The Effectiveness of Uncased Tokenization for Clinical Notes

Cory Paik[1] and **Katharina von der Wense**[1,2]
[1]University of Colorado Boulder [2]Johannes Gutenberg University Mainz
{cory.paik, katharina.kann}@colorado.edu

Abstract

The impact of case-sensitive tokenization on clinical notes is not well understood. While clinical notes share similarities with biomedical text in terminology, they often lack the proper casing found in polished publications. Language models, unlike humans, require a fixed vocabulary and case sensitivity is a trade-off that must be considered carefully. Improper casing can lead to sub-optimal tokenization and increased sequence length, degrading downstream performance and increasing computational costs. While most recent open-domain encoder language models use uncased tokenization for all tasks, there is no clear trend in biomedical and clinical models. In this work we (1) show that uncased models exceed the performance of cased models on clinical notes, even on traditionally case-sensitive tasks such as named entity recognition and (2) introduce independent case encoding to better balance model performance on case-sensitive and improperly-cased tasks.

1 Introduction

Casing is one of many ways to increase the readability of written text. However, more often than not it is not a necessity – after all we communicate verbally without specifying, “capital a” in conversation. Language models (LMs) have fixed vocabularies. For them, case sensitivity is a trade-off between retaining case information to resolve case-sensitive ambiguities and improving efficiency as well as robustness by introducing case-invariance.

Case-sensitive tokenization is often seen as useful for resolving ambiguties, e.g., in named entitiy recognition (NER). However, for some types of text – such as clinical notes – the observed case is merely an interpretation of spoken language during transcription, which should be re-constructable from lower-case text with sufficient context and knowledge. This is distinct from applications where preserving case is absolutely necessary, such as in programming languages, where case is semantically significant. Moreover, when case-sensitive tokenization results in truncated sequences, then the task becomes more challenging due to missing tokens. For this application, it should be obvious that missing case information will always be easier to infer than truncated tokens.

Prior work in the biomedical domain has shown mixed results on the impact of casing on downstream tasks, with some work showing that cased models perform better on NER tasks (Lee et al., 2020), and others showing that uncased models perform better across the board (Beltagy et al., 2019). To the best of our knowledge, no work has been done to investigate the impact of casing on clinical notes. Clinical notes share some similarities with biomedical text such as PubMed articles with regards to their domain and in the sense that both are littered with case-sensitive acronyms. PubMed articles are much more polished and typically have proper casing, whereas clinical notes are often semi-structured and lack proper casing.

In this work we (1) show that uncased models match or exceed the performance of cased models on clinical notes within 1%, even on traditionally case-sensitive tasks such as NER, and (2) introduce independent case encoding to better balance the trade-off of potentially case-sensitive or improperly-cased tasks.

2 Related Work

General-Purpose Models While the original transformer introduced by Vaswani et al. (2017) used cased tokens, Devlin et al. (2019) introduced both an uncased and a cased variant of BERT. In their work, they do not explicitly discuss the trade-offs between the two, but used the cased version for named entity recognition (NER) tasks.[1] Many models have been released since BERT, using a

[1] This is further detailed in their readme github.com/google-research/bert.

Findings of the Association for Computational Linguistics: ACL 2025, pages 14986–14992
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

similar (or identical) architecture. While RoBERTa uses cased tokenization (Liu et al., 2019), most models that are not initialized from BERT use uncased tokenization (Lan et al., 2020; Geiping and Goldstein, 2022; Portes et al., 2023).

Note that we are only concerned with encoder models, not decoder models, which typically use cased tokenization for the output (Radford and Narasimhan, 2018; Radford et al., 2019; Brown et al., 2020) due to other motivating factors (e.g., the fact that humans prefer text with proper casing).

