

Analiza rozwiązań

w zakresie anonimizacji danych
i generowania danych syntetycznych



Publikacja została opracowana w ramach projektu „Anonimizacja danych i generowanie danych syntetycznych” realizowanego przez Centrum Nowych Technologii dla Polityk Publicznych NASK-PIB pod nadzorem Kancelarii Prezesa Rady Ministrów.

Warszawa 2022

SPIS TREŚCI

Wstęp	5
Część I: Techniki depersonalizacji danych	9
Anonimizacja	10
Dodawanie zakłóceń	12
Permutacja	14
k-anonimizacja	15
l-dywersyfikacja	18
t-bliskość	20
Prywatność różnicowa	21
Dane nieustrukturyzowane	24
Pseudonimizacja	27
Szyfrowanie z kluczem tajnym	29
Funkcja skrótu	30
Funkcja skrótu z kluczem	30
Szyfrowanie deterministyczne	30
Tokenizacja	31
Dane syntetyczne	32
Część II: Generowanie danych syntetycznych	35
Generowanie języka naturalnego	36
Metody generowania danych tekstowych	38
Łańcuchy Markowa	39
Rekurencyjna sieć neuronowa (RNN)	40
Sieci LSTM	41
Transformery	42
GPT	44
BERT	47
XLNet	48
T5	50
BART	51
Generowanie danych tekstowych – wybrane zagadnienia	52
Generowanie data-to-text	52
Generowanie dialogów	53
Generowanie pytań	55
Generowanie odpowiedzi	55
Parafrazowanie	56
Generowanie historii	56

Generowanie obrazów	57
Metody generowania danych obrazowych	58
Autoenkodery wariacyjne (VAE)	59
Generatywne sieci przeciwstawne (GAN)	60
CycleGAN	62
StyleGAN	64
Transformery	66
Image GPT (iGPT)	66
DALL-E	67
Modele dyfuzji	69
DALL-E 2	70
Imagen	73
Generowanie danych obrazowych – wybrane zagadnienia	76
Generowanie text-to-image	76
Generowanie layout-to-image	77
Inpainting	78
Tłumaczenie image-to-image	78
Generowanie obrazów medycznych	79
Generowanie danych tabelarycznych	83
Metody generowania danych tabelarycznych	83
Metoda Monte Carlo	84
MCMC i próbkowanie Gibbsa	84
Sieci bayesowskie	85
Generatywne sieci przeciwstawne (GAN)	86
table-GAN	87
CTGAN	88
CTAB-GAN	89
TimeGAN	90
Wybrane narzędzia do generowania danych syntetycznych	91
pydbgen	91
Faker	92
Mimesis	92
Mesa	92
DataSynthesizer	93
Synthetic Data Vault	94
Uwagi końcowe	95
Glosariusz	98
Literatura	104

```
elif _operation == "MIRROR_Y":
    mirror_mod.use_x = False
    mirror_mod.use_y = True
    mirror_mod.use_z = False
elif _operation == "MIRROR_Z":
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True

#selection at the end -add back the deselected mirror
mirror_ob.select= 1
modifier_ob.select=1
bpy.context.scene.objects.active = modifier_ob
print("Selected" + str(modifier_ob)) # modifier ob is t
    #mirror_ob.select = 0
done = bpy.context.selected_objects[0]
bpy.data.objects[done.name].select = 1
```

Wstęp

Dane i dostęp do nich to obecnie jedno z wąskich gardel innowacji. Gromadzenie zbiorów danych o odpowiedniej jakości jest czasochłonne, lecz niezbędne m.in. do rozwoju systemów opartych na sztucznej inteligencji (*artificial intelligence*, AI). Postęp w zakresie AI to niezwykle pożądane zjawisko, gdyż w coraz większym stopniu właśnie ta technologia odpowiada za usprawnianie procesów, podnoszenie jakości czy zmniejszanie kosztów. Dotyczy to niemalże każdej branży.

Wykorzystywanie danych w analizach oraz w opracowywaniu algorytmów niesie jednak za sobą ryzyko naruszenia ochrony prywatności. Zapewnienie zgodności z wymaganiami prawnymi w zakresie ich przetwarzania może być realizowane poprzez m.in. anonimizację danych rzeczywistych lub tworzenie danych syntetycznych. Należy również wspomnieć o potrzebach związanych z bezpiecznym przepływem, przetwarzaniem oraz przechowywaniem danych, co jest niezbędne zarówno do budowania zaufania, jak i do zwiększenia poziomu satysfakcji klientów i interesantów. Coraz częściej podejmowany jest temat ochrony danych osobowych w kontekście rozwiązań AI i regulacji prawnych w tym obszarze. Świadczą o tym m.in. fragmenty podsumowania zorganizowanej przez Urząd Ochrony Danych Osobowych konferencji Sztuczna inteligencja w kontekście ochrony danych osobowych:

[...] System ochrony danych osobowych [...] nakłada na administratorów względny zakaz zautomatyzowanego podejmowania decyzji w (art. 22 RODO), obowiązek uwzględniania ochrony danych w fazie projektowania (data protection by design – art. 25 ust. 1 RODO) oraz domyślnej ochrony danych (data protection by default – art. 25 ust. 2 RODO), a także konieczność przeprowadzania oceny skutków przetwarzania dla ochrony danych (data protection impact assessment – art. 35 RODO) [...] Rozporządzenie o ochronie danych służy celom ochronnym jakim jest zapewnienie ochrony praw i wolności podmiotów danych w związku z przetwarzaniem ich danych osobowych, przy uwzględnieniu jednak zasad ochrony danych uregulowanych w art. 5 RODO zwłaszcza rzetelności i przejrzystości, minimalizacji danych oraz mechanizmu risk-based approach. Istotą data protection by design w rozumieniu art. 25 RODO jest obowiązek administratora uwzględnienia ochrony danych osobowych już w fazie projektowania określonego rozwiązania, usługi czy systemu sztucznej inteligencji. Ma to zagwarantować m.in., że ochrona danych osobowych stanie się immanentnym elementem każdego projektu już na etapie tworzenia [...] Wymóg uwzględniania ochrony danych w fazie projektowania, pozwala na zapewnienie

zgodności nie tylko z art. 25 RODO, ale także z pozostałymi wymogami w oparciu o mechanizm analizy ryzyka dla praw i wolności podmiotów danych, który implikuje konieczność rozpoczęcia analizy od weryfikacji zasad przetwarzania, legalizacji, możliwości realizacji uprawnień podmiotów danych, a skończywszy na bezpieczeństwie przetwarzania [...]¹.

Poprawne funkcjonowanie systemów AI zależy od jakości danych, które są wykorzystywane do budowania takich rozwiązań. Utworzenie repozytorium danych do pracy nad sztuczną inteligencją może okazać się kluczowe i wpłynąć w sposób bezpośredni na rozwój tej dziedziny w Polsce. Spektrum zastosowań anonimizacji oraz generowania danych syntetycznych w nauce, administracji państwowej i gospodarce jest bardzo szerokie, a postępująca cyfryzacja i automatyzacja tworzą niemal każdego dnia nowe możliwości.

Głównym celem niniejszej publikacji jest analiza metod anonimizacji i generowania danych syntetycznych (tekstowych, obrazowych i tabelarycznych), aby ukazać możliwości stosowania tych metod, ich charakterystykę oraz przedstawić rekomendacje dla projektu "Anonimizacja danych i generowanie danych syntetycznych". W publikacji zostały opisane metody maskowania i depersonalizacji danych, które służą do zapewnienia ochrony praw zarówno osób fizycznych, jak i przedsiębiorstw sektora publicznego i prywatnego. W związku z wszechobecnym procesem przetwarzania danych osobowych, wymagane jest przygotowanie szczegółowego przeglądu dostępnych metod i dobrych praktyk w tym zakresie. Analiza służy przede wszystkim weryfikacji możliwości zastosowania ww. technik, ale również kompensacji wiedzy dla zagadnień dotyczących ochrony danych poufnych, w tym minimalizacji ryzyka naruszenia prywatności i zapewnienia bezpieczeństwa procesu przetwarzania danych.

Publikacja została podzielona na dwie główne części. Pierwsza z nich to *Techniki depersonalizacji danych*, a omówiono w niej metody anonimizacji i pseudonimizacji danych, w szczególności te wymienione przez ekspertów z Grupy Roboczej Artykułu 29 w Opinii 05/2014 w sprawie technik anonimizacji. Dodatkowo rozpatrzono problem anonimizacji danych nieustrukturyzowanych oraz przybliżono zagadnienie wykorzystania danych syntetycznych jako alternatywnego podejścia do zapewnienia ochrony danych. Część druga to

1 Rapcewicz, A., Lubasz, D., Gawroński, M., Grzesiuk, M., Lewandowski, M. (2021). *SZTUCZNA INTELIGENCJA w kontekście ochrony danych osobowych. Materiały pokonferencyjne*. <https://uodo.gov.pl/pl/file/3775>

Generowanie danych syntetycznych. Została ona poświęcona metodom generowania danych, z podziałem na dane tekstowe, obrazowe i tabelaryczne. Rozpatrzono dużą liczbę technik, począwszy od najprostszych, takich jak łańcuchy Markowa i metoda Monte Carlo, aż do popularnych obecnie technik uczenia głębokiego, m.in. struktury Transformer, sieci GAN i modeli dyfuzji, wyznaczających trendy i kierunki dalszego rozwoju w dziedzinie AI.



Część I: Techniki depersonalizacji danych

Anonimizacja

Budowa i utrzymanie przewagi konkurencyjnej w świecie opartym na danych wiąże się z wieloma wyzwaniami, a jednym z najważniejszych jest przestrzeganie zasad rozporządzenia o ochronie danych osobowych (RODO). Wraz z nałożonym obowiązkiem zapewnienia bezpieczeństwa danych, możliwość zastosowania danych rzeczywistych do celów naukowych, produkcyjnych lub biznesowych została istotnie ograniczona. Aby wykorzystać lub udostępnić informacje i materiały do analiz, badań, budowania nowych produktów, usług lub aplikacji, należy zminimalizować ryzyko naruszenia ochrony danych osobowych. W tym celu większość podmiotów publicznych i prywatnych stosuje rygorystyczne kontrole bezpieczeństwa i sięga po różne metody maskowania i depersonalizacji danych, aby zapewnić ochronę podstawowych praw i wolności osób fizycznych w związku z przetwarzaniem danych osobowych.



Rysunek 1: Przykłady danych osobowych

Techniki anonimizacji służą do tworzenia zbiorów, które wyglądają strukturalnie podobnie do oryginalnych, ale ukrywają poufne informacje. Ich celem jest ograniczenie ryzyka naruszenia prywatności i zapewnienie gwarancji, że dane będą przetwarzane jedynie wtedy, gdy będą przestrzegane prawa wszystkich

osób, których dane dotyczą². Co więcej, maskowanie danych powinno być nieodwracalne, tzn. atakujący nie powinien zdołać odtworzyć prawdziwych informacji ani na podstawie samego zbioru, ani w połączeniu z innymi źródłami zewnętrznymi.

Utworzenie anonimowego zbioru danych przy jednoczesnym zachowaniu odpowiedniej ilości informacji niezbędnych na potrzeby wykonania zadania nie jest łatwe do zrealizowania. Zakres informacji, który powinien zostać zamaskowany jest bardzo mocno uzależniony nie tylko od ram prawnych w zakresie ochrony danych, ale również od wymagań niezbędnych przy realizacji zadania z użyciem danych anonimowych. Stosowane techniki i metody mogą być dodatkowo uwarunkowane wymaganiami przyjętymi wewnątrz organizacji. Przy wyborze odpowiedniej techniki należy uwzględnić jej ograniczenia, w szczególności możliwość potencjalnej identyfikacji osób fizycznych w zbiorze danych, a także tworzenia powiązań i wnioskowania, np. na podstawie wiedzy podstawowej (*background knowledge*). Gdy zbiór jest zanonimizowany i nie ma już możliwości zidentyfikowania żadnej osoby fizycznej, prawo o ochronie danych nie ma dalej zastosowania.

Przykładami technik anonimizacji danych są:

- zaciemnianie lub usuwanie danych (*nulling out, deleting*);
- zastępowanie danych osobowych i nazw wartościami pustymi, takimi jak XXXX, #### lub 9999;
- manipulacje numeryczne (np. zastępowanie wartości podlegających anonimizacji wartością średnią; zmiany w wariancji wartości danych; dodawanie zakłóceń itp.);
- zamiana danych (rzeczywiste wartości pozostają, ale są przypisane do różnych elementów, przez co niemożliwe jest połączenie wartości z właściwą osobą fizyczną);
- techniki randomizacji, np. dodawanie zakłóceń, permutacja, prywatność różnicowa;

2 Grupa Robocza Artykułu 29 (przyjęta w dniu 10 kwietnia 2014 r.). *Opinia 05/2014 w sprawie technik anonimizacji*. https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2014/wp216_pl.pdf

- techniki uogólnienia (generalizacji), takie jak k -anonimizacja, l -dywersyfikacja, t -bliskość.

Dodawanie zakłóceń

Jedną z najpopularniejszych technik anonimizacji jest dodawanie zakłóceń, inaczej perturbacja danych. Polega ona na nieznaczącej modyfikacji oryginalnych danych, tak aby stały się one mniej dokładne, przy jednoczesnym zachowaniu tych samych właściwości statystycznych³. Najczęściej stosowaną metodą jest przekształcenie danych poprzez dodanie szumu o znanym rozkładzie statystycznym. Dzięki temu oryginalny rozkład danych będzie mógł zostać zrekonstruowany, ale indywidualne (oryginalne) wartości będą niemożliwe do odtworzenia.

Używając tej techniki, należy wziąć pod uwagę przede wszystkim typ danych, a także zamierzony cel ich wykorzystania. Równie istotne są właściwości statystyczne zbioru danych przed i po dodaniu szumu, tj. rozkład danych, w szczególności średnia, wariancja, odchylenie standardowe, kowariancja i korelacja, zarówno dla oryginalnych, jak i zaburzonych danych. Z tego powodu dodawanie zakłóceń w postaci szumu najlepiej sprawdzi się do ochrony danych typu ciągłego. Wśród podstawowych typów zakłóceń można wyróżnić:

- szum addytywny (*additive noise*)

$$Z = X + \epsilon,$$

- szum multiplikatywny (*multiplicative noise*)

$$Z = X \cdot \epsilon,$$

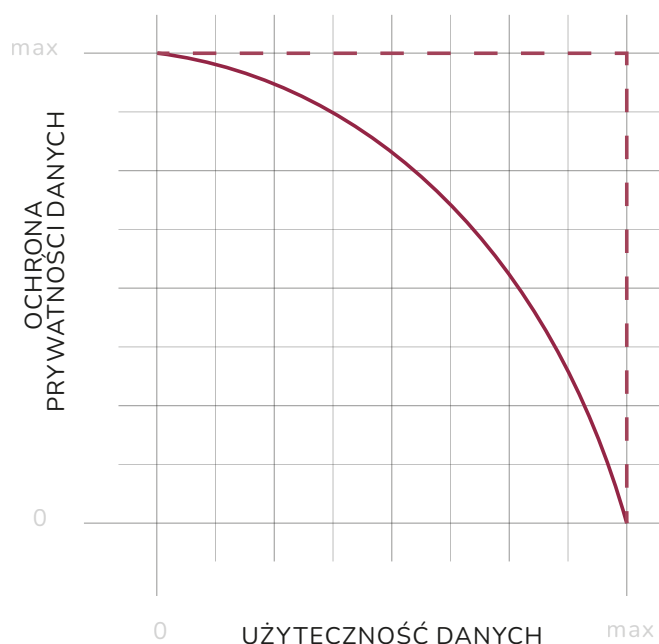
- szum logarytmiczny (*logarithmic multiplicative noise*)

$$Z = \ln X + \epsilon,$$

gdzie X oznacza zbiór oryginalny, Z oznacza zbiór zaburzony, natomiast ϵ jest zmienną ciągłą o standardowym rozkładzie normalnym $N(0,1)$.

3 Tamże.

Dodawanie zakłóceń w postaci szumu do pewnego stopnia może zapewnić ochronę danych, jednak przybliżone dopasowanie interwałów, w których zawierają się dane, może być wciąż osiągalne. Wielkość dodanego szumu powinna zależeć nie tylko od zakresu wartości pierwotnych zmiennych, ale również od wymaganego poziomu ochrony, pożądanego zakresu wartości po anonimizacji oraz dopuszczalnego poziomu ryzyka ujawnienia i utraty informacji. Określenie akceptowalnego poziomu anonimowości nie jest rzeczą trywialną: wartość zbyt mała może skutkować niewystarczającą ochroną, natomiast zbyt wysoka wiąże się z ryzykiem, że dane będą bezużyteczne.



Rysunek 2: Kompromis między prywatnością a użytecznością danych

Odnosnie do części danych ważne może być również sprawdzenie, czy zaburzone wartości mieszczą się w odpowiednim zakresie i zwrócenie szczególnej uwagi na wartości odstające. Jeżeli oryginalny zbiór danych zawiera wartości odstające, należy uwzględnić to, że dodanie zakłóceń w postaci szumu nie będzie tych wartości chronić. Zaburzone wartości odstające nadal będzie można wykrywać, a zatem ryzyko ich reidentyfikacji jest wysokie. Przykładowo w zbiorze danych rejestrującym wysokość wynagrodzeń w określonym regionie występuje pojedynczy, bardzo wysoki zarobek. Po dodaniu szumu wartość ta nadal będzie reprezentować najwyższe wynagrodzenie w tym regionie, więc łatwo będzie ją wykorzystać do ponownej identyfikacji. Stosując wyższy poziom szumu, ryzyko ponownej identyfikacji wartości odstających maleje, ale utrata pozostałych informacji rośnie, ponieważ oryginalna struktura danych nie zostanie zachowana, co również jest sytuacją niepożądaną. Rozwiązaniem

tego problemu jest dodanie do wartości odstających wyższego niż do pozostałych, poziomu szumu⁴.

Tworzenie zaburzonych danych, które są statystycznie bliskie danym rzeczywistym, przy zachowaniu kompromisu między użytecznością a prywatnością, nadal jest wyzwaniem. Dlatego dodawanie zakłóceń nie stanowi samodzielnego rozwiązania problemu anonimizacji danych, a jest jedynie środkiem uzupełniającym dla innych technik.

Permutacja

Permutacja, czyli zamiana danych, polega na reorganizacji (tasowaniu) wartości atrybutów danych, tak aby pozostały obecne w zbiorze, ale nie odpowiadały ich oryginalnym rekordom. Metodę permutacji można uznać za szczególną formę dodawania zakłóceń. Jednak zastosowanie permutacji, w przeciwieństwie do dodawania szumu, pozwala na zachowanie dokładnych wartości atrybutów w zbiorze danych⁵. Losowe tasowanie danych nie zmienia zakresów i rozkładów wartości atrybutów, ale sprawi, że ponowna identyfikacja oryginalnych danych będzie utrudniona. Ponieważ permutacja może wprowadzić odmienne korelacje między wartościami i poszczególnymi identyfikatorami, jej zastosowanie jest zasadne, gdy nie ma potrzeby zachowania relacji między informacjami zawartymi w poszczególnych rekordach: jeśli między atrybutami występują związki logiczne bądź korelacje, to w wyniku permutacji danych taki związek zostanie zlikwidowany. Aby zachować istotne do dalszej analizy zależności, podczas permutacji oryginalny zbiór danych dzieli się na grupy lub partycje, czyli podzbiory atrybutów, których wartości są powiązane silnym związkiem logicznym bądź korelacją. Podczas korzystania z grup (złożonych z jednej lub wielu kolumn) wartości atrybutów są tasowane w taki sposób, aby nie odpowiadały oryginalnym rekordom, ale wartości z tego samego wiersza są zawsze ze sobą kojarzone. Z kolei w przypadku korzystania z partycji dane są tasowane wewnątrz niej. Oznacza to, że proces tasowania dotyczy wierszy o tej samej wartości atrybutu, ze względu na którą została utworzona partycja, natomiast wartości z różnych partycji nigdy nie są ze sobą kojarzone.

4 International Household Survey Network. (2019). *Anonymization Methods*. SDC Practice Guide. Pobrano 28.06.2022 z https://sdcppractice.readthedocs.io/en/latest/anon_methods.html#anonymization-methods

5 Grupa Robocza Artykułu 29 (przyjęta w dniu 10 kwietnia 2014r.), dz. cyt.

Zamiana danych zapewnia wysoki poziom użyteczności danych, jednak, podobnie jak w przypadku perturbacji danych, nie stanowi samodzielnego rozwiązania problemu anonimizacji i zawsze powinna być połączona z innymi technikami, takimi jak maskowanie lub usuwanie oczywistych atrybutów łączyjących dane z osobą fizyczną.

k-anonimizacja

Pojęcie k -anonimowości zostało wprowadzone jako próba rozwiązania problemu identyfikacji osób fizycznych podczas dystrybuowania zbiorów danych standardowo przechowywanych w formie tabel, gdzie każdy rekord odpowiada jednej osobie fizycznej. k -anonimizacja gwarantuje bezpieczeństwo w taki sposób, że nie jest możliwe powiązanie rekordu z konkretną osobą fizyczną. Wykonuje się to poprzez grupowanie rekordu z co najmniej $k-1$ innymi rekordami (odpowiadającymi innym osobom fizycznym), a następnie uogólnianiu wartości atrybutów tak, aby dla k osób w obrębie utworzonej grupy przypisana była ta sama wartość. Definicja wprowadzona przez Sweeney (2002) brzmi:

Definicja (warunek k -anonimizacji): Tabela (zbiór danych) T spełnia warunek k -anonimizacji, jeżeli dla każdego rekordu (wiersza) $t \in T$ istnieje $k-1$ innych rekordów (wierszy) $t_{i1}, t_{i2}, \dots, t_{i(k-1)} \in T$ takich, że

$$t[C] = t_{i1}[C] = t_{i2}[C] = \dots = t_{i(k-1)}[C]$$

dla każdego $C \in Q$, gdzie Q jest zbiorem quasi-identyfikatorów⁶.

