead of /n/. A similar statement holds true for recognizing characters (letters and numerals). A human may have problems to distinguish a capital "o" ("O") from a zero ("0") but less so from an "m". A system that tries to identify these numerals and letters on a (filthy) license plate might have the same problems, but for a system that operates on digitalized text, all the numerals and letters are symbols of identical distance. As an example, the change of "APT40" to "APTA0" will be recognized by a system but might be overlooked

by a human reader. Even more so, the switching of two letters in the middle of a word is also often ignored by humans during reading, as shown by Grainger/Whitney (2004) in an article with the telling title "*Does the huamn mnid raed wrods as a wlohe?*" The reason for this is simple. Human expert readers do not waste time recognizing and identifying one character after the other. Instead, we fixate (relevant) content words on the second or third character, skip function words, and jump to the next (relevant) content word (Rayner 1997). During this process, words are recognized by their "gestalt", and only those characters fixated are precisely identified (Brysbaert/Nazir 2005).

A glance at the human cognitive process of reading explains why manipulations on character level can evade human attention and how "utility-preserving" can be achieved. The same holds for manipulation on the word level. Again, similarity effects are at work. For example, the formal similarity between "*flamenco*" and "*flamingo*" can be exploited. If, in a text of the category "culture", "*flamenco*" is substituted with "*flamingo*", the classification of that text might change to "nature". Of course, the substitution of "*flamenco*" with "*duck*" then would have the same effect, but this would much more easily be noticed by a human and no "utility-preserving" would have been achieved. If a human reads a text, the next following words are predicted ("*In Madrid's Retiro Park you can always see people dancing …*"). This resembles text processing by "Generative Pre-trained Transformer 3" (GPT-3, Brown et al. 2020). However, if the gestalt of the predicted next word is similar to the word in the text, humans might "see" the predicted word and not the printed word, whereas GPT-3 would "see" the word as it is printed. GPT-3 would not "see" the predicted word.

Since we are discussing text classification, it also seems appropriate to discuss human categorization in comparison to the classification by AI. A relevant aspect here is *the clearness or fuzziness of categories*. Although phonemes constitute very clear sound categories in human speech recognition, words do not. Words signify semantic categories but the categories are "flexible", as was first demonstrated by Labov (1973). In contrast, AI classifiers often (but not necessarily) partition input with clearly cut boundaries between two categories. It is obvious that humans who categorize fuzzily and AI classifiers that categorize within sharp borders will come to different results in some cases. Although it has to be determined which kind of categorizing – fuzzy or clear and sharp – is more appropriate for a given task, it suggests itself that some of an AI's categorization results may surprise a human user who

assumes that an AI will carry out a task much more rapidly and with fewer lapses, but, apart from that, like a human.

Let us take a look at an example, namely the Socio-political and Crisis Events Detection Shared Task at the CASE@ACL-IJCNLP 2021 workshop (Haneczok et al. 2021). The task is about the categorization of texts. The categories involved represent socio-political and crisis events. Two of the categories are "Violent Demonstration" and "Property Destruction". Obviously, violent demonstrations more often than not include property destruction. As an example, consider the demonstrations at the 2017 G20 Summit in Hamburg. This situation led to foreseeable results. On the one hand, AI classifiers submitted to the shared task (and trained with a corpus that the submitters had to develop beforehand), often confused the classes mentioned, but had no problem separating them from texts representing categories like "Air Strike" (cf. Kent/Krumbiegel 2021). In order to tip the classification from one of the critical categories to the other, only a few changes in the occurrence or non-occurrence of specific words are necessary (e.g. "*some of the protesters set a car alight*" or "*some of the activists set a car alight*" vs. "*some criminals set a car alight*"). On the other hand, humans would want to assign the problematic texts to both categories, "Violent Demonstration" and "Property Destruction". If they were asked to choose only one of them, the classification selected would differ, not only with regard to the subjects doing the classification, but also as a function of context like in the experiments by Labov. Like the AIs, humans might be influenced by the explicit wording. However, humans are able to recognize some parts in a text as purely ornamental. For instance, in the example provided by Liang et al. (2018) given above, the text about an aircraft is ornamented with the remark that the same aircraft had a scene in a specific movie. Humans would not consider such ornaments for classification. An AI does. Therefore, smartly constructed ornaments can be used to create adversarials.