Biomedical Models Lee et al. (2020) introduce BioBERT– which is initialized from the BERT cased model and trained on PubMed Abstracts. While Lee et al. (2020) report that a cased vocabulary performs slightly better on downstream tasks, they do not provide any further analyses or ablations. Beltagy et al. (2019) introduce SciBERT, which uses the same architecture as BERT, but was pretrained from scratch on scientific text with a new tokenizer. They use cased models for NER and uncased for all other tasks following the original BERT methodology, but note that the uncased models sometimes performed slightly better in their initial testing than cased ones on NER. Gu et al. (2021) introduce PubmedBERT, which is trained from scratch on PubMed articles and diverges from previous work by using uncased tokenization. They do not provide ablations to support this decision, but note that the cased and uncased versions had similar performance in prelimenary experiments.

Clinical Models Alsentzer et al. (2019) introduce ClinicalBERT and BioClinicalBERT, the latter of which is based on BioBERT and, thus, cased. Note that others have also published models using the same ClinicalBERT namesake, so we will denote this ClinicalBERT as ClinicalBERT$_{\mathrm{A}}$. For example, Huang et al. (2020) train their model which we denote ClinicalBERT$_{\mathrm{H}}$ directly from BERT uncased on MIMIC-III.[2] At a high level, ClinicalBERT$_{\mathrm{H}}$ is pretrained longer and with longer sequences than ClinicalBERT$_{\mathrm{A}}$, but much less than BioClinicalBERT. Both ClinicalBERT$_{\mathrm{A}}$ and BioClinicalBERT are trained for less time than ClinicalBERT$_{\mathrm{H}}$ on MIMIC-III. Moreover, these do not have overlapping downstream tasks, so it is not straightforward to directly compare them. Wang et al. (2023) re-use the ClinicalBERT name to train a cased model from scratch on clinical notes, but they do not provide much detail about the training process or data.

3 Independent Case Encoding

We introduce a novel model extension by adding independent case encoding to the existing uncased BERT model. This is implemented similar to the existing positional and token type embedding in that is an additive embedding with 4 possible values: uppercase, lowercase, capitlized, or mixed case. This case information is found during tokenization. While training from scratch may provide better results, in our experiments we simply add this embedding to the existing BERT uncased models. For this reason, we only provide independent casing results on models that continue pre-traininig on MIMIC-III.

The motivation for independent case encoding is to retain some useful case information without being overly burdened when the case is not important. This gives the model the ability to easily ignore case information, unlike the cased tokenization.

4 Experiments

4.1 Dataset Statistics

To analyze the prevalence of improper casing in clinical data, we count the total number of tokens and the number of sequences with greater than 128 tokens for MIMIC-III and the downstream datasets. We also include these metrics for the C4 common crawl dataset (Raffel et al., 2020) to serve as a baseline of tokenizing a different open-domain corpus.

4.2 Pretrained Models

To investigate the impact of case-sensitive tokenization on clinical notes, we train 2 models initialized from BERT base cased (BERT$_{\mathrm{C}}$) and BERT base uncased (BERT$_{\mathrm{U}}$) on MIMIC-III (Johnson et al., 2015, 2016; Goldberger et al., 2000), which is a freely available database with thousands of patients from an ICU between 2001 and 2012. It contains patient data, such as demographics, as well as structured timeseries data, such as vital signs, medications, imaging reports, and free-text notes by the nurses and doctors. We preprocess the MIMIC-III dataset using scripts provided by Alsentzer et al. (2019) and create pretraining data using the parameters used by Huang et al. (2020) using the original BERT implementation.[3] Following Huang et al. (2020) we train the models for 100k steps with a

[2] These are concurrent work.