Zgodnie z definicją zbiór danych jest k -anonimowy wtedy, gdy każdy pojedynczy rekord jest nie do odróżnienia od co najmniej $k-1$ innych rekordów pod względem atrybutów określanych jako quasi-identyfikatory, czyli identyfikatorów pośrednich. Quasi-identyfikatory to atrybuty, które indywidualnie lub w połączeniu z innymi atrybutami, źródłami zewnętrznymi czy wiedzą podstawową mogą posłużyć do jednoznacznej identyfikacji osób fizycznych. Są to np. data urodzenia, kod pocztowy i płeć. Celem k -anonimowości jest zapewnienie możliwości udostępnienia danych w taki sposób, aby szansa na

6 Sweeney, L. (2002). *Achieving k -anonymity privacy protection using generalization and suppression*. <https://dataprivacylab.org/dataprivacy/projects/kanonymity/kanonymity2.pdf>

zidentyfikowanie osoby i łączenie jej z innymi informacjami przy użyciu quasi-identyfikatorów była ograniczona. W przypadku k -anonimizowanego zbioru danych nie można zidentyfikować tożsamości pojedynczego rekordu, ponieważ w zbiorze istnieje jeszcze $k-1$ innych, podobnych rekordów. Podzbiór k rekordów nazywany jest klasą równoważności.

W modelu k -anonimizacji wartość k można traktować jako miarę prywatności: im wyższa wartość k , tym wyższa ochrona, czyli tym trudniej o odwrócenie procesu anonimizacji. W danej klasie równoważności prawdopodobieństwo identyfikacji osoby fizycznej jest równe lub mniejsze niż $1/k$.

Aby osiągnąć k -anonimowość, stosuje się różne algorytmy, ale zdecydowana większość opiera się na technikach uogólniania i tzw. supresji danych (*data suppression*). Konceptyjna prostota sprawia, że k -anonimowość jest szeroko dyskutowaną i stosowaną strategią anonimizacji danych. Jednak nie jest strategią bez wad. Jedną z nich wiąże się z częstym założeniem, że każdą osobę reprezentuje jeden i tylko jeden rekord w zbiorze danych. Jeśli tak nie jest, klasa równoważności składająca się z k rekordów niekoniecznie będzie odpowiadać k różnym osobom fizycznym. Kolejny problem dotyczy tego, że przy generowaniu k -anonimizowanego zbioru danych nie bierze się pod uwagę wartości atrybutów. Może to prowadzić do tworzenia klas równoważności, w których wartości niektórych atrybutów są równe dla wszystkich k rekordów, a w konsekwencji do braku ochrony prywatnych informacji dowolnej osoby należącej do takiej grupy. Innym problemem, często nieroztropnie pomijanym podczas tworzenia klas równoważności, jest nieuwzględnianie możliwości powiązania quasi-identyfikatorów z wiedzą podstawową na temat atrybutu. Problem ujawniania atrybutów można rozwiązać poprzez zwiększenie różnorodności wrażliwych wartości w ramach klas równoważności, co jest podejściem przyjętym w modelu l -dywersyfikacji⁷, omówionym w kolejnym punkcie.

7 Mendes, R., Vilela, J. (2017). *Privacy-Preserving Data Mining: Methods, Metrics and Applications*. <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7950921>

Chociaż k -anonimizacja jest podatna na niektóre ataki, to umożliwia publikację danych z zachowaniem ich wysokiej użyteczności, czego nie można powiedzieć o bardziej bezpiecznych modelach (np. prywatności różnicowej), które w celu zapewnienia rygorystycznej ochrony danych ostatecznie doprowadzają do rażącego obniżenia ich użyteczności. Dodatkowo, w porównaniu z innymi metodami, k -anonimowość jest łatwa do wdrożenia, a koszt jej stosowania jest stosunkowo niski. Dlatego jest atrakcyjną metodą z praktycznego punktu widzenia, łatwą do zastosowania, a jednocześnie niosącą ze sobą wystarczającą gwarancję ochrony prywatności.

	Imię	Stan cywilny	Wiek	Kod pocztowy	Przestępstwo
1	Piotr	rozwiedziony	29	32-042	zabójstwo
2	Anna	panna	20	32-021	kradzież
3	Maria	wdowa	24	32-024	znęcanie się
4	Jan	rozwiedziony	28	32-046	pobicie
5	Tomasz	wdowiec	25	32-045	korupcja
6	Barbara	panna	23	32-027	zniestawienie

Tabela 1: Przykład zastosowania techniki k -anonimizacji ($k=3$): tabela przed anonimizacją⁸

		Stan cywilny	Wiek	Kod pocztowy	Przestępstwo
1		niezameżny	[25-30]	32-04*	zabójstwo
2	1	niezameżny	[25-30]	32-04*	pobicie
3		niezameżny	[25-30]	32-04*	korupcja
4		niezameżny	[20-25]	32-02*	kradzież
5	2	niezameżny	[20-25]	32-02*	znęcanie się
6		niezameżny	[20-25]	32-02*	zniestawienie

Tabela 2: Przykład zastosowania techniki k -anonimizacji ($k=3$): tabela po anonimizacji. Każdy rekord jest nie do odróżnienia od co najmniej dwóch innych rekordów pod względem quasi-identyfikatorów (stan cywilny, wiek, kod pocztowy)⁹

⁸ Ayala-Rivera, V., McDonagh, P., Cerqueus, T., Murphy, L. (2014). A Systematic Comparison and Evaluation of k -Anonymization Algorithms for Practitioners. <http://www.tdp.cat/issues11/tdp.a169a14.pdf>

⁹ Tamże,

l-dywersyfikacja

Ponieważ *k*-anonimizacja jest podatna na pewne ataki, którym można zapobiec, opracowano jej modyfikację nazwaną *l*-dywersyfikacją. Jest to technika, która rozwiązuje problem związany z możliwością wnioskowania ze zbioru danych na podstawie źródeł zewnętrznych i wiedzy podstawowej, którą może posiadać atakujący¹⁰. Machanavajhala i in. (2007) przedstawili następującą definicję *l*-dywersyfikacji:

Definicja (warunek *l*-dywersyfikacji): *Mówi się, że klasa równoważności spełnia warunek *l*-dywersyfikacji, jeśli dla każdego wrażliwego atrybutu klasa równoważności w tabeli ma co najmniej *l* dobrze reprezentowanych wartości. Mówi się, że tabela spełnia warunek *l*-dywersyfikacji, jeśli każda klasa równoważności tabeli spełnia warunek *l*-dywersyfikacji¹¹.*

Jeśli w klasie równoważności, dla których quasi-identyfikator został uogólniony, znajduje się *l* dobrze reprezentowanych wartości, to atakujący będzie musiał wyeliminować *l*-1 możliwych wartości, aby zidentyfikować tożsamość osoby fizycznej, której dane dotyczą. Poprzez wartość *l* administrator danych może określić stopień ochrony prywatności w stosunku do wiedzy podstawowej (atakujący musiałby znać *l*-1 faktów podstawowych, żeby rozpoznać osobę fizyczną)¹².

Machanavajhala i in. (2007) zaproponowali szereg interpretacji sformułowania "dobrze reprezentowanych" wartości. Najprostsza polega na upewnieniu się, że w każdej klasie równoważności istnieje co najmniej *l* różnych wartości dla wrażliwego atrybutu. Jednak takie podejście może nie zapobiegać atakom opartym na wnioskowaniu probabilistycznym, ponieważ w klasie równoważności jedna wartość może pojawiać się znacznie częściej od pozostałych, umożliwiając atakującemu stwierdzenie, że jednostka w tej klasie z dużym prawdopodobieństwem przyjmuje tę wartość. Dlatego opracowano silniejsze

10 Patel, T., Amin, K. (2019). A Study on *k*-anonymity, *l*-diversity, and *t*-closeness Techniques of Privacy Preservation Data Publishing. <http://www.ijirst.org/articles/IJIRSTV6I6015.pdf>

11 Machanavajhala, A., Kifer, D., Gehrke, J., Venkatasubramanian, M. (2007). *l*-Diversity: Privacy beyond kanonymity. https://personal.utdallas.edu/~mxk055100/courses/privacy08f_files/ldiversity.pdf

12 Gehrke, J., Kifer, D., Machanavajhala, A. (2011). *l*-Diversity. Springer. Pobrano 05.07.2022 z https://link.springer.com/referenceworkentry/10.1007/978-1-4419-5906-5_899

pojęcia *l*-dywersyfikacji: *l*-dywersyfikację z entropią (*entropy l-diversity*) i rekursywną (*c, l*)-dywersyfikację (*recursive (c,l)-diversity*). W *l*-dywersyfikacji z entropią w każdej klasie równoważności wartości atrybutów wrażliwych muszą mieć nie tylko różne wartości, ale muszą być wystarczająco równomiernie rozłożone. Aby spełnić warunek *l*-dywersyfikacji z entropią dla każdej klasy równoważności, entropia całej tabeli musi wynosić co najmniej $\log(l)$. Natomiast rekursywna (*c, l*)-dywersyfikacja zapewnia, że najczęstsza wartość wrażliwego atrybutu nie będzie pojawiała się zbyt często, a rzadsze wartości nie będą pojawiały się zbyt rzadko¹³.

	Kod pocztowy	Wiek	Wynagrodzenie	Choroba
1	47-677	29	3K	wrzody żołądka
2	47-602	22	4K	nieżyt żołądka
3	47-678	27	5K	rak żołądka
4	47-905	43	6K	nieżyt żołądka
5	47-909	52	11K	grypa
6	47-906	47	8K	zapalenie oskrzeli
7	47-605	30	7K	zapalenie oskrzeli
8	47-673	36	9K	zapalenie płuc
9	47-607	32	10K	rak żołądka

Tabela 3: Przykład zastosowania techniki *l*-dywersyfikacji ($l = 3$): tabela przed anonimizacją¹⁴

¹³ Machanavajhala, A., Kifer, D., Gehrke, J., Venkatasubramanian, M. (2007). *ℓ-Diversity: Privacy ...*, dz. cyt.

¹⁴ Li, N., Li, T., Venkatasubramanian, S. (2007). *t-Closeness: Privacy Beyond k-Anonymity and l-Diversity*. https://www.cs.purdue.edu/homes/ninghui/papers/t_closeness_icde07.pdf

	Kod pocztowy	Wiek	Wynagrodzenie	Choroba
1	47-6***	2*	3K	wrzody żołądka
2	47-6***	2*	4K	nieżyt żołądka
3	47-6***	2*	5K	rak żołądka
4	47-90*	≥40	6K	nieżyt żołądka
5	47-90*	≥40	11K	grypa
6	47-90*	≥40	8K	zapalenie oskrzeli
7	47-6***	3*	7K	zapalenie oskrzeli
8	47-6***	3*	9K	zapalenie płuc
9	47-6***	3*	10K	rak żołądka

Tabela 4: Przykład zastosowania techniki l -dywersyfikacji ($l = 3$): tabela po anonimizacji ¹⁵

t-bliskość

W literaturze wskazuje się dwa rodzaje ujawnienia informacji: ujawnienie tożsamości i ujawnienie atrybutów. O ujawnieniu tożsamości mówimy wtedy, gdy konkretny rekord (wiersz) zawarty w zbiorze danych (tabeli) można powiązać z konkretną osobą fizyczną. Ujawnienie atrybutu ma miejsce, gdy ujawniane są nowe informacje o osobach, które pozwalają na dokładniejsze wnioskowanie o cechach danej osoby, niż byłoby to możliwe przed udostępnieniem danych. Co więcej, nawet ujawnienie fałszywych informacji o atrybutach może zaszkodzić jednostce, ponieważ może doprowadzić atakującego do błędnych, często szkodliwych wniosków o danej osobie¹⁶. Metoda k -anonimizacji chroni przed ujawnieniem tożsamości, ale nie jest wystarczająca, aby zapobiec ujawnieniu atrybutów. Chociaż metoda l -dywersyfikacji częściowo rozwiązuje ten problem, ma inne ograniczenia i w rezultacie jest, podobnie jak metoda k -anonimizacji, niewystarczająca, aby zapobiec ujawnieniu atrybutów. Dzieje się tak zwłaszcza wtedy, gdy rozkład jest skośny oraz gdy w klasie równoważności wrażliwe wartości atrybutów są różne, ale semantycznie podobne. Atakujący może uzyskać informacje o wrażliwym atrybucie, o ile posiada informacje o ogólnym rozkładzie wartości tego atrybutu. Rozkłady o tym samym

¹⁵ Tamże.

¹⁶ Tamże.

poziomie zróżnicowania mogą zapewniać różne poziomy prywatności, ponieważ istnieją semantyczne relacje między wartościami atrybutów (tzn. istnieje możliwość tworzenia powiązań), a na prywatność wpływa również relacja z rozkładem ogólnym¹⁷.

W celu rozwiązania tych ograniczeń Li i in. (2007) zaproponowali model prywatności nazwany *t*-bliskością, który wymaga, aby rozkład wrażliwego atrybutu, w dowolnej klasie równoważności, był zbliżony do rozkładu atrybutu w całej tabeli:

Definicja (warunek *t*-bliskości): *Mówi się, że klasa równoważności spełnia warunek *t*-bliskości, jeśli odległość między rozkładem wrażliwego atrybutu w tej klasie, a rozkładem atrybutu w całej tabeli (zbiorze danych) wynosi nie więcej niż *t*. Mówi się, że tabela spełnia warunek *t*-bliskości, jeżeli wszystkie klasy równoważności spełniają warunek *t*-bliskości¹⁸.*

Model *t*-bliskości zapobiega ujawnieniu atrybutów, chroni przed atakami opartymi na wnioskowaniu i wiedzy podstawowej, dzięki czemu rozwiązuje ograniczenia zarówno *k*-anonimowości, jak i *l*-dywersyfikacji. Niemniej *t*-bliskość nie jest modelem pozbawionym wad. Warunek, aby rozkład wartości wrażliwych w każdej klasie równoważności był zbliżony do odpowiedniego rozkładu w całej tabeli, wpływa na użyteczność danych, ponieważ ogranicza informacje o korelacji między quasi-identyfikatorami a atrybutami wrażliwymi. Jednak ograniczenie właśnie tych informacji wpływa na ochronę prywatności, ponieważ zbyt wyraźna korelacja między atrybutami może prowadzić do ujawnienia atrybutów. Kompromis między użytecznością danych a prywatnością osiąga się przez wybór wartości parametru *t*¹⁹.

Prywatność różnicowa

Pojęcie zostało opracowane w kontekście kontroli udostępniania danych statystycznych, gdzie pożądane jest udostępnianie ogólnych informacji przy zachowaniu prywatności dla szczegółów budujących zbiór danych. Prywatność różnicowa to formalna, matematyczna definicja prywatności. Nieformalnie

17 Patel, T., Amin, K. (2019). *A Study on k-anonymity ...*, dz. cyt.

18 Li, N., Li, T., Venkatasubramanian, S. (2007). *t-Closeness ...*, dz. cyt.

19 Tamże.

prywatność różnicowa gwarantuje każdej osobie, która dostarcza dane do analizy, że wynik analizy (z zachowaniem prywatności różnicowej) będzie mniej więcej taki sam, niezależnie od tego, czy ta osoba dostarczy swoje dane czy też nie. Zgodnie z definicją zaproponowaną przez Dwork (2006):

Definicja: Funkcja randomizująca K spełnia warunek prywatności różnicowej, jeżeli dla wszystkich zbiorów danych D i D' , różniących się co najwyżej jednym elementem i dla każdego S należącego do zakresu funkcji randomizującej K :

$$\ln \left[\frac{(P(K(D) \in S))}{(P(K(D') \in S))} \right] \leq \epsilon. ^{20}$$

Zbiory danych to zestaw wierszy (rekordów), z których każdy zawiera dane jednej osoby. Zbiory D , D' różnią się co najwyżej jednym wierszem, co oznacza, że jeden zbiór jest podzbiorem drugiego, a większy zbiór danych zawiera dokładnie jeden dodatkowy wiersz. Prawdopodobieństwo losowych wyborów dokonywanych przez K jest wyrażane przez parametr ϵ . Im mniejsza wartość ϵ , tym większa gwarancja ochrony prywatności²¹.

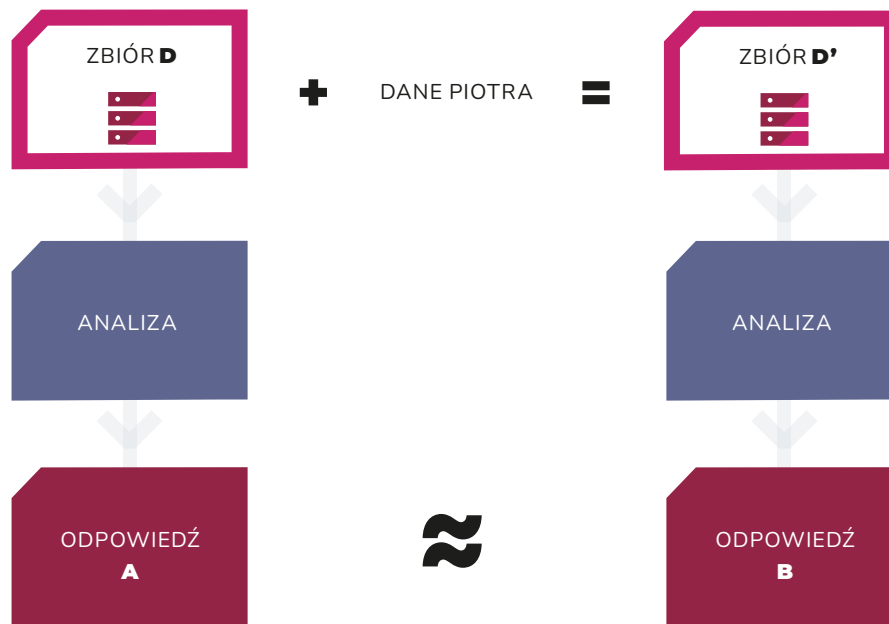
W modelu prywatności różnicowej ochrona prywatności dotyczy właściwości metody udostępniania danych, a nie właściwości, jaką można przypisać zbiorowi danych. Spełnienie warunku prywatności różnicowej oznacza, że możliwe jest uzyskanie dokładnych informacji ze zbioru danych przy jednoczesnym zapewnieniu braku możliwości identyfikacji poszczególnych osób bądź jednostek, których dane dotyczą. Spełnienie warunku prywatności różnicowej zapewnia, że pojedynczy rekord nie wpływa znacząco na wynik analizy w całym zbiorze danych.

Prywatność różnicowa wiąże się z paradoksem polegającym na zablokowaniu możliwości nauki informacji o jednostce podczas uczenia się przydatnych informacji o populacji. Model daje gwarancję, że prawdopodobieństwo wyników otrzymanych ze zbioru poddanego anonimizacji istotnie się nie zmieni, jeśli dane konkretnej osoby fizycznej zostaną dodane bądź usunięte ze zbioru wejściowego. Dlatego atakującym trudno jest uzyskać informacje o konkretnych osobach. Prywatność różnicowa wymaga, aby rozkład prawdopodobieństwa wyników analizy był zasadniczo taki sam, niezależnie od tego, czy jakakolwiek

²⁰ Dwork, C. (2006). *Differential Privacy*. <http://www.audentia-gestion.fr/MICROSOFT/dwork.pdf>

²¹ Tamże.

osoba zdecyduje się wziąć udział w badaniu czy też nie. Prywatność różnicowa zapewnia, że zanonimizowany widok zbioru danych generowany z danych rzeczywistych (kompletnych) na potrzeby określonej osoby trzeciej (np. analityka, badacza) doprowadzi do równie prawdopodobnych wyników niezależnie od obecności lub nieobecności jakiegokolwiek osoby.



Rysunek 3: Uproszczony schemat warunku prywatności różnicowej

Założmy, że dane, które chcemy chronić, to dane dotyczące palenia tytoniu w wybranym regionie uzupełnione o wyniki badań laboratoryjnych, a wśród palaczy, których dane wzbogacają zbiór danych, jest Piotr. Prywatność różnicowa gwarantuje Piotrowi oraz badaczom, że w wyniku analizy zbioru danych zostaną wyciągnięte te same wnioski, np.: palenie powoduje raka, niezależnie od tego, czy dane Piotra będą zawarte w zbiorze danych czy też nie. Odpowiedzi uzyskane ze zbioru zawierającego dane Piotra oraz ze zbioru bez tych danych powinny być nie do odróżnienia, a na podstawie samego zbioru danych nie będzie można stwierdzić, ani czy dane Piotra są w nim zawarte, ani co zawierały.

Chcąc spełnić warunek prywatności różnicowej, stosuje się głównie metody oparte na dodawaniu zakłóceń w postaci szumu. Ilość dodawanego szumu musi być na tyle duża, aby ukryć tożsamość poszczególnych osób i na tyle mała, aby nie zniekształcić prawdziwości odpowiedzi na pytanie, jakie stawia użytkownik danych. Dlatego prywatność różnicowa rozpatruje również to, jak bardzo zmieni się odpowiedź na pytanie (jak zmienia się wartość statystyki), biorąc pod uwagę nieobecność lub obecność najbardziej skrajnej wartości w zbiorze danych. Załóżmy, że dane, które chcemy chronić, to dane dotyczące zarobków w pewnym regionie, a statystyka będąca obiektem naszego badania to mediana. Dodatkowo w zbiorze występuje jedna wartość odstająca i dotyczy zarobków pewnego miliardera. Dzięki prywatności różnicowej w zanonimizowanym widoku danych wartość mediany nie powinna ulec drastycznej zmianie – nieważne czy dane miliardera będą w nim obecne czy też nie²². Prywatność różnicowa ma zastosowanie w niezliczonych środowiskach, np. w spisach, statystykach pracy, dokumentacji medycznej, analizie strategii biznesowych itd.