4. Linguistic Aspects

Similarity of characters due to their visual nature, or of words due to sharing letters in many of the same positions, influences the processing of words and texts in human language comprehension, unlike in the same processing run by text classification AIs. These differences can be exploited by generating adversarials, as has been discussed above. We will discuss in the

following how some linguistic aspects can contribute to this. We will focus on orthography/phonology, morphology, and pragmatics.

4.1 Orthography and Phonology

Weingarten (2011) discusses phonographic errors (in writing) in contrast to grammatographic errors. Phonographic errors occur if one substitutes a word with a sequence of letters that is pronounced like the target word but is spelled differently. If the error is a word, it is a homophone of the target word, but not a homograph ("*Mary Christmas*"). Phonographic errors can be used as adversarials. As an extreme example, let us assume that we have science fiction text snippets and want to assign each of them to a saga like Star Wars, Star Trek, Honorverse, Expanse etc. It is easy to assign the sentence "*During his training, Luke Skywalker saw a vision of his friends in danger*" to the Star Wars saga. However, a version with phonographic errors such as "*During his training, Lewk Skeyewolker saw a vision of his friends in danger*" might be problem for an AI. In contrast, a human would recognize the error and judge the writer as a jester (or as incompetent in spelling), but would not have a problem with the categorization. Substituting a "hot" word with a phonographic error might constitute a better adversarial than removing the word, since that is not always possible without changing a text's meaning. It might also be more effective than switching letters within the "hot" word since phonographic errors often have a larger Levenshtein distance to their target, and may thus withstand a preprocessing of the texts to be classified aimed at deleting typos.

4.2 Morphology

For a long time, preprocessing had been a smart idea to deal with morphological variations. In conventional, bag-of-words based classifiers (cf. Manning et al. 2008 or Jurafsky/Martin 2009), preceding current state-of-the-art DNN classifiers, words occurring in training and test data were primarily considered terminal symbols disregarding semantic or morphological relatedness. Highly inflecting languages such as German encode various information, such as case (in nouns and adjectives), person, tense and mode (in verbs) and number (in all these word classes), at the word-level, creating potentially dozens of word-forms for one lexeme. Thus, conventional classifiers based on word frequency and correlation between the appearance of

word forms and target label disregard relations between e.g. singular and plural inflected forms of their base lexeme by design, distorting statistics about the relevance of terms for a given class and tampering with feature selection when creating the vector space for classification. Preprocessing of texts, in particular lemmatization (the substitution of all the morphological forms with the base form), countered those problems.

The advent of word embeddings with Word2Vec (Mikolov et al. 2013) and its successors mitigated the issue of apparently distinct but related word forms by projecting words into a vector space capable of encoding similarities on various levels, including, but not limited to, semantical and morpho-syntactical aspects. Given sufficient training data, a DNN text classifier learns to systematically disregard non-relevant (for the classification task) morphological variations (Claeser 2021). While certain word forms of relevant terms might be required, the presented lexicon does not necessarily need to be exhaustive, i.e. number marking as in "match" vs. "matches" might be irrelevant when it comes to assigning a text to the category "Sports". While previous generations of word embeddings such as Word2Vec and FastText (Joulin et al. 2016) conflated occurrences of word forms including possible homographs into a single unigram word vector, state-of-the-art transformer models such as BERT (Devlin et al. 2019) additionally represent potential context information in multiple hidden layers shipped as a refinable pre-trained vector space model. Circumventing the common out-of-vocabulary problem (OOV), BERT stores sub word units, morphemes such as the English "-ing", in addition to a fixed-size full-form vocabulary to resolve unknown inflections of lexemes represented in the dictionary.

Embedding-based deep learning language models and applications such as text classifiers might be considered more robust to adversarial attacks. However, in an interview about GPT-3 and language processing AIs (Küchemann 2020), Sina Zarrieß argued for morphological preprocessing at least with regard to texts in languages like Finnish. Why? Distributions of inflected word forms that follow Zipf's Law (Zipf 1949) tend to produce few or no instances of less common values in morphological categories such as the subjunctive mood in German ("Konjunktiv"). Consequently, while a DNN classifier might be trained to disregard the "tense" or "person" category values in verb forms as in "soll – sollte" ("should", 1st person singular present and simple past, respectively), it might be tampered with using the conjunctive form "solle". Manipulating morphological mechanisms such as the German Umlaut, which is necessary to incorporate aspects such as number in certain