[3] github.com/google-research/bert

		Examples	Sequence Length > 128			Num Tokens (k)		
			Cased	Uncased	↓ %	Cased	Uncased	↓ %
c4	Train	364868892	269147123	263073464	2.257	178.56B	171.21B	4.116
	Validation	364608	269112	263040	2.256	177.87M	170.61M	4.081
MIMIC-III	Train	2083112	250761	159363	36.448	1.28B	1.13B	11.376
MedNLI	Train	11232	120	98	18.333	446.017	417.671	6.355
	Validation	1395	26	16	38.462	58.369	54.562	6.522
	Test	1422	15	15	0.000	53.799	50.690	5.779
i2b2/2006	Train	44948	92	58	36.957	699.924	594.722	15.030
	Validation	4994	10	6	40.000	78.670	66.682	15.238
	Test	18096	32	25	21.875	306.989	268.836	12.428
i2b2/2010	Train	14684	21	16	23.810	248.120	219.393	11.578
	Validation	1632	1	1	0.000	27.584	24.390	11.579
	Test	27625	39	31	20.513	484.395	433.451	10.517
i2b2/2014	Train	45794	668	591	11.527	960.979	872.000	9.259
	Validation	5088	80	74	7.500	107.213	97.167	9.370
	Test	32587	495	447	9.697	690.789	623.234	9.779

Table 1: Datasets statistics for all dowstream tasks. For MedNLI, we use the standard train, validation, and test splits provided by (Romanov and Shivade, 2018). For the i2b2 datasets, we split the training data into a 90/10 train/validation split and use the standard test split. For MIMIC-III, we report the sequence length of individual sentences to better represent the number of truncated sequences. All other datasets use the full sequence length as used by the model. For all datasets, note that uncased tokenization results in fewer overall tokens. For MIMIC-III, we report the number of tokens in billions, for all other datasets they are in thousands.

	MedNLI	i2b2 2006	i2b2 2010	i2b2 2014	MNLI MM	MNLI M
BERT$_C$	78.692	**94.031** ± **0.668**	83.710 ± 0.11	92.234 ± 0.581	83.452	82.985
BERT$_U$	78.129	90.846 ± 0.605	85.077 ± 0.081	93.318 ± 0.405	83.208	82.914
BERT$_C$ + MIMIC 200k	80.028	91.660 ± 1.875	87.197 ± 0.173	92.192 ± 1.128	82.618	82.354
BERT$_C$ + MIMIC 900k	81.294	91.324 ± 1.179	87.812 ± 0.110	92.881 ± 0.203	81.906	81.172
BERT$_U$ + MIMIC 200k	81.224	91.499 ± 0.755	88.042 ± 0.149	93.747 ± 0.072	83.411	83.107
BERT$_U$ + MIMIC 900k	**83.333**	92.533 ± 0.321	87.969 ± 0.152	93.530 ± 0.091	82.567	82.313
BERT$_I$ + MIMIC 200k	80.380	91.273 ± 1.085	87.688 ± 0.119	**93.901** ± **0.118**	**83.767**	**83.240**
BERT$_I$ + MIMIC 900k	82.841	**92.354** ± **1.776**	**88.192** ± **0.237**	93.723 ± 0.151	82.577	82.252

Table 2: Downstream performance of all models on different downstream tasks. Scores are Accuracy for MedNLI and Exact F1 for all i2b2 tasks. The best model is selected based on the validation set. Scores within the standard deviation of the best score are in bold.

sequence length of 128 and a batch size of 64 followed by 100k steps with a sequence length of 512 and a batch size of 16. We denote these models as BERT$_C$ + MIMIC 200k and BERT$_U$ + MIMIC 200k, respectively. These initial experiments show the uncased BERT model converging slower than the cased model, so we also train both models for a total of 900k steps: for 100k steps with a sequence length of 128 and a batch size of 64 followed by 800k steps with a sequence length of 512 and a batch size of 16. With this extended training, all models converge to a masked language modeling loss of approximately 0.25. We denote these models as BERT$_C$ + MIMIC 900k and BERT$_U$ + MIMIC 900k. The uncased BERT models with independent case encoding are denoted as BERT$_I$ + MIMIC 200k and BERT$_I$ + MIMIC 900k. The BERT base uncased models with independent case embedding are denoted as BERT$_I$ + MIMIC 200k and BERT$_I$ + MIMIC 900k.