Dane nieustrukturyzowane

Kwestie badawcze, prawnicze i techniczne dotyczące anonimizacji przez wiele lat zdominowane były przez dane ustrukturyzowane. Pod względem ilościowym ustępują one jednak danym nieustrukturyzowanym, które – jak szacują eksperci – do 2025 roku będą stanowiły 80% danych na świecie²³. Dane nieustrukturyzowane, takie jak dokumenty tekstowe lub obrazy, są więc znacznie powszechniejsze, a generują je codziennie tysiące organizacji, które często nie robią z nich żadnego użytku. Potencjał danych nieustrukturyzowanych do tworzenia wartości dla organizacji i firm, rozwoju uczenia maszynowego i innych technologii w świecie Big Data nie jest wykorzystywany.

Podział danych na ustrukturyzowane i nieustrukturyzowane opiera się na sposobie formatowania. Ustrukturyzowane zbiory danych cechują się precyzyjnie określonym formatem, zazwyczaj wyraźnie zdefiniowanym z góry

22 Dwork, C., Roth, A. (2014). *The Algorithmic Foundations of Differential Privacy*. <https://www.cis.upenn.edu/~aaroht/Papers/privacybook.pdf>

23 King, T. (2019, 28 marca). *80 Percent of Your Data Will Be Unstructured in Five Years*. Solutions Review. Pobrano 22.07.2022 z <https://solutionsreview.com/data-management/80-percent-of-your-data-will-be-unstructured-in-five-years/>

i mocno zależnym od systemu bazy danych. Natomiast dane nieustrukturyzowane nie mają predefiniowanego szablonu. Nie zmienia to jednak faktu, że zarówno dane ustrukturyzowane, jak i nieustrukturyzowane mogą zawierać dane osobowe, a zatem podlegają RODO. Chociaż RODO wskazuje, że zasady ochrony danych powinny mieć zastosowanie do wszelkich informacji o zidentyfikowanych lub możliwych do zidentyfikowania osobach fizycznych²⁴, większość wypracowanych przez lata metod, włącznie z rekomendowanymi przez grupę roboczą ds. ochrony osób fizycznych w zakresie przetwarzania danych osobowych, koncentruje się na anonimizacji danych ustrukturyzowanych. Wymagania, które muszą zostać spełnione, aby właściwie zanonimizować nieustrukturyzowane dane, pozostają niejasne i niedostatecznie rozpatrzone. Część operacji można zaadaptować do danych niestrukturalnych, ale zakres możliwych do użycia technik jest znacznie bardziej ograniczony.

Głównym wyzwaniem związanym z anonimizacją danych nieustrukturyzowanych jest to, że nie są one zgodne ze stałym szablonem. Dane tekstowe mogą kryć dane osobowe w różnych, trudnych do przewidzenia z góry, sformułowaniach językowych. Język naturalny jest niejednoznaczny, a dany wyraz lub fraza, przy uwzględnieniu czynników kontekstowych, może przyczynić się do naruszenia zasad RODO. Podobne wyzwania pojawiają się podczas przetwarzania obrazów, które mogą zawierać nie tylko najbardziej podstawowe elementy tożsamości, jak np. twarz, ale również bardziej pośrednie informacje możliwe do powiązania z osobami fizycznymi, np. tablice rejestracyjne, widok prywatnego domu lub jego wnętrza czy szczegóły anatomiczne w medycznych danych obrazowych.

Dane nieustrukturyzowane mają często charakter wielowymiarowy. W przypadku danych tekstowych wymiarem mogą być pojedyncze słowa (dokument jako sekwencja słów), natomiast w przypadku obrazów piksele (obraz jako zbiór pikseli). Oddzielne wymiary mogą nie wyrażać bezpośrednio związków z daną osobą, ale wspólnie mogą tworzyć atrybuty wrażliwe. Mapowanie między danymi nieustrukturyzowanymi a danymi osobowymi jest niejednoznaczne i często niejawne, dlatego anonimizacja danych nieustrukturyzowanych wymaga odpowiedniego podejścia. W ciągu ostatnich lat opracowano różne metody, głównie oparte na technikach uczenia maszynowego, do wykrywania i anonimizacji wrażliwych atrybutów i quasi-identyfikatorów w danych nieustrukturyzowanych. Zadanie to

24 Rozporządzenie Parlamentu Europejskiego i Rady (UE) 2016/679 z dnia 27 kwietnia 2016 r. w sprawie ochrony osób fizycznych w związku z przetwarzaniem danych osobowych i w sprawie swobodnego przepływu takich danych oraz uchylenia dyrektywy 95/46/WE (ogólne rozporządzenie o ochronie danych). <https://uodo.gov.pl/pl/131/224>

wciąż jednak pozostaje nietatwym wyzwaniem technologicznym, przede wszystkim ze względu na złożoność tego typu danych.

Problem anonimizacji danych tekstowych sprowadza się w dużej mierze do wykrywania danych osobowych i danych wrażliwych. Przykładowymi metodami wykrywania takich danych są:

- dopasowywanie wyrażeniami regularnymi – dzięki tej metodzie można w prosty i szybki sposób, poprzez zastosowanie odpowiednio skonstruowanego wyrażenia, wykryć np. kwoty, liczby, różnego rodzaju numery identyfikacyjne; dodatkowo można do takich wyrażen dołączyć walidację poprawności wykrytych danych (np. sprawdzanie cyfry kontrolnej przy numerze PESEL), co też ograniczy otrzymane wyniki do bardziej rzeczywistych;
- przeszukiwanie słownikowe – w podstawowym wariancie jest to porównanie słowa bądź frazy z danymi zawartymi w przygotowanym słowniku (przy czym można uwzględnić lub zignorować wielkość liter); takie przeszukiwanie nie zawsze jednak się sprawdzi, szczególnie w przypadku literówek w wyrazach, stąd też bardziej rozbudowanym wariantem tej metody jest zastosowanie rozmytego dopasowywania (*fuzzy matching*), które jest w stanie uwzględnić pewną różnicę (np. wykorzystując wyliczenia odległości Levenshteina) pomiędzy frazami zawartymi w słowniku a porównywanym tekstem;
- zastosowanie wytrenowanych modeli sztucznej inteligencji służących do rozpoznawania określonych jednostek nazwanych (*named entity recognition, NER*) – dzięki wcześniejszemu wytrenowaniu takich modeli możliwe jest wykrycie z pewną dokładnością bardziej skomplikowanych wyrażen; wykorzystywane są do tego różne właściwości przetwarzanego tekstu (treść wyrazów, części mowy, formy wyrazów i ich odmiany, a nawet kontekst otoczenia).

Po wykryciu danych do anonimizacji można z powodzeniem zastosować techniki opisane w pierwszej części tego rozdziału.

Anonimizacja danych obrazowych polega na nieodwracalnym zniekształceniu obrazów, tak aby nie można było odtworzyć danych osobowych. Powszechnie stosowane techniki to m.in.:

- zaciemnianie (*blacking out*) – zastąpienie odpowiedniego obszaru jednolitym kolorem;
- pikselizacja (*pixelation*) – znaczące obniżenie rozdzielczości odpowiedniego obszaru;
- rozmycie (*blurring*) – zamazywanie lub rozmazywanie detali odpowiednich obszarów poprzez nakładanie filtrów na obrazy, np. filtrów uśredniających lub gaussowskich.

Dużym wyzwaniem jest również wykrywanie obszarów zawierających informacje umożliwiające identyfikację osób fizycznych. W większości przypadków proces ten ogranicza się do wykrywania twarzy i tablic rejestracyjnych. Obecnie osiąga się to za pomocą metod głębokiego uczenia, zwłaszcza poprzez splotowe sieci neuronowe (*convolutional neural network*, CNN). Wymienione metody mogą oszukać ludzkie oko, ale są technicznie odwracalne. Nowatorską techniką, która przewycięża to zagrożenie, jest zastąpienie wrażliwych obiektów sztucznie generowanymi obiektami lub ich usuwanie poprzez zastąpienie ich sztucznie wygenerowanym tłem. W tym celu wykorzystuje się sieci neuronowe typu GAN (*generative adversarial network*), czyli generatywne sieci przeciwstawne. Stosowanie tej metody ma jednak swoje wady. W przypadku zastąpienia twarzy sztucznie wygenerowaną, istnieje prawdopodobieństwo, że będzie ona podobna do innej, prawdziwej twarzy. Ponadto po usunięciu obiektów zaklasyfikowanych jako wrażliwe, wygenerowane obrazy mogą okazać się bezwartościowe dla danego zbioru danych – np. usuwając z obrazu samochód, można usunąć również niektóre oznaczenia drogowe będące istotą zbioru danych.

Pseudonimizacja

Anonimizacja to trwałe zastąpienie lub usunięcie danych osobowych. Raz zanonimizowane dane powinny uniemożliwiać ponowną identyfikację osób fizycznych. Anonimizacja jest dobrą strategią ochrony danych osobowych, ale nie jest regulowana przez RODO, ponieważ przy przetwarzaniu danych prawdziwie anonimowych zasady ochrony danych nie mają dłużej zastosowania. Zgodnie z motywem 26 dyrektywy 95/46/WE:

„[...] Zasady ochrony danych nie powinny więc mieć zastosowania do informacji anonimowych, czyli informacji, które nie wiążą się ze zidentyfikowaną lub możliwą do zidentyfikowania osobą fizyczną, ani do danych osobowych zanonimizowanych w taki sposób, że osób, których dane dotyczą, w ogóle nie można zidentyfikować lub już nie można zidentyfikować”²⁵.

Innym środkiem w zakresie bezpieczeństwa danych wskazanym przez RODO jest pseudonimizacja. W przypadku pseudonimizacji dane osobowe są zastępowane w taki sposób, aby przy użyciu dodatkowych informacji możliwe było ponowne zidentyfikowanie osób, których dane dotyczą. Zgodnie z art. 4 pkt. 5 RODO:

„[...]„pseudonimizacja” oznacza przetworzenie danych osobowych w taki sposób, by nie można ich było już przypisać konkretnej osobie, której dane dotyczą, bez użycia dodatkowych informacji, pod warunkiem, że takie dodatkowe informacje są przechowywane osobno i są objęte środkami technicznymi i organizacyjnymi uniemożliwiającymi ich przypisanie zidentyfikowanej lub możliwej do zidentyfikowania osobie fizycznej”²⁶.

Zatem, w przeciwieństwie do danych zanonimizowanych, dane pseudonimizowane kwalifikują się jako dane osobowe i podlegają przepisom o ochronie danych. Dlatego należy pamiętać, że anonimizacja i pseudonimizacja nie są pojęciami równoznacznymi.

Pseudonimizacja polega na zastąpieniu wszelkich informacji o zidentyfikowanych lub możliwych do zidentyfikowania osobach fizycznych (np. imię i nazwisko, PESEL, adres e-mail itp.) fikcyjnymi identyfikatorami (pseudonimami), przy jednoczesnym wyodrębnieniu i zabezpieczeniu identyfikatorów umożliwiających ewentualne odwrócenie procedury. W procesie pseudonimizacji z danych pierwotnych otrzymuje się dwa zestawy informacji:

- zbiór danych pseudonimizowanych, zawierający informacje, których nie można powiązać z żadną osobą fizyczną,
- dodatkowe informacje, zawierające identyfikatory umożliwiające przypisanie danych do konkretnej osoby.

25 Tamże.

26 Tamże.

W razie potrzeby oryginalne dane są możliwe do zrekonstruowania, ale tylko przez uprawnionych użytkowników, mających dostęp do dodatkowych informacji. Anonimizacja danych może zniszczyć wartość, którą zawierają dane. Dzięki pseudonimizacji dane nie są uważane za bezpośrednio identyfikujące, ale nie tracą swojej pierwotnej wartości, umożliwiając analizę danych, przy jednoczesnej ochronie prywatności osób, których dane dotyczą. Pseudonimizacja danych osobowych może pomóc administratorom przetwarzającym dane osobowe wywiązać się z obowiązku ochrony danych, ale jej stosowanie powinno zależeć od okoliczności, zastosowanej technologii, a nawet poziomu związanego z nią ryzyka.

Grupa Robocza Artykułu 29 wymienia pięć technik pseudonimizacji: szyfrowanie z kluczem tajnym, funkcję skrótu, funkcję skrótu z kluczem, szyfrowanie deterministyczne i tokenizację²⁷.

Szyfrowanie z kluczem tajnym

Algorytmy szyfrujące przekształcają dane do postaci szyfru, który mogą odczytać tylko osoby posiadające tajny klucz. Jest to stosunkowo prosta i łatwa w użyciu metoda chroniąca dane osobowe przed osobami nieuprawnionymi do ich przetwarzania, jednak niesie za sobą pewne ryzyko: klucz musi być w posiadaniu zarówno nadawcy, jak i odbiorcy danych, co wymaga wypracowania bezpiecznej metody dystrybucji kluczy. Klucz to zazwyczaj długa, losowo lub pseudolosowo generowana sekwencja bitów. Jego złożoność i długość przekładają się na to, jak łatwo atakujący będzie mógł wykonać atak *brute force*, w którym wypróbowywane są wszystkie możliwe kombinacje znaków tworzących klucz, aż do momentu znalezienia tego właściwego. W przypadku szyfrowania z kluczem tajnym, do szyfrowania i deszyfrowania używany jest tylko jeden klucz. Zarówno nadawca, jak i odbiorca wiadomości muszą mieć kopię tajnego klucza, aby móc odczytać wiadomość. Wymiana klucza podczas wysyłania zaszyfrowanych wiadomości niesie za sobą ryzyko, że osoba trzecia przechwyci klucz, dzięki czemu będzie mogła łatwo odszyfrować dane^{28 29}.

27 Grupa Robocza Artykułu 29 (przyjęta w dniu 10 kwietnia 2014r.), dz. cyt.

28 Tamże.

29 Littlejohn, S., Michael C. (2008). *Chapter 12 - Understanding Cybercrime Prevention*. <https://www.sciencedirect.com/science/article/pii/B9781597492768000121>

Funkcja skrótu

Za pomocą funkcji skrótu można przekształcić dowolny zbiór danych w ciąg znaków o stałej długości (*hash value*), niezależnie od rozmiaru danych wejściowych. Wygenerowane z danych wejściowych wartości są trudne do odtworzenia: łatwo jest wyznaczyć skrót dla danej wartości, natomiast operacja odwrotna, tj. obliczenie wartości wejściowej na podstawie samego skrótu, nie jest bezpośrednio możliwa. Funkcja skrótu powinna być bezkonfliktowa, co oznacza, że prawdopodobieństwo uzyskania tego samego skrótu dla różnych wartości wejściowych powinno być bardzo niskie. Każda drobna zmiana wartości wejściowej spowoduje zmianę skrótu. Chociaż skróty nie mogą zostać użyte do odszyfrowania oryginalnych wartości, to mogą być użyte do określenia czy wartości te uległy zmianie. W ten sposób skróty zapewniają poufność, ale nie integralność³⁰.

Funkcja skrótu z kluczem

Algorytm ten łączy funkcję skrótu z kluczem kryptograficznym (tajnym kluczem). Aby otrzymać skrót, należy na wejściu podać nie tylko dane, ale również klucz. Zatem skróty nie są już tylko wrażliwe na zmiany samych danych, ale zależą również od wartości klucza. Funkcje skrótu używające różnych kluczy będą generować różne skróty dla tych samych wartości wejściowych. Bez klucza odtworzenie danych jest niemożliwe, dlatego zapewnienie ochrony danych zależy już nie tylko od zastosowanej funkcji skrótu, ale również od złożoności i długości klucza oraz jego bezpiecznego przechowywania³¹.

Szyfrowanie deterministyczne

Metoda szyfrowania deterministycznego w każdej iteracji algorytmu, przy użyciu tych samych danych wejściowych (tekstu jawnego) i klucza, zwraca ten sam zaszyfrowany tekst. Przeprowadzając analizę statystyczną bądź budując słownik par wartości jawnych z wartościami zaszyfrowanymi, możliwe jest skorelowanie zaszyfrowanych wartości z zaobserwowanymi działaniami

30 Grupa Robocza Artykułu 29 (przyjęta w dniu 10 kwietnia 2014r.), dz. cyt.

31 Tamże.

i odtworzenie wartości prawdziwych. Ten problem jest na tyle poważny, że kryptografowie zaproponowali pojęcie szyfrowania probabilistycznego, które wprowadza element przypadku: wartości prawdziwe będą przedstawione za pomocą jednej z wielu możliwych wartości wybranych losowo podczas procesu szyfrowania, co ma uniemożliwić atakującemu tworzenie powiązań między wartościami zaszyfrowanymi a prawdziwymi, gwarantuje więc tzw. bezpieczeństwo semantyczne. Każdy proces szyfrowania deterministycznego można przekształcić w proces szyfrowania probabilistycznego, dołączając nową losową wartość do wartości prawdziwych (tekstu jawnego) przed szyfrowaniem deterministycznym i usuwając ją po deterministycznym odszyfrowaniu³².

Tokenizacja

W procesie tokenizacji, która jest formą maskowania danych, prawdziwe wartości zastępowane są przez losowy ciąg znaków, tzw. tokeny. Co istotne, same w sobie nie mają one żadnej wartości, a także nie mają matematycznego związku z oryginalnymi danymi. Przypomnijmy, że techniki szyfrowania i funkcje skrótu jedynie zaciemniają dane. Pseudonimy, które powstają w ich procesie, uzyskuje się za pomocą określonej formuły matematycznej, więc odzyskanie oryginalnych wartości jest możliwe, gdy zostanie odkryty właściwy wzór. Tokenizacja usuwa oryginalne dane i zastępuje je tokenami, nie wykorzystując przy tym żadnych formuł, dlatego prawdziwe wartości są praktycznie niemożliwe do odtworzenia. Po zakończeniu procesu tokenizacji oryginalne dane, zawierające dane osobowe, są przechowywane w bezpiecznym miejscu, tzw. skarbcu, który jest jedynym sposobem na powiązanie danych osobowych z odpowiadającymi im tokenami. Tokenizację można zastosować do każdego rodzaju danych wymagających ochrony, ale technika ta jest zwykle kojarzona z sektorem finansowym, gdzie jest wykorzystywana przy przetwarzaniu transakcji bankowych, wniosków kredytowych, obrotu giełdowego i innych³³.

32 Bellare, M., Boldyreva, A., O'Neill, A. (2007). *Deterministic and Efficiently Searchable Encryption*. <https://eprint.iacr.org/2006/186.pdf>

33 PCI Security Standards Council. (2015). *Tokenization Product Security Guidelines*. https://lists.pcisecuritystandards.org/documents/Tokenization_Product_Security_Guidelines.pdf; Turner, C. (2020, 14 grudnia). *Tokenization vs. Encryption: Pros and Cons*. eSecurity Planet. Pobrano 13.07.2022 z <https://www.esecurityplanet.com/threats/tokenization-vs-encryption/>

Dane syntetyczne

Techniki anonimizacji danych wypracowane ponad dekadę temu nie są wystarczające w realiach współczesnego świata. Obowiązek zapewnienia ochrony danych i prywatności osób fizycznych, a także coraz większe możliwości atakujących i rosnące zagrożenie wyciekami danych powodują, że normy etyczne i prawne stają się bardziej rygorystyczne. Często dochodzi do sytuacji, że organizacje nie wykorzystują własnych surowych danych nawet do celów wewnętrznych. Ze względu na obowiązujące przepisy bardzo trudno jest udostępnić dane wysokiej jakości analitykom i twórcom nowych technologii. Wiele projektów jest odrzucanych jeszcze przed ich rozpoczęciem, ponieważ nie ma możliwości pozyskania niezbędnych do realizacji danych.

Aby udostępnić dane bez naruszenia prywatności, muszą być one w pełni anonimowe. Bez względu na to, jakie techniki anonimizacji zostaną zastosowane, zawsze należy liczyć się z tym, że kosztem wysokiego poziomu prywatności będzie utrata użyteczności danych. Dane, które są w pełni anonimowe, nie mają wielkiej wartości dla analizy statystycznej, a kompromis między prywatnością i użytecznością danych jest trudny do osiągnięcia. Ponowna identyfikacja osób na podstawie danych anonimizowanych lub pseudonimizowanych jest ciągle możliwa, ponieważ zbiory te (nawet po zastosowaniu odpowiednich technik) utrzymują niejawnie powiązanie z osobami fizycznymi.

Coraz popularniejszym sposobem na przezwycięzenie tych problemów i ograniczeń jest wykorzystanie danych w pełni syntetycznych. Stanowią one imitację rzeczywistych danych wygenerowaną z modeli, które są wytrenowane w celu odtwarzania cech danych rzeczywistych. W przeciwieństwie do danych anonimizowanych i pseudonimizowanych dane syntetyczne nie są modyfikacją danych rzeczywistych. Składają się z nowych, całkowicie fałszywych, ale realistycznych informacji, których nie można powiązać z prawdziwymi osobami.