German noun classes as in "Haus – Häuser" (house – houses), while facing little hesitation by a human classifier and leading to rejection of the term by a conventional classifier, might be mitigated to a certain degree by BERT's incorporation of unigram character representations. The exact extent of recent models' ability to cope with such modifications, however, is currently still under investigation. Finally, phenomena such as separable prefixes of verbs in a number of Germanic languages, such as German, Icelandic and, to a certain degree, even English, pose a challenge even to DNN classifiers. For example, in "*Tom hörte sich die Symphonie an*" (Tom listened to the symphony), the verb is "*anhören*" and the separable and displaced prefix is "*an*". "*Anhören*" and "*hören*" (to hear) are semantically connected, but "*aufhören*" (to stop) like in "*Tom hörte danach mit dem Klavierspielen auf*" (Tom stopped playing piano afterwards) is not. So, in German, "hot" verbs might be substituted with a verb with such a prefix (or the phrases might be rearranged) so that a verb stem is considered by the classification which is not semantically related to the verb as a whole. Again, preprocessing, namely substituting all words with their base form (lemmatization), would suppress that problem.

4.3 Pragmatics

The most interesting linguistic aspect might be pragmatics. This can be illustrated by taking a look at GPT-3. In principle, GPT-3 takes a piece of text, the "prompt", and continues it. In their review of GPT-3, Marcus/Davis (2020) provided the following example:

> Prompt:
> *You are a defense lawyer and you have to go to court today. Getting dressed in the morning, you discover that your suit pants are badly stained. However, your bathing suit is clean and very stylish. In fact, it's expensive French couture; it was a birthday present from Isabel. You decide that you should wear*
>
> GPT-3's continuation:
> *the bathing suit to court. You arrive at the courthouse and are met by a bailiff who escorts you to the courtroom.*

Gwern (2020) argued that GPT-3's continuation is as it is since GPT-3 had learned the Gricean maxims (Grice, 1975), in particular, the maxim of relation (relevance): *Be relevant*: "Prompts should obey Gricean maxims of communication – statements should be true, informative, and relevant. One should

not throw in irrelevant details or non sequiturs, because in human text, even in fiction, that implies that those details are relevant, no matter how nonsensical a narrative involving them may be" (Gwern 2020). The Gricean maxims explicate Grice's cooperative principle, the backbone of effective conversational communication among people. The principle and with it the maxims are incorporated into texts. For example, if in an Agatha Christie novel something from the past is mentioned, this always is relevant, either for the solving of the mystery or for introducing red herrings. That everything is relevant is the essence of the maxim of relation. In the example by Marcus and Davis, GPT-3 pays heed to the assumed relevance of the reference to the bathing suit and transfers it into its continuation of the text. The problem is that the maxim of relation does not always hold. Sometimes, people are not cooperative. Sometimes people would like to impress the hearer (or the reader), e.g. by adding some pieces of superfluous information that hint at the speaker's (or the writer's) huge and impressive knowledge. Above, we labeled respective parts of texts as "ornamental". These parts are not relevant but serve different purposes, e.g. to make the speaker look good. Humans are trained to recognize these parts as such and would not consider them for text classification or text continuation. As has already been mentioned, current AIs do not recognize ornaments and therefore "exploit" them for the classification and, obviously, also for continuation. Thus, to repeat our already achieved insight, adversarials can be generated as ornaments. However, tempering against such kinds of adversarials might be possible by "learning" to recognize text ornaments.

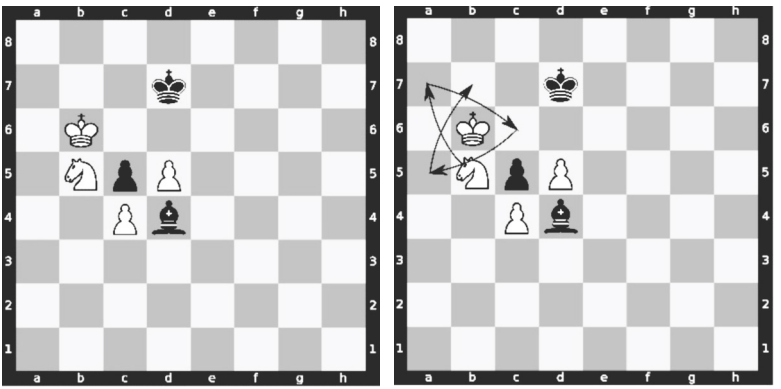
5. Conclusion

In our contribution, we discussed adversarials in the field of text classification. We referred to the mathematical approach on how to generate them and then focused on how to allow "utility-preserving". To do so, we took into account aspects of the human classification process as well as linguistic aspects. These examinations led to indications on how to generate adversarials against text classifications.