4.3 Downstream Tasks

Tasks We evaluate the performance of the models on 4 downstream tasks in the clinical domain: MedNLI (Romanov and Shivade, 2018), the i2b2 2006 de-identification task (Uzuner et al., 2007), the i2b2 2010 concept extraction task (Uzuner et al., 2011), and the i2b2 2014 de-identification task (Stubbs and Özlem Uzuner, 2015; Stubbs et al., 2015; Kumar et al., 2015). We use the preprocessing scripts provided by Alsentzer et al. (2019) for the i2b2 tasks.We additionally evaluate the models on the Multi-Genre Natural Language Inference Corpus (MNLI; Williams et al., 2018).

For MedNLI, we use the standard training, validation, and test splits provided by Romanov and

Shivade (2018). For the i2b2 datasets, we split the training data into 90% for training and 10% for validation and use the standard test splits. For MNLI, we use the standard train and validaton splits. We provide statistics for all datasets in Table 1.

Postprocessing We diverge slightly from Alsentzer et al. (2019) in i2b2 prediction postprocessing. All i2b2 tasks are trained using the IOB format and the exact F1 metrics are computed on predicted spans after post-processing. We utilize a simple post-processing scheme to limit predictions to valid sequences and select the prediction with highest probability out of each valid option. For example, given the invalid prediction "O I-problem," we select the tag with the next highest probability, e.g., "O B-problem" or "O O" rather than always selecting "O B-problem."

Finetuning For all tasks, we finetune the models using the same sweep of hyperparameters as Alsentzer et al. (2019): AdamW with a learning rate $\in \{2 \cdot 10^{-5}, 3 \cdot 10^{-5}, 5 \cdot 10^{-5}\}$, a batch size $\in \{16, 32\}$, and number of epochs $\in \{3, 4, 5\}$. We use a sequence length of 150 for all tasks.

Metrics We report accuracy for MedNLI and MNLI and exact F1 for all i2b2 tasks. Exact F1 is computed as the number of exact matches between the predicted and true NER spans.

4.4 Computational Budget

All pretraning and finetuning experiments use a single A100 40GB GPU and 16 threads on an AMD EPYC 7543 CPU. Dataset preprocessing is CPU-intensive: upwards of 24 hours for MIMIC-III (not including downstream tasks). Pretraining on MIMIC-III takes approximately 4-6 hours per model to train the 200k models, plus 24 hours per model for the 900k models.

5 Results

Dataset Statistics As shown by Table 1, there is a trade-off between more information by maintaining casing and more efficiency in terms of shorter sequence lengths. The cased tokenizer requires more tokens than the uncased tokenizer, especially on clinical datasets such as MIMIC-III: While uncased tokenization yields 11.38% fewer tokens and 36.45% fewer sequences with length greater than 128 on MIMIC-III, on C4, uncased tokenization only yields 4.08-4.12% fewer tokens and 2.26% fewer documents with length greater than 128. This demonstrates the impact that improper casing and formatting in clinical notes has compared to generic open-domain datasets.

Note that the model with independent case encoding uses the uncased tokenizer and vocabulary.

Downstream Tasks The results for all downstream tasks are provided in Table 2. On nearly all tasks, an uncased or independently cased model performs best – even on NER, which is traditionally considered a case-sensitive task. The only exception to this rule is i2b2 2006, where BERT_C outperforms BERT_U before training on MIMIC-III.