Z tego powodu wykorzystanie danych syntetycznych na potrzeby badań i wdrażania sztucznej inteligencji staje się atrakcyjną alternatywą dla danych zanonimizowanych (utrata zbyt wielu informacji) i pseudonimizowanych (duża podatność na ataki i reidentyfikację osób fizycznych). Dane syntetyczne można generować z danych rzeczywistych lub bez ich udziału, z wykorzystaniem istniejących modeli lub wiedzy specjalistycznej. W przypadku syntezy z danych rzeczywistych analitycy sięgają po zbiory danych (publiczne lub niepubliczne) do budowy modeli zdolnych do uchwycenia rozkładów i struktury danych,

zwłaszcza wieloczynnikowych relacji i interakcji między danymi. Dane syntetyczne są próbkowane lub generowane ze zbudowanego modelu, a ich użyteczność zależy od tego, jak dokładnym są zamiennikiem danych rzeczywistych (analiza danych syntetycznych powinna dawać wyniki porównywalne z wynikami analizy danych rzeczywistych). Synteza na podstawie istniejących modeli lub wiedzy specjalistycznej polega na wykorzystaniu opracowanych modeli procesów lub ich symulacji. Użyteczność tego typu danych będzie niższa, ponieważ zależy przede wszystkim od tego, jak dobrze analitycy znają dziedzinę, której dane dotyczą, oraz jak bardzo złożone jest badane zjawisko. Dobrą strategią jest zastosowanie podejścia hybrydowego, gdzie generowanie na bazie danych rzeczywistych jest uzupełniane danymi uzyskanymi w wyniku symulacji oraz wiedzą specjalistyczną.

Możliwości danych syntetycznych:

- ograniczenie obaw związanych z udostępnianiem i ponownym wykorzystaniem danych;
- zastosowanie do modelowania i przewidywania błędów istniejących modeli i rozwiązań;
- przyspieszenie i ujednoczenie badań nad głodnymi danymi modelami sztucznej inteligencji;
- kontrolowanie stroniczości danych;
- pomoc przy wyjaśnialności (*explainability*) modeli sztucznej inteligencji oraz weryfikacji wiarygodności polityk opartych na sztucznej inteligencji;
- potencjał tworzenia realistycznego świata cyfrowego.

Ograniczenia danych syntetycznych:

- wymagają wysokiego poziomu wiedzy dziedzinowej, statystycznej i z zakresu sztucznej inteligencji;
- wymagają dużych ilości danych źródłowych (na małych zbiorach synteza jest bardzo utrudniona lub wręcz niemożliwa);

- ograniczony dostęp do danych rzeczywistych, uniemożliwiający tworzenie wartościowych danych syntetycznych (jakość danych zależy od jakości modelu, a jakość modelu zależy od źródła danych).

W ostatnich latach opracowano wiele metod generowania danych syntetycznych, począwszy od wysoce precyzyjnych, ale o bardzo ograniczonej skalowalności modeli statystycznych, a skończywszy na modelach uczenia głębokiego o wysokiej spójności i skalowalności, określanymi jako czarne skrzynki (black-box). Dobrze opracowane i sprawdzone pod względem jakości dane syntetyczne mogą zminimalizować ryzyka związane z ochroną danych, ale nadal wymagają przeprowadzenia prac badawczych w zakresie metod syntezy danych, ich skuteczności oraz rzetelnego porównania metod generowania danych syntetycznych z innymi technikami ochrony prywatności danych. Nowe metody są stale rozwijane, ponieważ koncepcja danych syntetycznych niesie za sobą poważną zmianę w zakresie postrzegania możliwości efektywnego wykorzystania danych do rozwoju współczesnych technologii.

Szczegółowe omówienie tematu generowania danych tekstowych, obrazowych i tabelarycznych oraz możliwości ich zastosowania zawarte jest w drugiej części publikacji.



Część II: Generowanie danych syntetycznych

Generowanie języka naturalnego

Celem przetwarzania języka naturalnego (*natural language processing*, NLP) jest umożliwienie komputerom zrozumienia ludzkiego języka zarówno w formie pisemnej, jak i ustnej. Ta interdyscyplinarna dziedzina, będąca połączeniem informatyki, sztucznej inteligencji, lingwistyki oraz nauki o danych, składa się zasadniczo z dwóch komponentów: rozumienia języka naturalnego (*natural language understanding*, NLU) oraz generowania języka naturalnego (*natural language generation*, NLG). Podczas gdy NLU dąży do uchwycenia sensu informacji pozyskiwanych z mowy lub tekstu, NLG tworzy na podstawie danych możliwą do zrozumienia i znaczącą narrację.

Generowanie języka naturalnego polega na syntetycznym tworzeniu fraz i zdań, które opisują, podsumowują lub wyjaśniają dane w sposób podobny do ludzkiego. Podobnie jak w przypadku NLU, systemy NLG muszą uwzględniać reguły językowe oparte na morfologii, leksykonach, składni i semantyce oraz uwzględniać informacje najbardziej odpowiednie w bieżącym kontekście. Co więcej, generowana treść powinna być prezentowana w logiczny sposób, z zachowaniem odpowiedniej organizacji i struktury oczekiwanego tekstu. To wszystko sprawia, że NLG jest wieloetapowym procesem, w którym każdy krok dalej przyczynia się do tworzenia naturalnie brzmiących treści.

Rozwiązania NLG, mimo sukcesów z ostatnich lat, są w dalszym ciągu ograniczone w porównaniu z ludzkim – kreatywnym i nacechowanym emocjonalnie – pisaniem tekstów. Niemniej jednak dzisiejsze systemy NLG są zdolne do przekształcania danych w narracje przy użyciu szablonów lub dynamicznego tworzenia dokumentów. W podejściu opartym na szablonach teksty mają wstępnie zdefiniowaną strukturę z pustymi miejscami, które są uzupełniane danymi. Dzięki zastosowaniu m.in. łańcuchów Markowa, rekurencyjnych sieci neuronowych i architektury Transformer, systemy NLG ewoluowały, umożliwiając dynamiczne tworzenie treści: zdania są budowane z reprezentacji semantycznych i pożądanej struktury językowej, a następnie organizowane w logiczną całość dostosowaną do celu komunikacji.

Przykłady możliwych zastosowań NLG:

- generowanie odpowiedzi do chatbotów, asystentów głosowych (np. Alexa od Google lub Siri od Apple) i innych konwersacyjnych systemów AI (prowadzenie dwukierunkowych rozmów);

- sugerowanie tekstu w klientach poczty e-mail, personalizowanie odpowiedzi na wiadomości od klientów;
- generowanie narracji tekstowej na podstawie danych strukturalnych (np. arkusza kalkulacyjnego), przekształcanie danych biznesowych, raportów finansowych i innego rodzaju danych w łatwy do zrozumienia język;
- tworzenie opisów produktów na strony e-commerce;
- generowanie raportów o stanie urządzeń internetu rzeczy (IoT);
- agregowanie i przedstawianie w formie podsumowań doniesień informacyjnych;
- generowanie i personalizowanie scenariuszy wykorzystywanych w obsłudze klienta;
- generowanie większych fragmentów tekstu o spójnym temacie i rozwijającej się historii (np. artykułów naukowych, wpisów blogowych itp.);
- generowanie spójnej narracji z zestawu przesłanek lub krótkiego podsumowania;
- parafrazowanie, czyli przekształcanie zdania w języku naturalnym na nowe zdanie, które ma to samo znaczenie semantyczne, ale inną formę składniową lub leksykalną;
- tworzenie transkrypcji (połączenie technik rozpoznawania mowy, rozumienia dźwięku i NLG do przekształcania dźwięku w tekst);
- tworzenie podsumowań, które zawierają istotne informacje oryginalnej treści;
- przewidywanie brakujących fragmentów tekstu;
- tłumaczenie maszynowe.

Przetwarzanie języka naturalnego odgrywa istotną rolę w wielu systemach i powszechnie używanych technologiach. Rozwój NLP, a w szczególności NLG, któremu poświęcony jest ten rozdział, odbywa się jednak z pewnym ograniczeniem: dotyczy przede wszystkim języka angielskiego. Postęp często nie przekłada się na inne języki, ponieważ mają one odmienne struktury gramatyczne. Zastosowanie opracowanych dla języka angielskiego metod jest szczególnie trudne w językach silnie fleksyjnych (odnosi się to również do języka polskiego). Oczywiście występują modele tłumaczeń, również oparte na NLG, z pozostałych języków na język angielski. Jednak z uwagi na wielkość tych modeli, stosowanie obejścia w formie tłumaczenia w wersjach produkcyjnych systemów jest wysoce nieoptymalne (spowolnienie od jednego do trzech rzędów wielkości w czasie przetwarzania) oraz zdecydowanie biznesowo nieatrakcyjne.

Od niedawna wersje różnych modeli tworzone są w formie wielojęzycznej, lecz język polski jest najczęściej niewspierany. Wynika to ze skomplikowanej gramatyki, fleksji i specyficzności polszczyzny. Polskie wersje modeli NLP (od niedawna również NLG) trenowane są również przez zdecydowanie mniejszą społeczność i pozostawiają wiele do życzenia pod kątem jakości. Głównymi instytucjami ułatwiającymi rozwój narzędzi NLP/NLG w języku polskim są Instytut Podstaw Informatyki PAN, Politechnika Wrocławska i Ośrodek Przetwarzania Informacji – Państwowy Instytut Badawczy. Niemniej jednak najczęściej są to modele bazujące na tych opracowanych w języku angielskim 3–6 lat wcześniej, co jest wyjątkowo dużym opóźnieniem w tej szybko rozwijającej się dziedzinie.

Przedstawione w dalszej części publikacji modele bazują na najnowszych osiągnięciach światowych, przez co opis większości z nich dostępny jest aktualnie jedynie w języku angielskim. Wiele z nich pozostawia szerokie pole do adaptacji i rozwoju w języku polskim.

Metody generowania danych tekstowych

Do generowania języka naturalnego wykorzystuje się szereg metod prowadzących do osiągnięcia zadowalających wyników. Jednym z kierunków rozwoju tej dziedziny jest tworzenie metryk najlepiej oddających jakość tekstu. W aspektach składniowych, gramatycznych i wydźwiękowych generowane

teksty powinny być jak najbardziej zbliżone do ludzkich i jednocześnie poprawne językowo. Jednak w połączeniu z teorią informacji analizowane i ulepszone są również inne aspekty wypowiedzi. Podczas gdy do najlepszych metryk należą metoda największego prawdopodobieństwa (top-k) oraz zachłanne wyszukiwanie (*greedy algorithm*), rozwijane są coraz częściej metody bardziej skierowane na informatywność tekstu. Umożliwią one niektórym aktualnym i przyszłym modelom generowanie tekstów przenoszących ilość informacji uzależnioną odwrotnie proporcjonalnie od liczby słów oraz modyfikowalnego parametru³⁴. Różnego rodzaju złożone metody generowania tekstu, trenowane z użyciem kolejnych ulepszanych metryk, pozwalają na coraz bardziej abstrakcyjną kontrolę nad generowanym ciągiem wyrazów. Jednak również najprostsze metody znajdują zastosowanie. Poniżej przedstawiono wybrane techniki generowania języka naturalnego.

Łańcuchy Markowa

Jedną z pierwszych metod używanych do generowania języka naturalnego były łańcuchy Markowa. Ten matematyczny model określa proces stochastyczny, w którym prawdopodobieństwo wystąpienia zdarzenia zależy jedynie od zdarzenia poprzedniego, tj. nie zależy od przejść, które doprowadziły do obecnego stanu (własność Markowa). Łańcuch Markowa składa się ze skończonej liczby stanów i pewnych reguł probabilistycznych p_{ij} , czyli prawdopodobieństw przejścia procesu ze stanu i do stanu j , zapisywanych w postaci tzw. macierzy przejść.

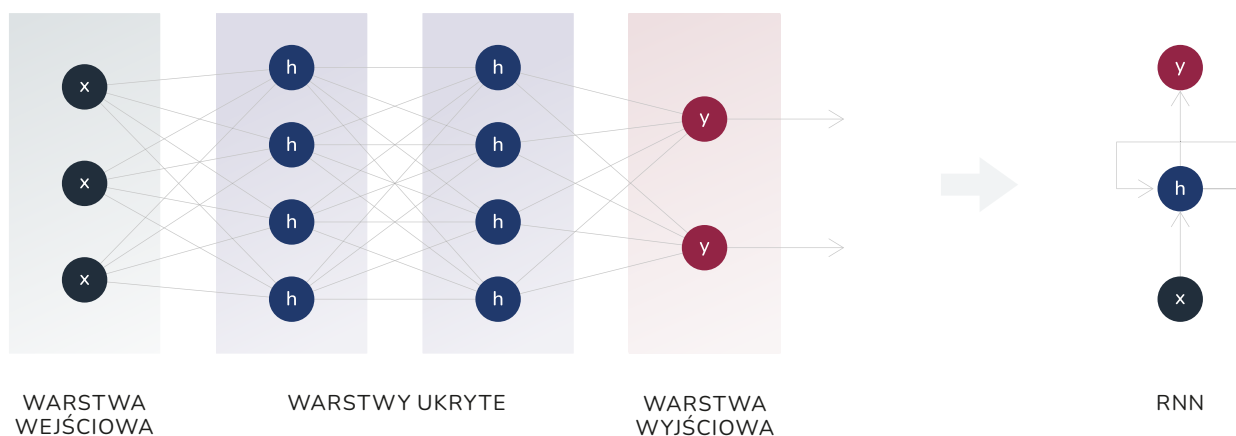
System przechodzi z jednego stanu do drugiego zgodnie z określonymi regułami probabilistycznymi. Przestrzeń stanów może być dowolną wartością. W kontekście generowania tekstu będą to słowa: model przewiduje następne słowo w zdaniu na podstawie ostatniego wpisanego słowa. Model był wykorzystywany m.in. do opracowania funkcji podpowiadania słów podczas pisania tekstu w aplikacjach klawiatury we wczesnych wersjach smartfonów³⁵.

34 Meister, C., Pimentel, T., Wiher, G., Cotterell, R. (2022). *Locally Typical Sampling*. <https://arxiv.org/pdf/2202.00666.pdf>

35 Bikel, D.M., Schwartz, R.M., Weischedel, R.M. (2004). *An Algorithm that Learns What's in a Name*. <http://people.csail.mit.edu/mcollins/6864/slides/bikel.pdf>

Rekurencyjna sieć neuronowa (RNN)

Szeregi czasowe, sygnały audio, obrazy, ruch obiektów, dane tekstowe – do przetwarzania tego rodzaju danych sekwencyjnych mogą służyć rekurencyjne sieci neuronowe (*recurrent neural network*, RNN). W przypadku danych tekstowych RNN, wykorzystując sekwencyjny charakter tekstu, zapamiętuje poprzednie słowa, aby przewidzieć następne słowo. Przekazuje każdy element sekwencji przez strukturę ze sprzężeniem zwrotnym: każdy element otrzymany na wyjściu modelu służy jako wejście do modelu generującego nowe elementy sekwencji. Dzięki temu informacje z poprzednich kroków są przechowywane w sekwencji, czyli pamięci RNN. W każdej iteracji model zapamiętuje poprzednio napotkane słowa. Do każdego słowa przypisuje prawdopodobieństwo jego wystąpienia (na podstawie poprzedniego słowa) i wybiera słowo o największym prawdopodobieństwie. Wraz ze wzrostem długości sekwencji możliwości RNN stają się ograniczone.



Rysunek 4: Rekurencyjna sieć neuronowa

Rekurencyjne sieci neuronowe borykają się z niestabilnym gradientem, zarówno z rzadziej występującym "eksplodującym", jak i tzw. problemem zanikającego gradientu (*vanishing gradient problem*). Znalazienie globalnego minimum funkcji kosztu w sieciach neuronowych, który umożliwia tym samym znalezienie optymalnej konfiguracji sieci, jest realizowane przy wykorzystaniu metody gradientu prostego. W przypadku obliczania kosztu dla sieci RNN informacja jest propagowana wstecz zarówno w kierunku warstwy wyjściowej, jak i przez etapy czasowe warstwy rekurencyjnej. Gdy algorytm propagacji wstecznej przechodzi wstecz przez wszystkie neurony sieci, aby zaktualizować ich wagi, można zauważyć tendencję gradientu do spłaszczania

i stopniowego zanikania spowodowane multiplikacją: jeżeli wielokrotnie wykonywane jest mnożenie przez małą liczbę, wartość gradientu bardzo szybko spada, aż zbliży się do zera. Im dalej algorytm przechodzi przez sieć, tym gradient jest niższy i tym trudniej aktualizować wagi. Wreszcie wagi nie są już aktualizowane lub ich aktualizacja staje się nieistotna, co oznacza, że paraliżowana jest zdolność sieci do uczenia się. W konsekwencji sieć nie jest w stanie przekazywać użytecznych informacji z wyjścia modelu z powrotem do warstw w pobliżu wejścia modelu. To ograniczenie sprawia, że RNN nie jest w stanie zapamiętywać długich sekwencji oraz generować długich i spójnych zdań³⁶.

Sieci LSTM

Aby rozwiązać problem zapamiętywania długich sekwencji, opracowano specjalny wariant RNN – tzw. sieci LSTM (*long short-term memory*). Zyskały one dużą popularność w zadaniach NLP, ponieważ mogą uczyć się kontekstu niezbędnego przy przetwarzaniu długich sekwencji danych tekstowych. Zamiast pojedynczej funkcji aktywacji w LSTM istnieją trzy bramki: bramka wejściowa, bramka zapominania i bramka wyjściowa. W każdym momencie bramki określają, które informacje z przeszłości należy zachować, a które zapomnieć (usunąć). Ogranicza to liczbę poprzednich elementów sekwencji mających wpływ na bieżący stan. Bramka wejściowa kontroluje, jaki sygnał zostanie zapisany w wewnętrznej jednostce stanu, natomiast bramka zapominania reguluje wpływ poprzedniego stanu na stan obecny. Razem wpływają na to, co ma zostać zapisane, a co zapomniane. Bramka wyjściowa reguluje ilość informacji przepływających ze stanu wewnętrznego do wyjścia jednostki i dalej do następnej warstwy. Regulacja przepływu informacji pozwala na zapamiętywanie lub zapominanie słów (np. historii długich sekwencji), które w danym momencie nie są istotne. Dzięki temu sieć selektywnie śledzi informacje, propagując wstecz tylko te istotne, co jednocześnie minimalizuje problem znikającego gradientu. Jednak złożoność ścieżek od poprzedniego do aktualnego stanu skutkuje wysokimi wymaganiami obliczeniowymi utrudniającymi trenowanie modelu i koniecznością wprowadzenia obliczeń równoległych. Ta sama złożoność wpływa również na pamięć sieci, która jest ograniczona do kilkuset słów³⁷.

36 Sherstinsky, A. (2020). *Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network*. <https://arxiv.org/pdf/1808.03314.pdf>; Pascanu, R., Mikolov, T., Bengio, Y. (2012). *On the difficulty of training Recurrent Neural Networks*. <http://proceedings.mlr.press/v28/pascanu13.pdf>

37 Hochreiter, S., Schmidhuber, J. (1997). *Long Short-term Memory*. <http://www.bioinf.jku.at/publications/older/2604.pdf>; Santhanam, S. (2018). *Context based Text-generation using LSTM networks*. <https://arxiv.org/pdf/2005.00048.pdf>

Transformery

Sieć neuronowa typu Transformer³⁸ to innowacyjna architektura, która łączy w sobie strukturę koder-dekoder oraz tzw. mechanizm uwagi (*attention mechanism*).

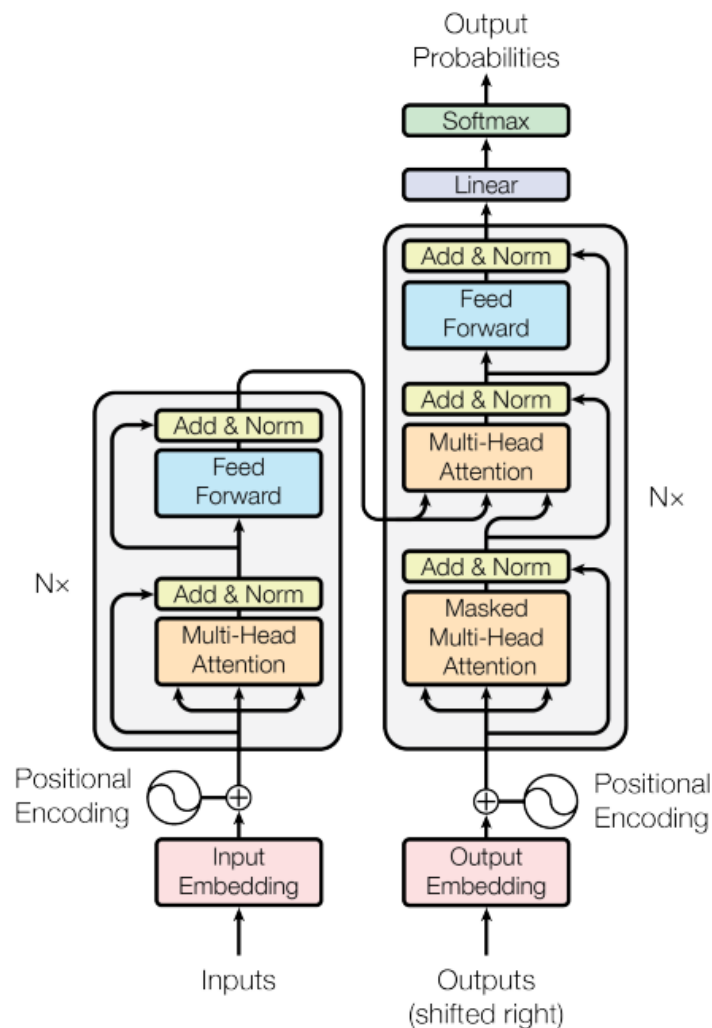


Figure 1: The Transformer - model architecture.