Goodfellow et al. (2015: 2) expressed their disappointment that DNNs in the field of image classification had not learned concepts, as demonstrated by the existence of adversarials. Adversarials also exist in the field of text classification (Liang et al. 2018; Xu et al. 2020). So, we may assume that text-

processing AIs might also not have learned concepts. GTP-3 had learned to build sentences and texts in English and other languages that are grammatically correct. It can even answer questions correctly, although it fails to adequately deal with questions that include an empty definite description (Who was the Roman emperor last year?). Text-processing AIs can learn to use language adequately on the orthographic, the morphological and the syntactic level by building vector space representations of a language's symbols, the words, the phrases, and the sentences. However, to complete the "semiotic triangle" (Ogden/Richards 1923), the reference to the real world is missing. During lexical development, a child resorts to cognitive development (Piaget 1923). We would like to suggest that a language-processing AI is not capable of developing concepts, since it does not establish references to the real world for its representations of language symbols and therefore cannot establish these references, since it lacks cognitive development. Adversarials strongly illustrate that inaptness.

Figure 1



In order to substantiate our claim, we would like to point to one of the most awe-inspiring achievements of DNNs, their mastery of board games like chess. Adversarials can even be found within this area of strength as figure 1 shows. The figure (left) displays a position from the final of the Top Chess Engine Championship (TCEC), Season 14, November 17th, 2018 to February 24th, 2019. In the final, the engine Stockfish 10 won narrowly by 50.5 to 49.5 against Leela Chess Zero (Lc0) (Schormann 2019). In the position displayed, Leela, playing White, is to move. The position is easily won but

Leela failed. The position is an adversarial for Leela. So, why is the position won? Simple effect-based reasoning shows the following: (a) White will win if the black pawn on c5 falls. (b) The pawn will fall if the white horse reaches b7 since then the knight attacks the pawn and in addition prevents the black king from defending the pawn from d6. (c) White can move its horse from b5 to a7 to c6 to a5 to b7 (right figure) to take the pawn. (d) Black cannot prevent all this. Leela is not able to do effect-based reasoning. It also does not calculate trees of moves like chess engines of the period before Deep Learning engines did. Leela decides on elaborate pattern matching. Pattern matching favors positions in which one's own pawns are advanced as far as possible. Thus, Leela is fooled and moves its d-pawn to d6, but then victory is lost and the draw is certain.

References

Alain, Guillaume, and Yoshua Bengio (2016). "Understanding Intermediate Layers using Linear Classifier Probes." In: *arXiv.org*. URL: https://arxiv.org/abs/1610.01644v4, (4. Version von 2018).

Arras, Leila, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek (2017). "'What is Relevant in a Text Document?': An interpretable machine learning approach". In: *PLoS ONE* 12, pp. 1–23.

Baehrens, David, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller (2010). "How to Explain Individual Classification Decisions." In: *Journal of Machine Learning Research* 11, pp. 1803–1831.

Brown, Tom B. et al. (2020). "Language Models are Few-Shot Learners." In: *arXiv.org*. URL: https://arxiv.org/abs/2005.14165v4.

Brysbaert, Marc, and Tatjana Nazir (2005). "Visual Constraints in Written Word Recognition: Evidence from the Optimal Viewing-position Effect." In: *Journal of Research in Reading* 28, pp. 216–228.

Claeser, Daniel (2021). *Zur Rolle der Flexionsmorphologie in der automatischen Klassifikation deutschsprachiger Textdokumente*. PhD Thesis, Rheinische Friedrich-Wilhelms-Universität Bonn.

Devlin, Jacob, 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. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, MN: Association for Computational Linguistics, pp. 4171–4186.

Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy (2015). "Explaining and Harnessing Adversarial Examples." In: *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*. Ed. by Yoshua Bengio and Yann LeChun. URL: arxiv.org/abs/1412.6572.

Grainger, Jonathan, and Carol Whitney (2004). "Does the huamn mnid raed wrods as a wlohe?" In: *Trends in Cognitive Science* 8, pp. 58–59.

Grice, Herbert Paul (1975). "Logic and Conversation." In: *Syntax and Semantics*. Volume 3. Ed. by Peter Cole and Jerry L. Morgan. New York: Academic Press, pp. 41–58.