The standard BERT models are generally worse than the models trained on MIMIC-III, but this is to be expected, as the standard BERT models are not trained on in-domain text. We are more interested in the relative performance of the cased and uncased models. In this case, we see that the uncased models have a more clear advantage once trained on MIMIC-III. For example, on MedNLI, BERT_C has a slightly higher accuracy of 78.692 than BERT_U which has 78.129. This advantage disappears once both models are trained on MIMIC-III, where the accuracy of BERT_C improves to 81.294 and BERT_U to 83.333. The independently cased model, BERT_I, had the second highest score on MedNLI at 82.841. Similar to Alsentzer et al. (2019), we find that the i2b2 2006 de-identification task is the most challenging task for models trained on MIMIC-III, likely due to the difference in formatting. However, out of the models trained on MIMIC-III, the uncased and independently cased model both outperform the cased model.

6 Analysis: Robustness of the Independent Case Encoding

To demonstrate the robustness of the independent case encoding, we construct 2 evaluation scenarios by transforming the evaluation dataset either to all upper-case or all lower-case. These transformations naturally do not affect the uncased model, but have a profound impact on the cased model. The upper part of Table 3 shows the difference in performance with respect to the original results for lower-cased data. We see a small but measurable impact across all tasks and a light increase in performance for the independently cased model on MedNLI and MNLI Mismatched.

However, the results on the upper-cased evaluation data in the bottom part of Table 3 are much more profound. Across all tasks, the cased model

	MedNLI	i2b2 2006	i2b2 2010	i2b2 2014	MNLI MM	MNLI M
BERT$_C$	1.617	19.341	4.077	14.358	0.529	0.632
BERT$_C$ + MIMIC 200k	0.985	14.976	2.128	11.152	0.498	0.143
BERT$_C$ + MIMIC 900k	0.352	13.688	1.696	10.565	0.559	0.000
BERT$_I$ + MIMIC 200k	-0.352	4.113	0.311	1.468	-0.081	0.071
BERT$_I$ + MIMIC 900k	0.141	6.257	0.239	1.592	-0.081	0.224
BERT$_C$	38.115	10.774	37.781	19.899	36.381	37.779
BERT$_C$ + MIMIC 200k	33.896	9.962	23.092	13.454	32.242	32.888
BERT$_C$ + MIMIC 900k	21.519	9.833	16.387	12.484	27.278	27.224
BERT$_I$ + MIMIC 200k	-0.844	1.293	0.591	0.799	0.203	0.245
BERT$_I$ + MIMIC 900k	0.070	2.518	0.228	0.800	-0.122	0.234

Table 3: Difference in performance on the lowercased (top) or uppercased (bottom) development set with respect to the original results. A positive number implies reduced performance. Smaller absolute values mean the model is more robust to the change in case. Uncased models are not shown as they are unaffected.

is impacted by 9.833-38.115 percentage points. In contrast, the independent case model is impacted by less than a percent across all tasks except for i2b2 2006, for which its performance is reduced by 1.29 and 2.52 for the 200k and 900k variants, respectively. Compared to the cased model, this demonstrates a much better balance of retaining case information where possible while ignoring it where not necessary.

7 Conclusion

Our paper provides an analysis of the impact of different casing strategies on clinical notes. We find that uncased models outperform cased models on downstream tasks while requiring fewer tokens. Our findings highlight the trade-off of case-sensitive tokenization, especially as it pertains to clinical data and other text that may not have proper casing. Finally, we introduce independent case encoding to better balance performance on case-sensitive and improperly-cased tasks.

Limitations

Language Models For language models, especially large open-domain language models such as GPT, cased tokenization is, for good reason, the default. This is justifiable because the model needs to produce text that will be read by humans — and humans prefer text with proper casing. There are also applications where preserving case is important, such as in programming languages, where case is semantically significant. Beyond the obvious cased vs. uncased conflicts if case was dropped (e.g., vector `x` vs. matrix `X`), case is regularly used to distinguish between functions, types, and variables and in Go is even used to distinguish between public and private. Conceptually, this makes sense – using text-to-speech for code is hardly as effective for code as text-to-speech is for prose.