Rysunek 5: Architektura Transformer³⁹

38 Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, Ł., Polosukhin, I. (2017). *Attention Is All You Need*. <https://arxiv.org/pdf/1706.03762.pdf>

39 Tamże.

Transformer składa się z zestawu koderów do przetwarzania danych wejściowych o dowolnej długości oraz zestawu dekodek do wyprowadzania wygenerowanych zdań. Mechanizm uwagi umożliwia uwzględnienie zależności między wszystkimi elementami sekwencji, niezależnie od odległości, która je dzieli. Transformer może modelować zależności w dłuższych sekwencjach, czyli zapamiętywać kontekst słowa. Najpierw przetwarzane są jednocześnie wszystkie słowa zawarte w wejściowej sekwencji i modelowane relacje między nimi. Aby uwzględnić, w jaki sposób słowa odnoszą się do siebie nawzajem, informacje o relacjach są włączane do reprezentacji wektorowej słów (uwagę można modelować za pomocą wag: bardziej istotne relacje lub ważniejsze słowa mają większe wagi). Wszystkie słowa są reprezentowane indywidualnie w przestrzeni wektorowej, bez konieczności redukcji wszystkich informacji do pojedynczego wektora o stałej długości. Dzięki temu możliwe jest modelowanie dłuższych zdań oraz dalekosiężnych zależności językowych bez zwiększania wymagań obliczeniowych^{40 41}.

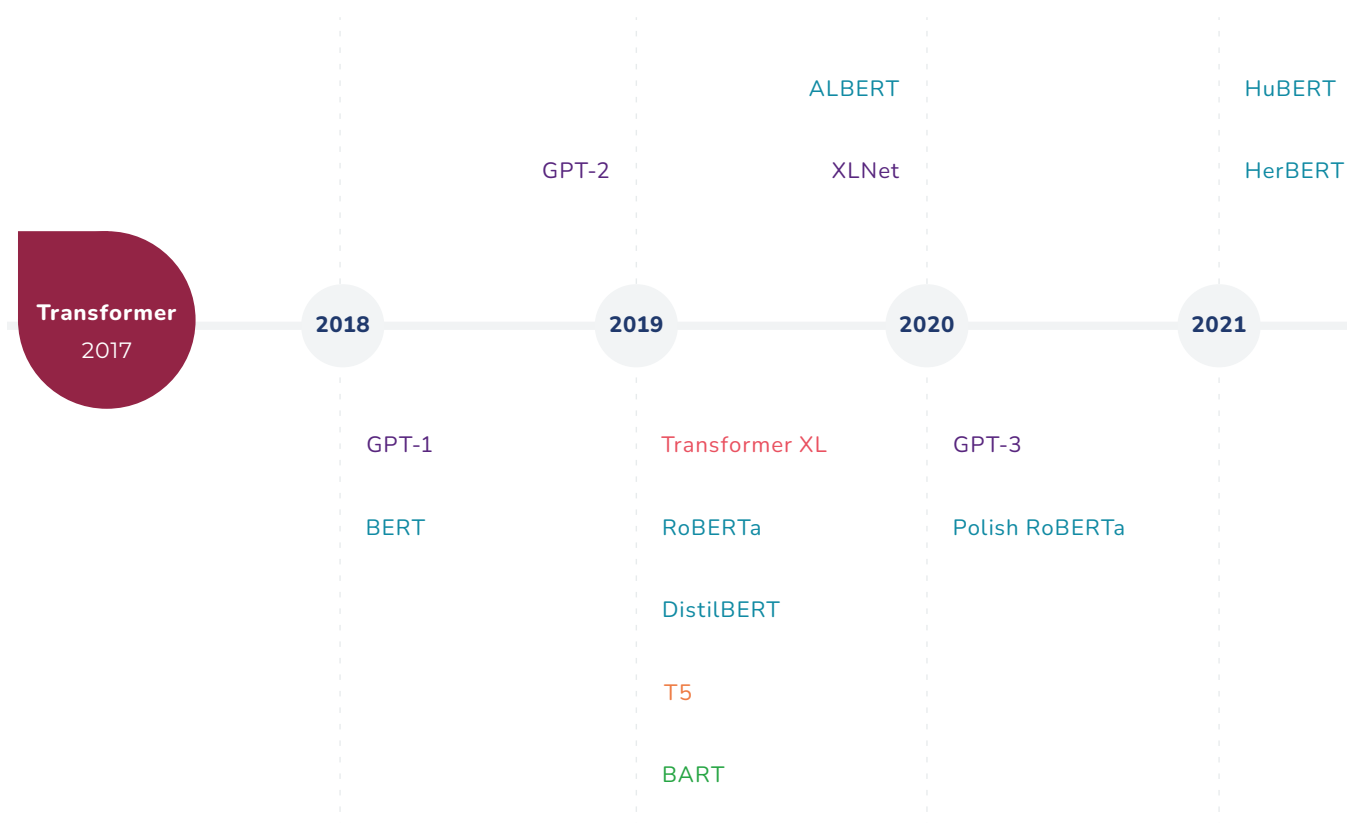
Zastosowanie architektury Transformer istotnie podniosło wydajność rozwiązań NLP, w szczególności NLG. Modele językowe wykorzystujące tę architekturę uwzględniają relacje między wszystkimi słowami w zdaniu, nie ma więc potrzeby redukcji ich w jedną reprezentację (wektor) o stałej długości. Wszystkie słowa są reprezentowane indywidualnie w przestrzeni wektorowej, co umożliwia przetwarzanie dłuższych zdań bez konieczności zwiększania mocy obliczeniowych. Architektura Transformer pozwala również na wprowadzenie obliczeń równoległych. Ze względu na te kwestie modele oparte o mechanizmy uwagi przodują aktualnie w większości testów porównawczych (*benchmark*) NLP i NLG, powoli wypierając inne modele⁴².

Najbardziej znane przykłady modeli wykorzystujących architekturę Transformer do generowania języka to: GPT, BERT, XLNet, T5, BART.

40 Tamże.

41 Topal, M.O., Bas, A., Heerden, I.V. (2021). *Exploring Transformers in Natural Language Generation: GPT, BERT, and XLNet*. <https://arxiv.org/ftp/arxiv/papers/2102/2102.08036.pdf>

42 Tamże.



Rysunek 6: Przykłady modeli wykorzystujących architekturę Transformer

GPT

Model GPT (*generative pre-trained transformer*) to opracowany przez zespół z OpenAI model generatywny dla języka naturalnego, którego celem jest przewidzenie, jakie słowo wystąpi jako kolejne w niepełnym zdaniu, biorąc pod uwagę wszystkie poprzedzające słowa. Ten model językowy w swojej architekturze wykorzystuje bloki dekodera struktury Transformer. Mechanizm uwagi w GPT uwzględnia słowa, pary słów, pary par słów itd., ale tylko z sekwencji poprzedzającej bieżące słowo. Uzupełniona o nowe słowo sekwencja ponownie trafia do wejścia modelu i przewidywane jest kolejne słowo. Trwa to do czasu, aż zdania zostaną uzupełnione: model GPT generuje dalszy tekst w najlepszy i zgodny z tematem sposób, na podstawie otrzymanych na wejściu zdań.

Przed GPT modele NLP były trenowane na dużych ilościach odpowiednio anotowanych danych tekstowych, co było znaczącym ograniczeniem dla ich rozwoju. Powszechnie dostępne są głównie nieoznakowane dane, a ich

przygotowanie jest bardzo czasochłonne. Model GPT-1⁴³ z 2018 roku, został zbudowany przy użyciu nieoznakowanych danych i umożliwił użytkownikom dostrojenie modelu do wykonywania wybranych przez nich zadań. Dzięki uczeniu transferowemu (*transfer learning*) GPT-1 stał się potężnym narzędziem ułatwiającym wykonywanie zadań NLP i dającym możliwość ciągłego zwiększania jego potencjału w połączeniu z innymi modelami oraz większymi zbiorami danych i parametrów.

Następcami GPT-1 są GPT-2⁴⁴ oraz GPT-3⁴⁵. Nie tylko obejmują większe zbiory danych, ale też dodano do nich więcej parametrów w celu zbudowania silniejszych modeli językowych, zdolnych do lepszego modelowania i wnioskowania. GPT-2, który ma 1,5 miliarda parametrów, jest 10 razy większy od swojego poprzednika. Natomiast GPT-3 zawiera około 175 miliardów parametrów i jest 100 razy większy niż GPT-2. Obydwa modele, a zwłaszcza GPT-3, zostały wytrenowane z zastosowaniem znacznie większej niż pierwotny model GPT ilości danych zebranych w internecie.

GPT-3 uczył się na korpusie znanym jako Common Crawl, zawierającym około 500 miliardów słów w wypowiedziach pochodzących ze zróżnicowanych i wielopłaszczyznowych stron internetowych. Może automatycznie generować unikalne teksty, które są zarówno kreatywne, jak i poprawne kontekstowo (nowy fragment tekstu będący odpowiedzią na dowolny tekst). Dodatkowo brzmią prawie tak samo, jakby napisał je człowiek. Możliwości oferowane przez GPT-3 mogą zostać również wykorzystane do rozwiązania innych zadań NLG, takich jak tworzenie pytań i odpowiedzi, generowanie raportów czy kodu, przeszukiwanie dokumentów i wielu innych. GPT-3 jest narzędziem o ogromnym potencjale, ale ze względu na swoją złożoność i ogromny rozmiar sprawia, że jego zastosowanie jest bardzo kosztowne i obciąża dostępne zasoby. W swojej obecnej formie jest często wyzwaniem dla praktycznego zastosowania.

43 Radford, A., Narasimhan, K., Salimans, T., Sutskever, I. (2018). *Improving language understanding by generative pre-training*. https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf

44 Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I. (2018). *Language Models are Unsupervised Multitask Learners*. https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

45 Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D. (2020). *Language Models are Few-Shot Learners*. <https://arxiv.org/pdf/2005.14165.pdf>

Context (human-written): In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

GPT-2: The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

Pérez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez.

Pérez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them – they were so close they could touch their horns.

While examining these bizarre creatures the scientists discovered that the creatures also spoke some fairly regular English. Pérez stated, "We can see, for example, that they have a common 'language,' something like a dialect or dialectic."

Dr. Pérez believes that the unicorns may have originated in Argentina, where the animals were believed to be descendants of a lost race of people who lived there before the arrival of humans in those parts of South America.

While their origins are still unclear, some believe that perhaps the creatures were created when a human and a unicorn met each other in a time before human civilization. According to Pérez, "In South America, such incidents seem to be quite common."

However, Pérez also pointed out that it is likely that the only way of knowing for sure if unicorns are indeed the descendants of a lost alien race is through DNA. "But they seem to be able to communicate in English quite well, which I believe is a sign of evolution, or at least a change in social organization," said the scientist.

Rysunek 7: Przykład tekstu wygenerowanego przez model GPT-2⁴⁶

46 Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. Sutskever, I. (2018). *Language Models ...*, dz. cyt.

BERT

W 2018 roku naukowcy z Google AI opracowali model językowy BERT⁴⁷ (*bi-directional encoder representations from transformers*), wykorzystujący mechanizm kodera z architektury Transformer. BERT jest modelem uczonym na dwóch strategiach nienadzorowanych:

- MLM (*masked language model*) – usuwa ograniczenie jednokierunkowości i umożliwia przewidywanie zamaskowanych tokenów na podstawie zarówno poprzedników, jak i następników;
- NSP (*next sentence prediction*) – pozwala na wstępne trenowanie reprezentacji par tekstowych.

MLM umożliwia dwukierunkowe uczenie się z tekstu: losowo maskowane są słowa w zdaniu wejściowym, a zadaniem modelu jest przewidzenie brakujących słów na podstawie kontekstu, zarówno z prawej, jak i lewej strony. Podczas gdy MLM uczy BERT rozumienia relacji między słowami, NSP uczy rozumienia zależności między zdaniami poprzez przewidywanie, czy jest prawdopodobne, aby określone zdanie wystąpiło jako następne po zadanym zdaniu. BERT uczy się języka naturalnego na podstawie informacji syntaktycznych (relacje między słowami) oraz semantycznych (znaczenie słów). Ponieważ w trakcie trenowania brakujące słowa są przewidywane na podstawie innych słów zdania, reprezentacje wektorowe (*embedding*) dla słów są kontekstowe, czyli są odmienne w różnych zdaniach, w zależności od kontekstu ich występowania. BERT korzysta również z reprezentacji wektorowej całych zdań.

Model BERT został wytrenowany na ogromnej ilości danych internetowych z BooksCorpus (800 mln słów) oraz Wikipedii w języku angielskim (2500 mln słów). Aby stosować go w różnych aplikacjach, badacze dodają jedną lub kilka warstw sieci do wstępnie wytrenowanego modelu i trenują sieć do własnych zadań. Z powodu swojego rozmiaru (około 110 milionów parametrów w wersji bazowej) praktyczne zastosowanie BERT jest utrudnione. Dlatego powstało wiele mniejszych wersji do zastosowań komercyjnych, np. DistilBERT⁴⁸,

47 Devlin, J., Chang, M., Lee, K., Toutanova, K. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. <https://arxiv.org/pdf/1810.04805.pdf>

48 Sanh, V., Debut, L., Chaumond, J., Wolf, T. (2019). *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. <https://arxiv.org/pdf/1910.01108.pdf>

ALBERT⁴⁹, RoBERTa⁵⁰, HuBERT⁵¹. BERT stanowi również podstawę większości liczących się prób adaptacji modeli językowych na język polski. W rankingu KLEJ (Kompleksowa Lista Ewaluacji Językowych⁵²) na podium znalazły się takie modele jak HerBERT⁵³ oraz modele multijęzykowe XLM-RoBERTa⁵⁴ i Polish RoBERTa⁵⁵.

BERT można wykorzystać do rozwiązania wielu różnych zadań językowych: analizy sentymentu, generowania odpowiedzi, klasyfikacji tekstu, tworzenia podsumowań czy generowania treści (np. artykułu lub akapitu na dowolny temat), wprowadzając tylko kilka zdań początkowych.

XLNet

Biorąc pod uwagę zalety i wady autoregresyjnych (*autoregressive*, AR) modeli językowych oraz modelu BERT, zespół naukowców z Carnegie Mellon University i Google Brain opracował model XLNet⁵⁶. Ponieważ modele AR, mające na celu oszacowanie rozkładu prawdopodobieństwa korpusu tekstu poprzez predykcję kolejnego tokena na podstawie wszystkich poprzednich, biorą pod uwagę tylko kontekst jednokierunkowy (z prawej lub lewej strony), nie są zbyt skuteczne w zadaniach rozumienia języka, które wymagają kontekstu

-
- 49 Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R. (2020). *ALBERT: A Lite BERT for Self-supervised Learning of Language Representations*. <https://arxiv.org/pdf/1909.11942.pdf>
- 50 Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. <https://arxiv.org/pdf/1907.11692.pdf>
- 51 Hsu, W., Bolte, B., Tsai, Y.H., Lakhota, K., Salakhutdinov, R., Mohamed, A. (2021). *HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units*. <https://arxiv.org/pdf/2106.07447.pdf>
- 52 Rybak, P., Mroczkowski, R., Tracz, J., Gawlik, I. (2020). *KLEJ: Comprehensive Benchmark for Polish Language Understanding*. <https://arxiv.org/pdf/2005.00630.pdf>
- 53 Mroczkowski, R., Rybak, P., Wróblewska, A., Gawlik, I. (2021). *HerBERT: Efficiently Pretrained Transformer-based Language Model for Polish*. <https://arxiv.org/pdf/2105.01735.pdf>
- 54 Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V. (2020). *Unsupervised Cross-lingual Representation Learning at Scale*. <https://arxiv.org/pdf/1911.02116.pdf>
- 55 Dadas, S., Peretkiewicz, M., Poświata, R. (2020). *Pre-training Polish Transformer-Based Language Models at Scale*. <https://arxiv.org/pdf/2006.04229.pdf>
- 56 Yang, Z., Dai, Z., Yang, Y., Carbonell, J.G., Salakhutdinov, R., Le, Q.V. (2019). *XLNet: Generalized Autoregressive Pretraining for Language Understanding*. <https://arxiv.org/pdf/1906.08237.pdf>

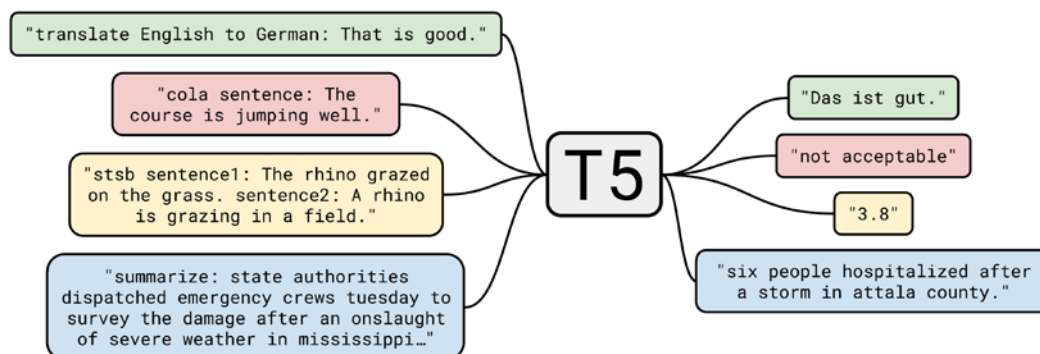
dwukierunkowego. Z kolei BERT, choć zapewnia możliwość modelowania kontekstów dwukierunkowych i lepszą wydajność, w porównaniu ze wstępnie trenowanymi modelami AR ma pewne ograniczenia. Model trenowany jest w celu rekonstrukcji zamaskowanych w tekście słów. Maskowanie odbywa się na wejściu, a model uczy się nie na rzeczywistych danych, ale na ich zmodyfikowanej wersji. Powoduje to rozbieżność między trenowaniem a dostrajaniem sieci. Co więcej, nie jest możliwe modelowanie łącznego rozkładu prawdopodobieństwa korpusu – BERT zakłada, że brakujące słowa są od siebie niezależne, co jest zbyt dużym uproszczeniem dla języka naturalnego.

XLNet łączy zdolność BERT do uczenia się kontekstów dwukierunkowych z uogólnioną autoregresyjną metodą wstępnego uczenia zapożyczoną z modelu Transformer-XL⁵⁷. XLNet, jako model AR, nie jest trenowany na zmodyfikowanym zbiorze danych, dzięki czemu unika ograniczeń BERT związanych z maskowaniem danych. Umożliwia również obliczenie łącznego prawdopodobieństwa przewidywanych słów, eliminując założenie niezależności przyjęte w BERT. XLNet uczy się dwukierunkowego kontekstu za pomocą mechanizmu PLM (*permutation language modeling*) poprzez trenowanie na wszystkich możliwych permutacjach słów w zdaniu. W tym celu XLNet maksymalizuje oczekiwany logarytm prawdopodobieństwa we wszystkich możliwych permutacjach wejściowej sekwencji. Aby ulepszyć architekturę sieci, XLNet integruje również mechanizm rekurencji segmentów ze schematem kodowania względnego architektury Transformer-XL. Ten model osiąga lepsze niż BERT wyniki w wielu zadaniach NLP, takich jak generowanie odpowiedzi, wnioskowanie w języku naturalnym (*natural language inference*) czy analiza sentymentu.

57 Dai, Z., Yang, Z., Yang, Y., Carbonell, J.G., Le, Q.V., Salakhutdinov, R. (2019). *Transformer-XL: Attentive Language Models beyond a Fixed-Length Context*. <https://arxiv.org/pdf/1901.02860.pdf>

T5

Model T5⁵⁸ (*text-to-text transfer Transformer*) został przedstawiony w 2020 roku przez zespół z Google. Sugeruje on przekształcenie wszystkich zadań NLP w ujednoczony format text-to-text, w którym dane wejściowe i wyjściowe są zawsze ciągami tekstowymi. Zastosowanie formatu text-to-text sprawia, że model T5 może być wykorzystany do wielu różnych zadań NLP przy użyciu tych samych hiperparametrów, funkcji straty i procedury dekodowania. Podejście można z powodzeniem zastosować do generowania odpowiedzi na pytania, tworzenia podsumowań abstrakcyjnych, rozwiązywania zadań klasyfikacyjnych, wnioskowania w języku naturalnym, a nawet w regresji. Aby umożliwić T5 wykonywanie różnych zadań, do oryginalnych danych wejściowych dodawany jest prefiks tekstowy określający zadanie, jakie ma wykonać model. Przykładowo przetłumaczenie zdania "That is good" z języka angielskiego na niemiecki wymaga sekwencji wejściowej "translate English to German: That is good". Prefiks tekstowy używany do określenia danego zadania jest zasadniczo uważany za hiperparametr.

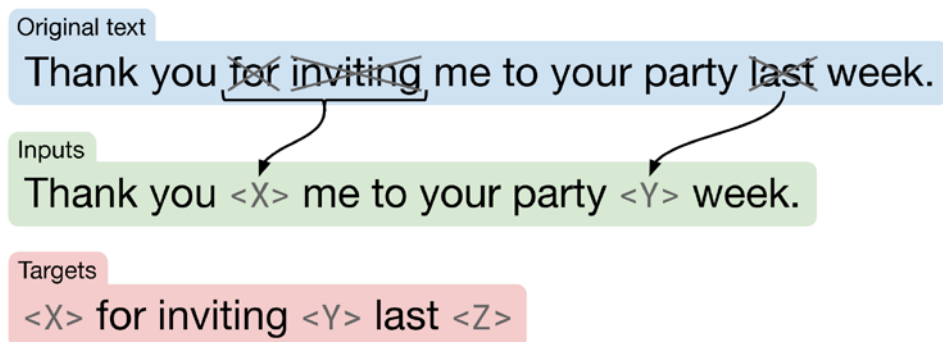


Rysunek 8: Schemat działania modelu T5 dla wybranych zadań (tłumaczenie, ocena językowa, podobieństwo zdań, streszczenie)⁵⁹

Model T5 bazuje na oryginalnej strukturze Transformer. Podobnie jak BERT, korzysta z MLM z niewielką modyfikacją: BERT każde kolejne zamaskowane słowa zastępuje osobnymi symbolami (np. tokenem <MASK>), natomiast T5 ciąg kolejnych zamaskowanych słów zastępuje tylko jednym symbolem.