Gu, Tianyu, Brendan Dolan-Gavitt, and Siddharth Garg (2017). "BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain." In: *arXiv.org*. URL: https://arxiv.org/abs/1708.06733.

Gwern (2020). "GPT-3 Creative Fiction". URL: https://www.gwern.net/GPT-3#expressing-uncertainty.

Haneczok, Jacek, Guillaume Jacquet, Jakub Piskorski, and Nicolas Stefanovitch (2021). "Fine-grained Event Classification in News-like Text Snippets Shared Task 2, CASE 2021." In: *Proceedings of the Workshop on Challenges and Applications of Automated Text Extraction of Socio-Political Event from Text (CASE 2021), co-located with the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021)*. Ed. by Ali Hürriyetğlu.

Joulin, Armand, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov (2016). "Bag of Tricks for Efficient Text Classification." In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. Volume 2, Short Papers.

Jurafsky, Dan, and James H. Martin (2009). Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics (2nd Edition), Upper Saddle River, NJ: Prentice-Hall.

Kent, Samantha, and Theresa Krumbiegel (2021). "CASE 2021 Task 2: Sociopolitical Fine-grained Event Classification using Fine-tuned RoBERTa Document Embeddings." In: *Proceedings of the Workshop on Challenges and Applications of Automated Text Extraction of Socio-Political Event from Text (CASE 2021), co-located with the Joint Conference of the 59th Annual Meeting of the Asso-*

ciation for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021). Ed. by Ali Hürriyetğlu.

Küchemann, Fridtjof, (2020). "Nimmt uns der Computer die Sprache ab?" In: *Frankfurter Allgemeine Zeitung*, November 26, 2020. URL: https://www.faz.net/aktuell/feuilleton/debatten/die-informatikerin-sina-zarriess-ueber-sprachmodelle-wie-gpt-3-17070713.html.

Labov. William (1973). "The Boundaries of Words and Their Meanings." In: *New Ways of Analyzing Variation in English*. Ed. by Charles-James Bailey and Roger W. Shuy, Washington, DC: Georgetown Press, pp. 340–373.

Liang, Bin, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi(2018). "Deep Text Classification Can Be Fooled." In: Jérôme Lang (ed.), Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18), pp. 4208–4215 (arXiv:1412.6572v3).

Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze (2008). *Introduction to Information Retrieval*. Cambridge, UK: Cambridge University Press.

Marcus, Gary, and Ernest Davis (2020). "GPT-3, Bloviator: OpenAI's Language Generator Has No Idea What It's Talking About." In: *MIT Technology Review*. URL: https://www.technologyreview.com/2020/08/22/1007539/gpt3-openai-language-generator-artificial-intelligence-ai-opinion.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean (2013). "Efficient Estimation of Word Representations in Vector Space." In: *International Conference on Learning Representations*. Scottsdale, AZ. URL: https://arxiv.org/abs/1301.3781.

Ogden, Charles Kay, and Ivor Armstrong Richards (1923). *The Meaning of Meaning*, London: Routledge & Kegan Paul.

Piaget, Jean (1923). *La langage et la pensée chez l'enfant*, Neuchâtel: Delchaux et Niestlé.

Rayner, Keith (1997). "Understanding Eye Movements in Reading." In: *Scientific Studies of Reading* 1, pp. 317–339.

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin (2016). "'Why Should I Trust You?': Explaining the Predictions of Any Classifier." *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Ed. by Balaji Krishnapuram and Mohak Shah. New York: Association for Computing Machinery. URL: https://arxiv.org/abs/1602.04938.

Schormann, Conrad (2019). "Computerschach: Inoffizielle Engine-WM TCEC – Finale LC0 vs. Stockfish". In: *Rochade Europa* 4/2019, pp. 42–43.

Szegedy, Christian, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus (2014). "Intriguing Properties of Neural Networks." In: 2nd International Conference on Learning Representations (ICLR). Banff, Canada. URL: https://arxiv.org/abs/1312.6199

Weingarten, Rüdiger (2011). "Comparative Graphematics." In: *Written Language and Literacy* 14, pp. 12–38.

Xu, Han, Yao Ma, Haochen Liu, Debayan Deb, Hui Liu, Jiliang Tang, and Anil K. Jain (2020). "Adversarial Attacks and Defenses in Images, Graphs and Texts: A Review." In: *International Journal of Automation and Computing* 17, pp. 151–178. URL: https://arxiv.org/abs/1909.08072.

Zipf, George Kingsley (1949): *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*, Cambridge, MA: Addison Wesley.