Scope We intentionally limit the scope of our work to clinical data and, more specifically, to the datasets used by existing ClinicalBERT models. While all of the tasks we evaluated on showed strong performance of uncased tokenization this is in no way an exhaustive list of tasks. While our findings may transfer to other domains where improper casing negatively effects the performance of cased models, more experimentation would be required in those domains and we did not investigate any such tasks. However we hope this work motivates others to think more carefully about the decision of cased vs. uncased tokenization both in clinical applications and more generally.

References

Emily Alsentzer, John R. Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew B. A. McDermott. 2019. Publicly Available Clinical BERT Embeddings. *Preprint*, arxiv:1904.03323.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A Pretrained Language Model for Scientific Text. *Preprint*, arxiv:1903.10676.

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. *Preprint*, arXiv:2005.14165.

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.

Jonas Geiping and Tom Goldstein. 2022. Cramming: Training a language model on a single gpu in one day. *Preprint*, arXiv:2212.14034.

Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. 2000. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing. *Preprint*, arxiv:2007.15779.

Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. 2020. ClinicalBERT: Modeling Clinical Notes and Predicting Hospital Readmission. *Preprint*, arxiv:1904.05342.

Alistair Johnson, Tom Pollard, and Roger Mark. 2015. MIMIC-III Clinical Database.

Alistair E.W. Johnson, Tom J. Pollard, Lu Shen, Liwei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. 2016. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3(1):160035.

Vishesh Kumar, Amber Stubbs, Stanley Shaw, and Özlem Uzuner. 2015. Creation of a new longitudinal corpus of clinical narratives. *Journal of Biomedical Informatics*, 58:S6–S10. Supplement: Proceedings of the 2014 i2b2/UTHealth Shared-Tasks and Workshop on Challenges in Natural Language Processing for Clinical Data.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics (Oxford, England)*, 36(4):1234–1240.

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*, abs/1907.11692.

Jacob Portes, Alexander R Trott, Sam Havens, DANIEL KING, Abhinav Venigalla, Moin Nadeem, Nikhil Sardana, Daya Khudia, and Jonathan Frankle. 2023. MosaicBERT: A bidirectional encoder optimized for fast pretraining. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pretraining.

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.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Preprint*, arxiv:1910.10683.

Alexey Romanov and Chaitanya Shivade. 2018. Lessons from natural language inference in the clinical domain. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1586–1596, Brussels, Belgium. Association for Computational Linguistics.

Amber Stubbs, Christopher Kotfila, and Özlem Uzuner. 2015. Automated systems for the de-identification of longitudinal clinical narratives: Overview of 2014 i2b2/uthealth shared task track 1. *Journal of Biomedical Informatics*, 58:S11–S19. Supplement: Proceedings of the 2014 i2b2/UTHealth Shared-Tasks and Workshop on Challenges in Natural Language Processing for Clinical Data.

Amber Stubbs and Özlem Uzuner. 2015. Annotating longitudinal clinical narratives for de-identification: The 2014 i2b2/uthealth corpus. *Journal of Biomedical Informatics*, 58:S20–S29. Supplement: Proceedings of the 2014 i2b2/UTHealth Shared-Tasks and Workshop on Challenges in Natural Language Processing for Clinical Data.

Özlem Uzuner, Yuan Luo, and Peter Szolovits. 2007. Evaluating the State-of-the-Art in Automatic De-identification. *Journal of the American Medical Informatics Association*, 14(5):550–563.

Özlem Uzuner, Brett R South, Shuying Shen, and Scott L DuVall. 2011. 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, 18(5):552–556.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *Preprint*, arxiv:1706.03762.

Guangyu Wang, Xiaohong Liu, Zhen Ying, Guoxing Yang, Zhiwei Chen, Zhiwen Liu, Min Zhang, Hongmei Yan, Yuxing Lu, Yuanxu Gao, Kanmin Xue, Xiaoying Li, and Ying Chen. 2023. Optimized glycemic control of type 2 diabetes with reinforcement learning: a proof-of-concept trial.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.