58 Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J. (2020). *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. <https://arxiv.org/pdf/1910.10683.pdf>

59 Tamże



Rysunek 9: Schemat strategii uczenia w modelu bazowym T5⁶⁰

Dodatkowym wyróżnikiem T5 jest jego skala. Dostępnych jest pięć różnych rozmiarów tego modelu, każdy z inną liczbą parametrów: T5-small (60 mln parametrów), T5-base (220 mln parametrów), T5-large (770 mln parametrów), T5-3B (3 mld parametrów), T5-11B (11 mld parametrów). Modele te zostały wstępnie wytrenowane na około 1 bilionie tokenów. Nieoznakowane dane pochodzą ze zbioru C4 (*Colossal Clean Crawled Corpus*) zawierającego około 750 GB tekstu, który jest wyczyszczoną wersją zbioru Common Crawl (m.in. wyodrębniono tylko treści w języku angielskim, zastosowano deduplikację itp.).

BART

Autoenkoder odszumiający (*denoising autoencoder*) BART⁶¹ służy do trenowania modeli typu *sequence-to-sequence* (Seq2Seq), które biorą ciąg elementów (np. liter, słów, szeregów czasowych itp.) i przekształcają go na inny ciąg elementów. BART wykorzystuje standardową architekturę Transformer, a sami autorzy tego rozwiązania (Facebook AI, dziś Meta AI) mówią, że można je postrzegać jako uogólnienie BERT i GPT. W BART zastosowano dwukierunkowy koder i jednokierunkowy dekodek. Dane treningowe zawierają zniekształcone teksty (uszkodzone lub zaszumione), które model uczy się

⁶⁰ Tamże.

⁶¹ Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L. (2020). *BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension*. <https://arxiv.org/pdf/1910.13461.pdf>

rekonstruować. Wybór metody wprowadzania szumu (zniekształceń) jest całkowicie dowolny. Autorzy modelu rozważali m.in. maskowanie tokenów, usuwanie tokenów, permutację zdań, uzupełnianie tekstem (*text infilling*), gdzie fragmenty tekstu zastępowane są jednym tokenem, a także rotację dokumentów (tekst jest obracany tak, aby zaczynał się od losowo wybranego słowa).

BART jest szczególnie skuteczny, gdy jest dostrojony do generowania tekstu, ale działa również dobrze w zadaniach NLU. Model jest odpowiedni m.in. do tłumaczenia maszynowego, odpowiadania na pytania, tworzenia podsumowań tekstu (streszczeń lub parafraz długiego dokumentu tekstowego) oraz klasyfikacji sekwencji (zdań lub tokenów).

Generowanie danych tekstowych – wybrane zagadnienia

Niektóre z zadań, które realizują modele NLG, mają swoje uogólnienia bądź modyfikacje. Ich lista się wydłuża, ponieważ wciąż powstają nowe modele będące w stanie przetwarzać coraz bardziej abstrakcyjne aspekty. Warto krótko przeanalizować najpopularniejsze obecnie zadania NLG.

Generowanie data-to-text

Zamianę danych strukturalnych (np. tabele, grafy, wykresy) na tekst w języku naturalnym, który prawidłowo i płynnie opisuje te dane, umożliwia generowanie data-to-text (D2T)^{62 63}. Ta forma NLG zmagą się z dwoma wyzwaniami: wyborem odpowiedniego podzbioru danych wejściowych do omówienia (co powiedzieć) oraz wygenerowaniem treści (jak to powiedzieć). Tradycyjne sposoby rozwiązania problemu D2T obejmują kilka kolejnych etapów: od planowania tekstu (określenie, co należy powiedzieć), przez tworzenie zdań (agregowanie treści w zdania) po realizację językową (stosowanie odpowiedniej

62 Radev, D.R., Zhang, R., Rau, A., Sivaprasad, A., Hsieh, C., Rajani, N.F., Tang, X., Vyas, A., Verma, N., Krishna, P., Liu, Y., Irwanto, N., Pan, J., Rahman, F., Zaidi, A., Mutuma, M., Tarabar, Y., Gupta, A., Yu, T., Tan, Y.C., Lin, X.V., Xiong, C., Socher, R. (2021). *DART: Open-Domain Structured Data Record to Text Generation*. <https://arxiv.org/pdf/2007.02871.pdf>

63 Keymanesh, M., Benton, A., Dredze, M. (2022). *What Makes Data-to-Text Generation Hard for Pretrained Language Models?*. <https://arxiv.org/pdf/2205.11505.pdf>

składni, morfologii). Najnowsze podejścia opierają się na sieciach neuronowych, a w szczególności wykorzystują architekturę Transformer. Do zadania D2T używane są wstępnie wytrenowane modele językowe (*pre-trained language models*), które po dostrojeniu na odpowiednich do zadania danych, dobrze radzą sobie z generowaniem opisów relacji w języku naturalnym. Do tego celu zastosowano m.in. modele GPT-2 oraz T5.

Example #1

Input: <H> Aleksey Chirikov (icebreaker) <R> SHIP_BEAM <T> 21.2 <H> Aleksey Chirikov (icebreaker) <R> STATUS <T> ""In service"" <H> Aleksey Chirikov (icebreaker) <R> BUILDER <T> Helsinki <H> Aleksey Chirikov (icebreaker) <R> BUILDER <T> Arctech Helsinki Shipyard

Reference: The icebreaker Aleksey Chirikov was built at the Arctech Helsinki Shipyard in Helsinki. The Aleksey Chirikov (icebreaker) is in service and its ship beam is 21.2

Description: BUILDER means a person who constructs something

GPT2-XL (0-shot): <H> Aleksey Chirikov (icebreaker) <R> SHIP_BEAM <T> 21.2 <H> Aleksey Chirikov (icebreaker) <R> STATUS <T> "In service" <H> Aleksey Chirikov (icebreaker) <R> BUILDER <T> Helsinki <H> Aleksey Chirikov (icebreaker) <R> BUILDER <T>

GPT2-XL (3-shot): Aleksey Chirikov (icebreaker) is of type 21.2.

GPT2-XL (3-shot) + Description: Aleksey Chirikov (icebreaker) was a Russian icebreaker.

GPT2-XL-PT: Aleksey Chirikov (icebreaker) was built by Arctech Helsinki Shipyard.

GPT2-XL-PT + Description: Aleksey Chirikov (icebreaker) was built by Arctech Helsinki Shipyard.

GPT2-XL-PT + Reranking: Aleksey Chirikov (icebreaker) is in service at Arctech Helsinki Shipyard.

T5_{large}: The ship Aleksey Chirikov was built by Arctech, Helsinki Shipyard and has a ship beam of 21.2 metres. It is currently in service.

Rysunek 10: Przykład generowania data-to-text przy użyciu różnych modeli i strategii⁶⁴

Generowanie dialogów

Problem generowania dialogów sprowadza się do stworzenia inteligentnego systemu zdolnego do prowadzenia naturalnej rozmowy z człowiekiem. Wśród systemów dialogowych wyróżnia się systemy zorientowane zadaniowo (*task-oriented system*), które mają wspomagać użytkownika w wykonaniu określonych zadań (np. rezerwacja hotelu, wyszukiwanie produktów, rekomendacja restauracji), oraz systemy typu open-domain (*open-domain dialog systems*), których celem jest prowadzenie swobodnej rozmowy i dostarczanie odpowiednich reakcji i rozrywki, bez ograniczeń dotyczących tematów rozmów. Można wyróżnić dwa główne podejścia do zadania generowania dialogów. Pierwsze to metody oparte na wyszukiwaniu, uczące się wybierać właściwą odpowiedź z puli możliwości dostępnych w zgromadzonym repozytorium. Drugie podejście obejmuje metody generatywne, które mogą formułować odpowiedzi bezpośrednio

64 Tamże.

podczas rozmowy z użytkownikiem. Do popularnych modeli językowych generowania dialogów należą Seq2Seq, GPT oraz sieci GAN. Przykładami stosowanych modeli generatywnych są: DialoGPT⁶⁵, Meena⁶⁶, PLATO⁶⁷, Blender⁶⁸.

Conversation C	
Human:	<i>Haha, I think mathematics has a presence beyond human existence.</i>
Meena:	<i>So does philosophy.</i>
Human:	<i>Can you state with examples how this is true.</i>
Meena:	<i>Can you state how it isn't?</i>
Human:	<i>Without human existence there will be no one left to comprehend whats right whats wrong, and thus philosophy wi [sic]</i>
Meena:	<i>That's not the definition of philosophy.</i>
Human:	<i>Without human existence there will be no one left to comprehend whats right whats wrong, and thus philosophy will not be needed. How do you define philosophy?</i>
Meena:	<i>The study of general knowledge or the fundamental nature of things.</i>

Rysunek 11: Przykład rozmowy typu open-domain z systemem Meena⁶⁹

⁶⁵ Zhang, Y., Sun, S., Galley, M., Chen, Y., Brockett, C., Gao, X., Gao, J., Liu, J., Dolan, W. B. (2020). DIALOGPT: Large-Scale Generative Pre-training for Conversational Response Generation. <https://arxiv.org/pdf/1911.00536.pdf>

⁶⁶ Adiwardana, D., Luong, M.Y., So, D.R., Hall, J., Fiedel, N., Thoppilan, R., Yang, Z., Kulshreshtha, A., Nemade, G., Lu, Y., Le, Q.V. (2020). Towards a Human-like Open-Domain Chatbot. <https://arxiv.org/pdf/2001.09977.pdf>

⁶⁷ Bao, S., He, H., Wang, F., Wu, H., Wang, H. (2020). PLATO: Pre-trained Dialogue Generation Model with Discrete Latent Variable. <https://arxiv.org/pdf/1910.07931.pdf>

⁶⁸ Roller, S., Dinan, E., Goyal, N., Ju, D., Williamson, M., Liu, Y., Xu, J., Ott, M., Shuster, K., Smith, E.M., Boureau, Y., Weston, J. (2021). Recipes for Building an Open-Domain Chatbot. <https://arxiv.org/pdf/2004.13637.pdf>

⁶⁹ 69 Adiwardana, D., Luong, M.Y., So, D.R., Hall, J., Fiedel, N., Thoppilan, R., Yang, Z., Kulshreshtha, A., Nemade, G., Lu, Y., Le, Q.V. (2020). Towards ..., dz. cyt.

Generowanie pytań

Automatyczne tworzenie pytań (*question generation*, QG) w języku naturalnym na podstawie tekstu krótkiego (np. zdania) lub dłuższego (np. akapity) cieszy się coraz większym zainteresowaniem w obszarze NLG. Badania dotyczące QG standardowo skupiają się na tym, o co pytać (identyfikacja informacji, o które należy zapytać) oraz jak zapytać (wykorzystanie zidentyfikowanych informacji do tworzenia pytań zachowujących poprawność gramatyczną, precyzję semantyczną i elastyczność językową). W edukacji konstruowanie dobrych pytań ma kluczowe znaczenie dla oceny wiedzy uczniów i stymulowania samokształcenia. Płynne tworzenie pytań jest też ważną umiejętnością w systemach dialogowych (np. dla chatbotów), m.in. w inicjowaniu rozmów lub uzyskiwaniu określonych informacji od użytkowników. Większość modeli wykorzystuje strukturę Seq2Seq⁷⁰.

Generowanie odpowiedzi

Zadanie automatycznego odpowiadania na pytania (*question answering*, QA) jest częścią konwersacyjnej sztucznej inteligencji, która koncentruje się na generowaniu odpowiedzi w języku naturalnym na pytania zadawane przez użytkowników. Systemy QA wyodrębniają lub tworzą odpowiedź na podstawie kontekstu (np. tekst, tabela), w jakim jest zadawane pytanie. Ogólna idea zaproponowanych w ostatnich latach rozwiązań problemu QA jest dość spójna: gromadzone są duże zbiory danych treningowych obejmujące zestawy pytań, odpowiedzi i powiązanych kontekstów, na których są dostrajane i testowane wybrane modele językowe. Najnowsze podejścia wykorzystują architekturę Transformer, gdzie model uczy się łączyć cechy obecne w pytaniach z cechami zawartymi w kontekście, za pomocą mechanizmu uwagi⁷¹.

70 Pan, L., Lei, W., Chua, T., Kan, M. (2019). *Recent Advances in Neural Question Generation*. <https://arxiv.org/pdf/1905.08949.pdf>; Liu, B., Wei, H., Niu, D., Chen, H., He, Y. (2020). *Asking Questions the Human Way: Scalable Question-Answer Generation from Text Corpus*. <https://arxiv.org/pdf/2002.00748.pdf>

71 Yin, J., Jiang, X., Lu, Z., Shang, L., Li, H., Li, X. (2016). *Neural Generative Question Answering*. <https://www.ijcai.org/Proceedings/16/Papers/422.pdf>; Andreas, J., Rohrbach, M., Darrell, T., Klein, D. (2016). *Learning to Compose Neural Networks for Question Answering*. <https://arxiv.org/pdf/1601.01705.pdf>

Parafrazowanie

Automatyczne tworzenie parafraz jest ważnym, ale trudnym zadaniem NLG. Polega na przekształceniu danego zdania w języku naturalnym w nowe zdanie o tym samym znaczeniu semantycznym, ale z zastosowaniem innej formy syntaktycznej lub leksykalnej. Tradycyjne podejścia do generowania parafraz obejmują metody oparte na regułach, tezaurusach, statystycznym tłumaczeniu maszynowym (*statistical machine translation*, SMT) oraz metody gramatyczne. Obecnie problem generowania parafraz interpretuje się jako problem trenowania modeli Seq2Seq⁷².

Figure 3: Examples of the generated paraphrases by different models on Quora-II.

Input	Pointer-generator	RL-ROUGE	RbM-SL	RbM-IRL	Reference
where we go after death ?	what are the chances of getting death in death ?	what are the chances of getting death in go after death ?	where can i go after death ?	where can i go after death ?	where do we go to when we die ?
what can i do to make \$ 2000 in one week ?	should i make \$ 2000 and week ?	how can i make a 2000 rupee website ?	how do i make \$ 2000 one week ? what are some tips ?	how can i make 2000 rupee a week ?	how do i make \$ 2000 a week ?
what is the difference between scripting language , programming language and network protocol ?	what is the [UNK] scripting programming language and a scripting language scripting ?	what is the difference between programming languages and network language and network ?	what is the difference between scripting languages and programming language ?	what is the difference between programming languages and network protocol ?	what is the difference between a programming language and a scripting language ?
which books can change your life ?	which is the best books for quora ?	which is the one beautiful books to change your friends ?	which is the best book to change our life ?	which is the one thing should change my life ?	what are your top 5 non-fiction books ? something that can change my life ?
why is donald trump still 'ducking' his income tax return issue ?	why does donald trump [UNK] us [UNK] [UNK] ?	why did trump 's victory [UNK] as issue to [UNK] ?	why is donald trump still 's income tax return ?	why did trump deal tax issue ?	why is trump refusing to release his tax return ?



Rysunek 12: Przykłady generowanych parafraz przy użyciu różnych modeli⁷³

Generowanie historii

Automatyczne tworzenie fikcyjnych historii (*story generation*) to problem od dawna rozpatrywany przez badaczy sztucznej inteligencji. Kreatywny system, który potrafiłby budować spójne, płynne i kompletne treści, byłby doskonały

⁷² Li, Z., Jiang, X., Shang, L., Li, H. (2018). *Paraphrase Generation with Deep Reinforcement Learning*. <https://aclanthology.org/D18-1421.pdf>

⁷³ Tamże.

do sprawdzenia, czy AI naprawdę coś rozumie. Tworzenie kreatywnych tekstów wymaga nie tylko przedstawienia historii, ale również jej zrozumienia, dlatego problem łączy w sobie elementy zarówno NLG, jak i NLU. Do generowania historii używa się najnowocześniejszych modeli językowych, takich jak GPT. Kluczowym wyzwaniem związanym z zastosowaniem wstępnie wytrenowanych modeli jest zapewnienie, że wygenerowana treść nie tylko pozostanie na temat, ale też zachowa spójność jednostek i wydarzeń. Początkowe rozwiązania do problemu generowania historii nie angażowały systemów uczących. Tworzenie fabuł opierało się na bazach wiedzy zawierających ręcznie kodowane struktury wiedzy. Przejście na rozwiązania oparte na sieciach neuronowych co prawda umożliwiło generowanie treści o szerszym zakresie tematów (na podstawie wiedzy pozyskiwanej z ogromnych korpusów treningowych), ale wpłynęło na zmniejszenie koherencji w generowanych historiach. Chociaż GPT-3 ma duży potencjał pozwalający na rozwiązanie tego problemu, generowanie historii wciąż uznaje się za obszar sztucznej inteligencji otwarty na nowe pomysły⁷⁴.

Generowanie obrazów

Koncepcja wykorzystania systemów AI do tworzenia obrazów na podstawie innych obrazów lub z różnych typów danych (np. tekst, graf, szkic, określony układ obiektów) jest od dawna obecna w dziedzinie *computer vision*. Jednocześnie stanowi jedno z jej najtrudniejszych wyzwań. Istotny rozwój technik głębokiego uczenia i sztucznej inteligencji umożliwił opracowanie wielu rozwiązań zdolnych do syntetycznego generowania obrazów zawierających określoną treść, co tworzy szerokie spektrum możliwości i stawia wiele nowych wyzwań zarówno badawczych, jak i społecznych.

Modelowanie generatywne danych obrazowych zasadniczo polega na wykorzystaniu modelu do generowania nowych przykładów, pochodzących z wyuczonych rozkładów próbek. Jednym z wariantów może być stosowanie techniki tzw. super rozdzielczości (*super-resolution*), w której model szkolony jest w celu zwiększenia rozdzielczości obrazu z zachowaniem jego szczegółów. Chociaż super

74 Fan, A., Lewis, M., Dauphin, Y. (2018). *Hierarchical Neural Story Generation*. <https://arxiv.org/pdf/1805.04833v1.pdf>; Riedl, M. (2021, 21 sierpnia). *An Introduction to AI Story Generation*. The Gradient. Pobrano 22.08.2022 z <https://the-gradient.pub/an-introduction-to-ai-story-generation/>

rozdzielczość ma wiele zastosowań, od renowacji starych fotografii, po ulepszenie systemów obrazowania medycznego, nie jest typowym zadaniem kojarzonym z generowaniem obrazów. Typowym skojarzeniem jest bowiem generowanie warunkowe, gdzie obraz tworzony jest tak, aby pasował do konkretnej, wskazanej przez użytkownika etykiety lub klasy.

Istnieją różne podejścia do rozwiązania tego problemu. Większość opiera się na głębokich sieciach generatywnych, takich jak VAE (autoenkodery wariacyjne) i GAN (generatywne sieci przeciwstawne), wyróżniające się tworzeniem różnorodnych obrazów wysokiej jakości. Naukowcy coraz częściej sięgają jednak również po architekturę Transformer oraz modele dyfuzji. Generatory typu text-to-image wzbudzają zainteresowanie zarówno badaczy, jak i opinii publicznej. Systemy takie jak DALL-E od OpenAI i Imagen od Google są w stanie utworzyć fotorealistyczne obrazy zacierające granicę między obrazami rzeczywistymi i syntetycznymi. Analizując rozwój technik generacyjnych, można nawet zaryzykować stwierdzenie, że w ciągu najbliższych kilku lat zdjęcie o tej samej szczegółowości i jakości łatwiej będzie wygenerować niż zrobić. Wynika to m.in. z przyspieszającego ostatnio rozwoju wspomnianych wyżej modeli.

Postęp w dziedzinie generowania obrazów wiąże się już nie tylko z zasobami obliczeniowymi, ale też z dostępnością i jakością danych treningowych. Jeśli dane uczące nie będą właściwie reprezentować rozkładu danych, które mają być objęte modelem, to bez względu na potencjał modelu, jego wydajność końcowa będzie niezadowalająca. Choć jesteśmy coraz bliżej modeli do poprawnego rozwiązywania problemów o wiedzy zerowej (*zero-knowledge problems*, np. *zero-shot classification*), nie mogą one być odpowiednio wyuczone bez dużych ilości danych treningowych. W odpowiedzi na problemy niedoboru danych syntetyczne obrazy generowane za pomocą omawianych modeli stały się obiecującą perspektywą dla uzupełniania, dywersyfikacji, a nawet tworzenia reprezentatywnych zbiorów danych, odpowiadając na wymagania dotyczące nie tylko ilości, lecz również jakości i realistyczności próbek.

Metody generowania danych obrazowych

Modele generujące dane obrazowe różniące się pod względem budowy i architektury. Ponadto różne ich odmiany ukierunkowane są na rozwiązywanie szerokiego spektrum problemów i zastosowań. Popularnością cieszą się zwłaszcza sieci generatywne i autoenkodery, mające szeroki wachlarz

zastosowań przy tworzeniu obrazów, jednak w niezmienionej formie nie są w stanie zapewnić zbytnej kontroli nad generowanymi obrazami. Podczas gdy przy użyciu wielu warstw splotowych możliwy jest transfer nawet bardzo abstrakcyjnych cech, wymagają przy tym dużej bazy obrazów referencyjnych i nie są w stanie powiązać więcej niż dwóch konceptów. Odpowiedzią na te problemy są większe modele wykorzystujące strukturę Transformer oraz, zyskujące coraz większe zainteresowanie, modele dyfuzji. W dalszej części przedstawione zostały najbardziej reprezentatywne architektury modeli generacyjnych wraz z przykładami oraz zastosowaniami w różnych typach zadań.

Autoenkodery wariacyjne (VAE)

Rodzajem modelu generatywnego zdolnego do syntetycznego generowania nowych danych, przypominających te zawarte w zbiorze treningowym jest autoenkoder wariacyjny (*variational autoencoder*, VAE). Podstawowa architektura autoenkodera składa się z dwóch elementów: kodera i dekodera. Koder pobiera dane wejściowe (obrazy) i generuje ich reprezentacje (kodowanie) w znacznie mniejszej przestrzeni wektorowej, tzw. przestrzeni ukrytej (*latent space*). Dekoder uczy się rekonstruować oryginalne dane wejściowe, bazując na zawartych w przestrzeni ukrytej kodowaniach. Ogólna idea mechanizmu działania autoenkoderów jest dość prosta i polega na nauczaniu najlepszego schematu kodującego-dekodującego (*the best encoding-decoding scheme*) przy użyciu iteracyjnego procesu optymalizacji: w każdej iteracji dane wyjściowe porównywane są z wejściowymi, a wagi są aktualizowane za pomocą algorytmu propagacji wstecznej. Model VAE wykorzystuje podstawową architekturę autoenkoderów, ale wprowadza pewną modyfikację, tak aby zamiast zwykłej rekonstrukcji możliwe było tworzenie nowych obrazów. Dane wejściowe są kodowane jako parametry wektora zmiennych losowych: koder nie tworzy bezpośrednio kodowania obrazu, ale zwraca wektor średnich i wektor odchyłeń standardowych, czyli niejako koduje obraz jak rozkład nad przestrzenią ukrytą. Dzięki temu, że dane wejściowe są kodowane w formie rozkładów możliwa jest regularyzacja przestrzeni ukrytej. Kodowania próbkowane są losowo z wybranego rozkładu (standardowo jest to rozkład Gaussa) określonego przez specyficzne dla tego rozkładu parametry (np. średnia i odchylenie standardowe), co oznacza, że nawet dla tych samych danych wejściowych, podczas gdy parametry rozkładu pozostają takie same, dane wyjściowe będą się nieco różnić na skutek próbkowania⁷⁵.

75 Kingma, D.P., Welling, M. (2014). *Auto-Encoding Variational Bayes*. <https://arxiv.org/pdf/1312.6114v10.pdf>

Z uwagi na rzeczoną regularyzację, tradycyjne autoenkodery nie sprawdzają się najlepiej przy generowaniu wielu szczegółów na obrazie. Niemniej jednak prostota modeli opartych na tej strukturze sprawia, że można je szybciej wytrenować. Są dobrym wyborem przy zastosowaniach mających eliminować lub redukować skomplikowanie obrazów. Ponadto zastosowanie rozkładów pozwala na dobre wykrywanie anomalii w zbiorze obrazów bądź na obrazie. Autoenkodery stosowane są ponadto w połączeniu z innymi technikami – np. generatywnymi sieciami przeciwstawnymi (VAE-GAN) – oraz jako elementy bardziej złożonych struktur głębokich sieci modeli.

Generatywne sieci przeciwstawne (GAN)

Struktura GAN⁷⁶ (*generative adversarial networks*), po raz pierwszy opisana przez Goodfellow i in. (2014), szybko została uznana za model z dużym potencjałem do generowania obrazów syntetycznych. Od tamtego czasu jest ciągle aktywnie rozwijana. Podstawowy mechanizm działania GAN obejmuje dwie sieci neuronowe: generator i dyskryminator, które konkurują ze sobą w grze o sumie zerowej (*zero-sum game*), gdzie zysk jednego modelu oznacza przegraną drugiego. Generator uczy się tworzyć nowe, wiarygodne przykłady obrazów. Pobiera wektor z wybranego rozkładu losowego (np. rozkładu Gaussa) i generuje z niego obrazy próbne, które stają się fałszywymi przykładami treningowymi dla dyskryminatora. Próby utworzone przez generator oraz rzeczywiste dane trafiają do dyskryminatora, który uczy się klasyfikować dostarczane mu przykłady jako prawdziwe (obrazy rzeczywiste) lub fałszywe (obrazy wygenerowane). W początkowej fazie uczenia generator tworzy ewidentnie sztuczne dane i dyskryminator szybko uczy się je rozpoznawać. Jednak w kolejnych iteracjach parametry generatora są aktualizowane na podstawie tego, jak dobrze sztuczne obrazy oszukały dyskryminator, natomiast parametry dyskryminatora są aktualizowane tak, aby lepiej rozróżniać obrazy rzeczywiste od fałszywych. Generator i dyskryminator konkurują ze sobą i doskonalą się dzięki tzw. uczeniu przeciwstawnemu (*adversarial learning*). W miarę postępów dyskryminator staje się coraz gorszy w odróżnianiu obrazów sztucznych od rzeczywistych, a jego dokładność spada. Jeżeli dyskryminator jest zmuszony do zgadywania, tzn. jest oszukiwany w około połowie przypadków, oznacza to, że generator tworzy doskonałe imitacje obrazów, a sieć GAN osiągnęła tzw. równowagę Nasha (*Nash equilibrium*). Chociaż jest to sytuacja jak najbardziej pożądana, nie

76 Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y. (2014). *Generative Adversarial Networks*. <https://arxiv.org/pdf/1406.2661.pdf>

ma gwarancji, że zostanie kiedykolwiek osiągnięta. Standardowa technika trenowania modeli typu GAN opiera się na optymalizacji gradientu w wielowymiarowej przestrzeni rzeczywistej (*gradient descent-ascent, GDA*). Technika ta, mimo szerokiego zastosowania i osiągnięcia dobrych wyników ma kilka problemów, np. brak stabilności modelu czy katastroficzne zapomnienie (*catastrophic forgetting*) znane z technik optymalizacji podstawowych głębokich sieci neuronowych. Najbardziej charakterystycznym dla modeli GAN problemem jest jednak *mode collapse*⁷⁷. Polega on na doprowadzeniu modelu do sytuacji, w której generator produkuje niewielką ilość próbek z powodu nieznacznie różniących się wartości mediany.

Od roku 2014 naukowcy zaproponowali wiele ulepszonych struktur i nowych możliwości zastosowań sieci GAN. Ważnym wariantem jest cGAN⁷⁸ (*conditional generative adversarial network*), czyli rozszerzenie GAN o możliwość warunkowego generowania obrazów. W sieciach cGAN zarówno generator, jak i dyskryminator są uzależnione (uwarunkowane) od pewnego rodzaju informacji pomocniczych, które są dodatkowymi danymi wejściowymi do modelu. Informacją pomocniczą może być wartość klasy, np. mężczyzna lub kobieta w przypadku generowania twarzy lub cyfra, gdy generowany jest obraz odręcznie pisanych cyfr. W ten sposób możliwe jest kontrolowanie obrazów tworzonych przez generator, tzn. na wejściu możliwe jest określenie etykiety obrazu, który chcemy wygenerować. Inną rozpowszechnioną strukturą jest głęboka splotowa sieć GAN (*deep convolutional generative adversarial network, DCGAN*⁷⁹), umożliwiająca tworzenie większych obrazów o wyższej jakości. Takie modele są ostatnimi czasy bardzo popularne w dziedzinie super rozdzielczości, pozwalając na poprawę jakości i rozmiarów zdjęć dwu-, a nawet czterokrotnie.

77 Durall, R., Chatzimichailidis, A., Labus, P., Keuper, J. (2021). *Combating Mode Collapse in GAN training: An Empirical Analysis using Hessian Eigenvalues*. <https://arxiv.org/pdf/2012.09673.pdf>

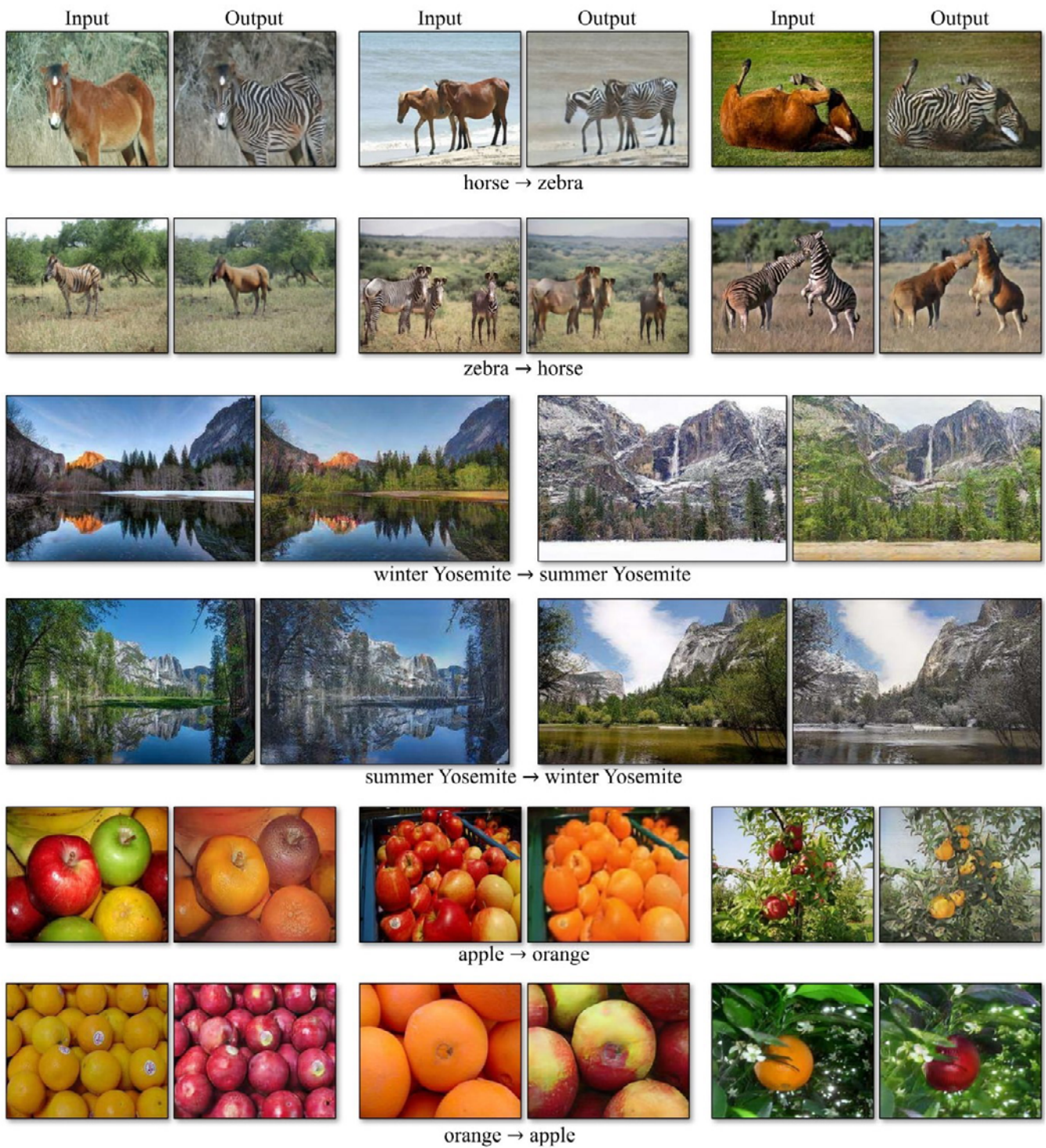
78 Mirza, M., Osindero, S. (2014). *Conditional Generative Adversarial Nets*. <https://arxiv.org/pdf/1411.1784.pdf>

79 Radford, A., Metz, L., Chintala, S. (2016). *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. <https://arxiv.org/pdf/1511.06434.pdf>

CycleGAN

Z myślą o rozwiązaniu problemu tłumaczenia obrazu na obraz (*image-to-image translation*, I2I), czyli przenoszeniu obrazu z jednej domeny do innej, powstał CycleGAN⁸⁰ (*cycle generative adversarial network*). Większość badań nad rozwiązaniem I2I czerpała z metod uczenia nadzorowanego, gdzie model ma zagwarantowany dostęp do par odpowiadających sobie obrazów z domen, pomiędzy którymi ma powstać przekształcenie. Jednak uzyskanie takiego zbioru treningowego nie zawsze jest możliwe, zwłaszcza wtedy, gdy pożądany wynik jest złożony pod względem oczekiwanych walorów artystycznych. CycleGAN uczy się bez konieczności użycia sparowanych przykładów treningowych. Na wejściu sieć otrzymuje dwa zbiory obrazów, każdy reprezentujący cechy jednej z domen (dziedzin), które nie muszą bezpośrednio ze sobą korespondować. Zakładając, że istnieje pewien związek między domenami, model stara się nauczyć tego związku, przechwytyjąc cechy charakterystyczne obrazów reprezentujących domenę źródłową i przekształcając je na cechy charakterystyczne domeny wyjściowej. Dla dwóch domen X i Y , CycleGAN uczy się dwóch odwzorowań: $G: X \rightarrow Y$, $F: Y \rightarrow X$, które powinny być swoimi odwzorowaniami odwrotnymi oraz bijekcjami, czyli funkcjami wzajemnie jednoznaczными. Odwzorowania G i F to generatory znane z GAN: ich zadaniem jest efektywne przekształcanie obrazów z dziedziny X na Y i odwrotnie. Każdemu z nich odpowiada dyskryminator próbujący odróżnić syntetyczne obrazy od rzeczywistych. W trakcie uczenia sieci CycleGAN zastosowanie standardowych procedur optymalizacji prowadzi do problemu *mode collapse* (opisany w sekcji poświęconej modelom GAN). Aby go uniknąć, stosowana jest metryka optymalizacji zapewniająca bijekcyjność i wzajemną odwrotność G oraz F – *cycle consistency loss*. Polega ona na optymalizacji cyklicznej zależności $F(G(x)) \approx x$ oraz $G(F(y)) \approx y$. W połączeniu z metryką przeciwstawną (*adversarial loss*) używaną w bazowych modelach GAN oraz tą stosowaną do optymalizacji dyskryminatora, *cycle consistency loss* daje najlepsze wyniki do trenowania generatywnych sieci cyklicznych.

80 Zhu, J., Park, T., Isola, P., Efros, A. (2017). *Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks*. <https://arxiv.org/pdf/1703.10593v7.pdf>



Rysunek 13: Przykład tłumaczenia obrazu na obraz za pomocą CycleGAN ⁸¹

81 Tamże.

Możliwość tłumaczenia obrazu na obraz niesie za sobą prawie nieograniczoną liczbę zastosowań. Oprócz wspomnianych już modeli zwiększających rozmiar, jakość lub punkt ostrości obrazu (super rozdzielczość, *photograph enhancement*) CycleGAN może w pełnym bądź ograniczonym zakresie rozwiązywać problemy typu: *style transfer* (przenoszenie stylu w oparciu o parę obrazów), *season transfer* (zmiana pory roku), czy też *object transfiguration* (zamiana koloru/tekstury/konturu obiektu na obrazie, np. zamiana zebry na konia czy pomarańcza na jabłko). Z uwagi na swoją strukturę CycleGAN znajdzie również zastosowanie w zamianie malowanych obrazów (pejzaży) na zdjęcia (*photograph generation from paintings*).

StyleGAN

Wersją sieci GAN wykorzystującą w generatorze techniki przenoszenia stylów (*style transfer*) jest StyleGAN⁸², opracowany przez zespół Nvidia w 2018 roku. Powstały model może generować imponująco realistyczne zdjęcia wysokiej jakości, zapewniając kontrolę nad stylem obrazu na różnych poziomach: od ogólnych (np. osoba dorosła lub dziecko) do szczegółowych cech zdjęcia (np. piegi lub kolor włosów). Generator StyleGAN składa się z dwóch sieci: sieci mapującej (*mapping network*) oraz sieci syntetyzującej (*synthesis network*). Sieć mapująca przenosi kodowanie obrazów na wektor, który następnie podlega wielu przekształceniom afinicznym. W ten sposób tworzone są wektory określające wiele stylów generowanego obrazu, które trafiają do sieci syntetyzującej odpowiedzialnej za tworzenie nowych obrazów. Sieć syntetyzująca przetwarza dane wejściowe poprzez wiele warstw splotowych. Dodatkowo do wszystkich wyjść warstw splotowych dodawany jest pewien szum, niezależnie od kodowań. Dzięki dodawaniu szumu oraz zastosowaniu technik splotowych, model StyleGAN generuje bardzo realistyczne zdjęcia. W porównaniu z poprzednikami innych modeli opisanych wcześniej, zdecydowanie lepiej łączy bardziej abstrakcyjne cechy dwóch obrazów, tworząc wiarygodne połączenia. Skutecznie wykorzystywany jest nie tylko do tradycyjnego transferu stylów, tj. zamiany stylu fotografii na styl wybranego malarza, lecz również do bardziej wysokopoziomowego scalania cech zdjęć. Przykładem może być łączenie rysów twarzy dwóch osób, co oryginalny model zaprezentowany przez Nvidia potrafi wykonać na różnych poziomach abstrakcji.

82 Karras, T., Laine, S., Aila, T. (2019). A Style-Based Generator Architecture for Generative Adversarial Networks. <https://arxiv.org/pdf/1812.04948.pdf>



Rysunek 14: Przykłady obrazów wygenerowanych przez StyleGAN⁸³

83 Tamże.

Transformery

Na architekturze podobnej do autoenkoderów i innych modeli generatywnych opierają się transformery, które w swojej budowie mają dekodery i kodery. Główną różnicą i cechą je wyróżniającą jest mechanizm uwagi (*attention mechanism*), czyli zależności między poszczególnymi wartościami wektorów. Modele te zyskały popularność w 2017 roku, jednak do niedawna stosowane były jedynie dla danych tekstowych (jak opisano w części poświęconej generowaniu danych tekstowych). Niemniej wraz ze wzrostem ich popularności znaleziono sposób na przeniesienie danych obrazowych do przestrzeni tekstowej, co otworzyło wiele nowych możliwości i kierunków badań. Aktualnie modele oparte na mechanizmie uwagi konkurują z sieciami splotowymi o pierwsze miejsca w popularnych testach porównawczych (*benchmark*) dla danych obrazowych. Umożliwiają również połączenie danych obrazowych i tekstowych w jednym modelu, rozszerzając obie dziedziny⁸⁴.

Image GPT (iGPT)

Sukces modeli generatywnych dla języka naturalnego skłonił naukowców do sprawdzenia, czy architektura wykorzystywana do uzupełniania języka naturalnego może być również z powodzeniem zastosowana do rozwiązania problemu uzupełniania obrazów. Rezultatem tych rozważań był zaprezentowany w 2020 roku przez OpenAI model Image GPT (iGPT). Zbudowany został na strukturze Transformer i zamiast na sekwencjach tekstu operuje na sekwencjach pikseli. Zdolny jest do generowania spójnych uzupełnień obrazów po podaniu początkowych pikseli.

iGPT przekształca obrazy, zmniejszając ich rozdzielczość i reprezentując je w postaci sekwencji jednowymiarowych wektorów. Następnie wykorzystuje jeden ze wstępnie wytrenowanych modeli językowych: BERT (do przewidywania zamaskowanych pikseli) lub GPT (do autoregresyjnego przewidywania kolejnych pikseli). Utworzone reprezentacje oceniane są za pomocą próbkowania liniowego (*linear probing*) lub metody dostrajania (*fine-tuning*). Pierwsza używa wstępnie wytrenowanego modelu językowego do ekstrakcji cech. Dla danej pary (X, Y) , gdzie X jest obrazem, a Y etykietą, tworzone

84 Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, L., Polosukhin, I. (2017). *Attention ...*, dz. cyt.

są reprezentacje cech F_X , które wraz z Y są przekazywane do klasyfikatora liniowego, np. regresji logistycznej. Druga metoda polega na precyzyjnym dostrójeniu modelu do konkretnego zadania, w tym przypadku do klasyfikacji obrazów. Stosując te podejścia, autorzy wykazali, że iGPT jest zdolny do wychwytywania istotnych cech obrazów⁸⁵.

Chociaż iGPT zaprezentowało niesamowite wyniki przy małej ilości informacji podawanych do sieci, to ma swoje ograniczenia, a najważniejszym z nich jest jego mała wydajność pod względem obliczeniowym. Jak jednak przekonują autorzy modelu, iGPT jest jedynie dowodem słuszności koncepcji, że techniki generowania sekwencyjnego są uniwersalną metodą, a modele językowe wykorzystujące strukturę Transformer mogą być z sukcesem stosowane do nie nadzorowanego uczenia się reprezentacji cech w nowych dziedzinach, w tym przypadku do obrazów.

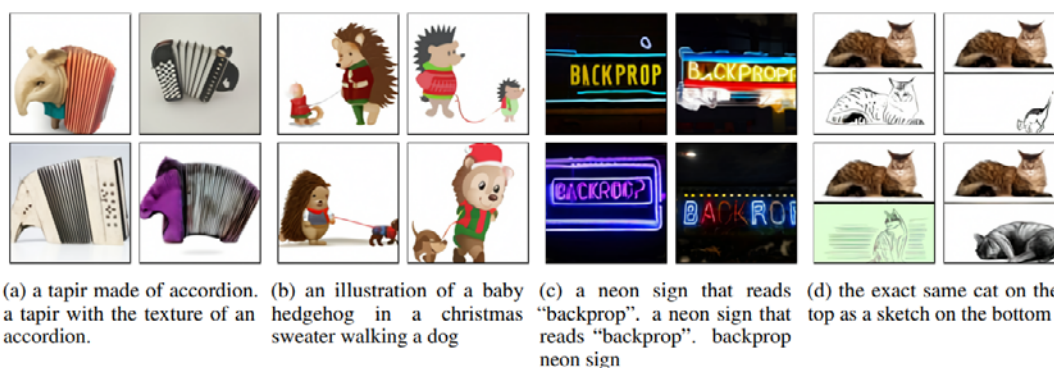
DALL-E

W 2021 roku zespół OpenAI zaprezentował model DALL-E⁸⁶, który jest zdolny do generowania obrazów z opisów tekstowych (*text-to-image-generation*, T2I). Jak mówią twórcy, model DALL-E jest wersją GPT-3, zawierającą 12 miliardów parametrów, wytrenowaną na 250 milionach par obrazów i tekstów, pochodzących ze zróżnicowanych i wielopłaszczyznowych stron internetowych. DALL-E przyjmuje tekst jako dane wejściowe i generuje obrazy zgodnie z jego zrozumieniem. Podobnie jak GPT-3, model DALL-E jest modelem językowym zbudowanym na strukturze Transformer, wykorzystującym bloki dekodera, ale zawiera również elementy dyskretnego autoenkodera wariacyjnego (*discrete variational autoencoder*, dVAE). Celem dVAE jest staranne przekształcanie obrazów w skompresowaną przestrzeń ukrytą. Transformer analizuje opisy tekstowe, uczy się korelacji istniejących między językiem i reprezentacją obrazu, a następnie odpowiada nie słowami, jak było w przypadku GPT-3, ale obrazami. Wszystkie pary obrazów i tekstów przekazywane są do dekodera jako pojedynczy strumień danych: sekwencje będące kombinacją słów i pikseli. Wytrenowany model po otrzymaniu opisu tekstowego na

85 Chen, M., Radford, A., Wu, J., Jun, H., Dhariwal, P., Luan, D., Sutskever, I. (2020). *Generative Pretraining From Pixels*. https://cdn.openai.com/papers/Generative_Pretraining_from_Pixels_V2.pdf

86 Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I. (2021). *Zero-Shot Text-to-Image Generation*. <https://arxiv.org/pdf/2102.12092.pdf>

wejściu tworzy zbiór 512 obrazów, z którego wybierane są najlepsze za pomocą opracowanego przez OpenAI modelu CLIP⁸⁷ (łączy znajomość pojęć anglojęzycznych z semantyczną znajomością obrazów). Oprócz generowania DALL-E jest również zdolny m.in. do: kontrolowania atrybutów obrazów, jednoczesnego rysowania wielu obiektów i kontrolowania atrybutów oraz ich relacji przestrzennych, przedstawiania perspektywy i trójwymiarowości, łączenia niepowiązanych pojęć, w tym antropomorfizacji zwierząt i obiektów, przedstawiania koncepcji związanych z przestrzenią (fakty geograficzne, punkty orientacyjne, okolice) i czasem (określone ramy czasowe).



Rysunek 15: Przykłady obrazów wygenerowanych przez DALL-E⁸⁸

W 2022 roku zespół OpenAI zaprezentował kolejną iterację projektu zajmującego się problemem generowania obrazów z opisów tekstowych: DALL-E 2. W porównaniu ze swoim poprzednikiem DALL-E 2 obiecuje zdolność do tworzenia bardziej realistycznych obrazów, z większą dokładnością oraz znacznie większą rozdzielczością. Zawdzięcza to m.in. większej i lepiej opisanej bazie zdjęć. Ponadto model ma ulepszoną i większą architekturę sieci. Nowa wersja zdecydowanie lepiej radzi sobie z generowaniem obrazów zawierających czytelny tekst, chociaż w tym aspekcie pozostaje jeszcze wiele do poprawy. Niemniej jednak zdecydowana większość zdjęć generowanych przez model jest trudna do odróżnienia od rzeczywistych. Zbliżone wyniki uzyskują również inne modele dyfuzji.

⁸⁷ Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M. (2022). *Hierarchical Text-Conditional Image Generation with CLIP Latents*. <https://arxiv.org/pdf/2204.06125.pdf>

⁸⁸ Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I. (2021). *Zero-Shot ...*, dz. cyt.

Modele dyfuzji

Modele generatywne wykorzystujące w swoich strukturach metodę dyfuzji (*diffusion models*) zyskały w ostatnim czasie dużą popularność w większości zadań związanych z generowaniem obrazów, w szczególności z opisów tekstowych. W zakresie syntezy obrazów wysokiej jakości oraz syntezy abstrakcyjnych pojęć na obrazach systemy wykorzystujące metodę dyfuzji są w stanie konkurować z wynikami systemów zbudowanych na generatywnych sieciach przeciwstawnych (GAN) i autoenkoderach wariacyjnych (VAE), a nawet je przewyższać.

Idea działania modeli dyfuzji jest następująca: jeżeli możliwe jest zbudowanie modelu zdolnego do nauczenia się systematycznego rozpadu informacji, to powinno być możliwe odwrócenie tego procesu i odzyskanie informacji z szumu. Zgodnie z tym założeniem system wykorzystujący metodę dyfuzji będzie generował nowe dane z czystego szumu poprzez jego stopniowe odszumianie, aż do uzyskania satysfakcjonującego wyniku. Model składa się z dwóch procesów: procesu dyfuzji, w którym następuje progresywne uszkodzenie danych poprzez dodawanie szumu Gaussa, oraz procesu odwrotnego – tzw. odwróconej dyfuzji (*reverse diffusion process*), gdzie szum jest przekształcany z powrotem w próbkę z rozkładu docelowego. Proces dyfuzji sformułowany jest jako łańcuch Markowa, a jego celem jest modelowanie serii rozkładów szumu. W procesie odwrotnym model uczy się rekonstrukcji danych poprzez znalezienie przejść odwrotnych w łańcuchu⁸⁹.

Model dyfuzji, w swojej podstawowej formie, generuje dane z losowo pobranych próbek z przestrzeni ukrytej. Aby możliwe było nadzorowanie procesu tworzenia obrazów, modele dyfuzji rozszerzono m.in. o *classifier guidance*, technikę umożliwiającą warunkowanie modeli dyfuzji na etykietach klasyfikatora (*class-conditional diffusion models*). Klasyfikator jest najpierw trenowany na zaszumionych obrazach, a podczas próbkowania z przestrzeni ukrytej (powstałej w procesie dyfuzji) używane są gradienty z klasyfikatora do prowadzenia próby w mechanizmie odwróconej dyfuzji w stronę docelowej klasy.

89 Dhariwal, P., Nichol, A. (2021). *Diffusion Models Beat GANs on Image Synthesis*. <https://arxiv.org/pdf/2105.05233v4.pdf>

W 2021 roku, wraz z opublikowaniem modelu GLIDE⁹⁰ (*guided language to image diffusion for generation and editing*), popularność zaczęła zyskiwać metoda generowania obrazów niewymagająca klasyfikatora (*classifier-free guidance*). Metoda ta pozwala uczyć wektor gradientu, przechodząc sukcesywnie przez kolejne stopnie odwróconej dyfuzji. Podczas uczenia modelu klasa klasyfikatora jest zastępowana przez klasę pustą, co pozwala na nauczenie modelu wiedzy bardziej ogólnej. Ponadto upraszcza to generowanie obrazów za pomocą wektorów tekstu, które przenoszą więcej informacji niż klasa i nie są łatwo rozróżnialne.

Inną możliwością kierowania generacji modeli dyfuzji jest – również zaprezentowane jako jeden z modeli GLIDE – warunkowanie techniką CLIP⁹¹. Polega ona na wstępnym wytrenowaniu kodera obrazu zamieniającego dane z pikseli na wektor analogiczny do tekstowego. Wektory tekstowy i obrazowy są następnie łączone, a w procesie uczenia minimalizowany jest ich iloczyn skalarny. Taki sposób optymalizacji pozwala na dowolne formułowanie opisu obrazu. Na czas trenowania generatora moduł CLIP zostaje zamrożony, co implikuje ścisłą zależność między jakością bazowego modelu CLIP a generatorem.

Podobnych technik używają następcy GLIDE, które z uwagi na wielkość modeli wymagają znacznie większych mocy obliczeniowych oraz niejednokrotnie mniejszych modeli bazowych. Pozwala to jednak na zbliżanie się modeli do osiągnięcia zadowalających wyników w zadaniach coraz bardziej abstrakcyjnych.

DALL-E 2

Wspomniany już DALL-E 2 to wszechstronniejszy i wydajniejszy następca DALL-E, który zdolny jest do generowania obrazów bardziej realistycznych i o wyższej rozdzielczości. DALL-E 2 może generować obrazy łączące różne i niepowiązane ze sobą obiekty w całkowicie przekonujący sposób,

90 Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., Chen, M. (2021). *GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models*. <https://arxiv.org/pdf/2112.10741.pdf>

91 Radford, A., Kim, J., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I. (2021). *Learning Transferable Visual Models From Natural Language Supervision*. <https://arxiv.org/pdf/2103.00020.pdf>

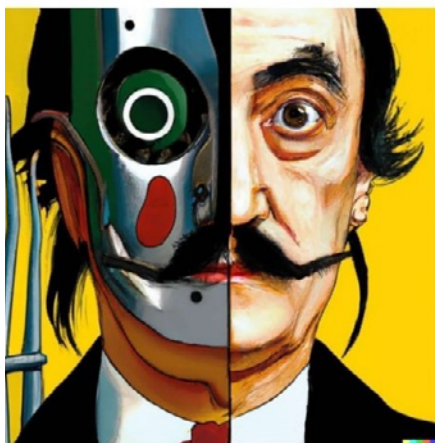
modyfikować istniejące obrazy, tworząc ich wariacje z zachowaniem istotnych cech, czy też dokonywać realistycznych edycji istniejących obrazów (*inpainting*).

U podstaw DALL-E 2 leżą dwa kluczowe elementy: moduł *prior* i dekodery. Najpierw opis tekstowy przekształcany jest do postaci wektorowej, który jest następnie przekazywany do modułu *prior* w celu utworzenia reprezentacji obrazu. Ta reprezentacja przekazywana jest dalej do dekodera, który zamienia ją w syntetyczny obraz. Reprezentacje tekstów i obrazów są tworzone za pomocą modelu CLIP, składającego się z kodera tekstu i kodera obrazów. Kodery mapują dane wejściowe do wspólnej przestrzeni, tak aby pasujące pary leżały w swoim bliskim sąsiedztwie. Ponieważ z danym opisem może być zgodna nieskończona liczba obrazów, w module *prior* potrzebny jest dodatkowy model do przetłumaczenia reprezentacji tekstu (*text embedding*) na reprezentację obrazu (*image embedding*). Autorzy przetestowali dwie klasy modeli możliwych do zastosowania: model autoregresyjny i model dyfuzji. Ten drugi na drodze eksperymentalnej uznano za lepszy, szczególnie z obliczeniowego punktu widzenia. Dekoder, który generuje obrazy, przekształcając reprezentację obrazu w rzeczywisty obraz, również wykorzystuje model dyfuzji. Jest to zmodyfikowana wersja GLIDE, którą autorzy nazwali unCLIP. Jak sama nazwa wskazuje, unCLIP skutecznie odwraca mapowanie wyuczone przez koder obrazów CLIP. Główną zmianą, w stosunku do prototypu, jest dodanie dodatkowego wektora reprezentacji obrazu w przestrzeni modelu CLIP (stworzonego przez moduł *prior*) do kodera tekstu GLIDE^{92 93}.

Twórcy DALL-E 2 mają nadzieję, że system umożliwi ludziom kreatywne wyrażanie siebie, a także pomoże zrozumieć, jak zaawansowane systemy AI widzą i rozumieją nasz świat. Niezwykły potencjał DALL-E 2 sprawił, że model nie został udostępniony publicznie, aby zapobiec jego niewłaściwemu wykorzystaniu. Twórcy planują przeprowadzenie większej liczby testów i badań, w szczególności tych związanych z możliwymi skutkami niewłaściwego użytkowania opracowanej technologii.

92 Ramesh, A . How DALL-E 2 Works. Pobrano 06.09.2022 z <http://adityaramesh.com/posts/dalle2/dalle2.html>

93 Ramesh, A ., Dhariwal, P., Nichol, A ., Chu, C., Chen, M. (2022). *Hierarchical Text-Conditional ...*, dz. cyt.



vibrant portrait painting of Salvador Dalí with a robotic half face



a shiba inu wearing a beret and black turtleneck



a close up of a handpalm with leaves growing from it



an espresso machine that makes coffee from human souls, artstation



panda mad scientist mixing sparkling chemicals, artstation



a corgi's head depicted as an explosion of a nebula



a dolphin in an astronaut suit on saturn, artstation



a propaganda poster depicting a cat dressed as french emperor napoleon holding a piece of cheese



a teddy bear on a skateboard in times square

Rysunek 16: Przykłady obrazów wygenerowanych przez DALL-E 2⁹⁴

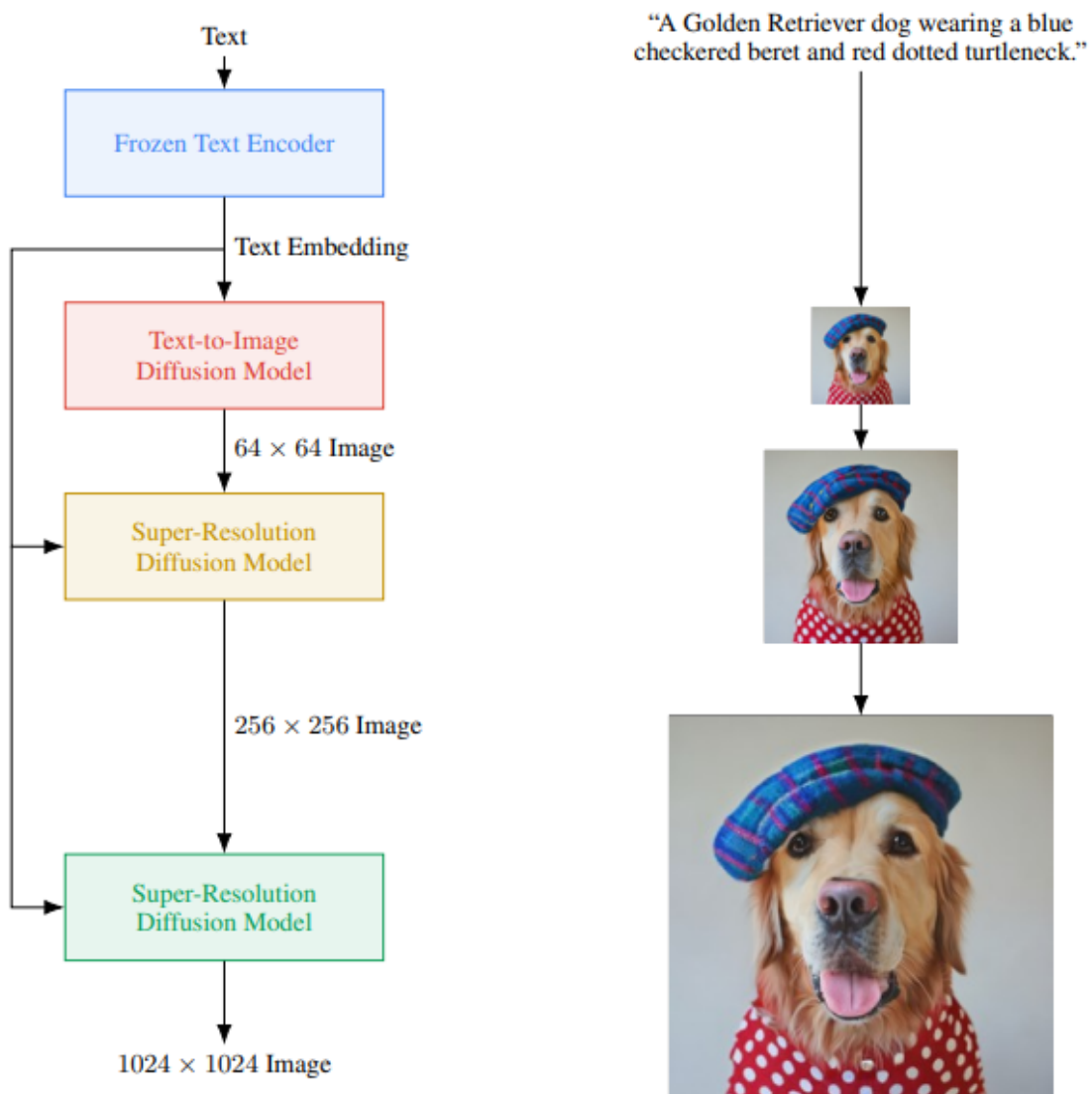
94 Tamże.

Imagen

W krótkim czasie po zaprezentowaniu DALL-E 2 zespół badaczy z Google Brain przedstawił Imagen^{95 96}, kolejny model do generowania obrazów z opisów tekstowych, korzystający z metody dyfuzji. Imagen, dzięki zastosowaniu modeli językowych opartych na strukturze Transformer oraz modeli dyfuzji, łączy w sobie głęboki poziom zrozumienia języka z możliwością generowania wysoce realistycznych obrazów.

Wstępnie wytrenowany model językowy służy do tworzenia reprezentacji tekstów, które, przechodząc przez sekwencję warunkowych modeli dyfuzji, są przekształcane w zdjęcia o rosnących rozdzielczościach. Podejścia zastosowane w Imagen i DALL-E 2 są koncepcyjnie zbieżne. Jednakże w różnych elementach architektury Imagen autorzy dodali wiele nowatorskich ulepszeń, które doprowadziły do osiągnięcia najnowocześniejszego modelu w dziedzinie generowania obrazów, osiągając najwyższy wynik FID (*Fréchet inception distance*), czyli metryki służącej do oceny jakości obrazów tworzonych przez model generatywny. Imagen, zamiast uczyć model językowy wraz z modelem generowania obrazu, korzysta z gotowego kodera tekstu T5-XXL⁹⁷, który nie zmienia się podczas uczenia modelu generatywnego. Dzięki temu lepiej rozumie kontekst opisów tekstowych i tworzy bardziej wartościowe reprezentacje przekształcane przez sekwencję warunkowych modeli dyfuzji w syntetyczny obraz. W pierwszym modelu dyfuzji generowany jest obraz o niskiej rozdzielczości. Zadaniem kolejnych jest iteracyjne poprawienie jakości wygenerowanego obrazu (modele super rozdzielczości). Do tych modeli autorzy zastosowali ulepszoną wersję sieci neuronowej o nazwie *Efficient U-net*.

-
- 95 Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, S.K., Ayan, B.K., Mahdavi, S.S., Lopes, R.G., Salimans, T., Ho, J., Fleet, D.J., Norouzi, M. (2022). *Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding*. <https://arxiv.org/pdf/2205.11487.pdf>
- 96 Google Research, Brain Team. *Imagen: unprecedented photorealism x deep level of language understanding*. Pobrano 07.09.2022 z <https://imagen.research.google/>
- 97 Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J. (2020). *Exploring the Limits ...*, dz. cyt.



Rysunek 17: Uproszczona struktura Imagen ⁹⁸

Podobnie jak w przypadku DALL-E 2, autorzy nie udostępnili publicznie Imagen, powołując się na możliwe skutki niewłaściwego użytkowania opracowanej technologii. Zespół ma jednak nadzieję, że ich praca posłuży jako inspiracja do przyszłych badań nad zastosowaniem jeszcze większych modeli językowych do generowania obrazów z opisów tekstowych.

⁹⁸ Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, S.K., Ayan, B.K., Mahdavi, S.S., Lopes, R.G., Salimans, T., Ho, J., Fleet, D.J., Norouzi, M. (2022). *Photorealistic Text-to-Image ...*, dz. cyt.



Sprouts in the shape of text 'Imagen' coming out of a fairytale book.



A photo of a Shiba Inu dog with a backpack riding a bike. It is wearing sunglasses and a beach hat.



A high contrast portrait of a very happy fuzzy panda dressed as a chef in a high end kitchen making dough. There is a painting of flowers on the wall behind him.



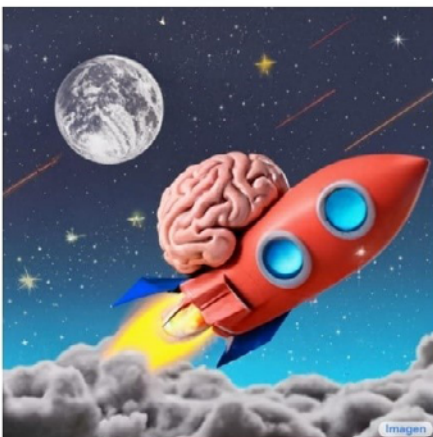
Teddy bears swimming at the Olympics 400m Butterfly event.



A cute corgi lives in a house made out of sushi.



A cute sloth holding a small treasure chest. A bright golden glow is coming from the chest.



A brain riding a rocketship heading towards the moon.



A dragon fruit wearing karate belt in the snow.



A strawberry mug filled with white sesame seeds. The mug is floating in a dark chocolate sea.

Rysunek 18: Przykłady obrazów wygenerowanych przez Imagen ⁹⁹

⁹⁹ Tamże.

Generowanie danych obrazowych – wybrane zagadnienia

Problem generowania danych obrazowych jest zagadnieniem wieloaspektowym, ukierunkowanym na realizację różnorodnych celów. Poniżej przedstawiono wybrane z nich, cieszące się największą popularnością i korzystające z osiągnięć omówionych metod generowania danych obrazowych.

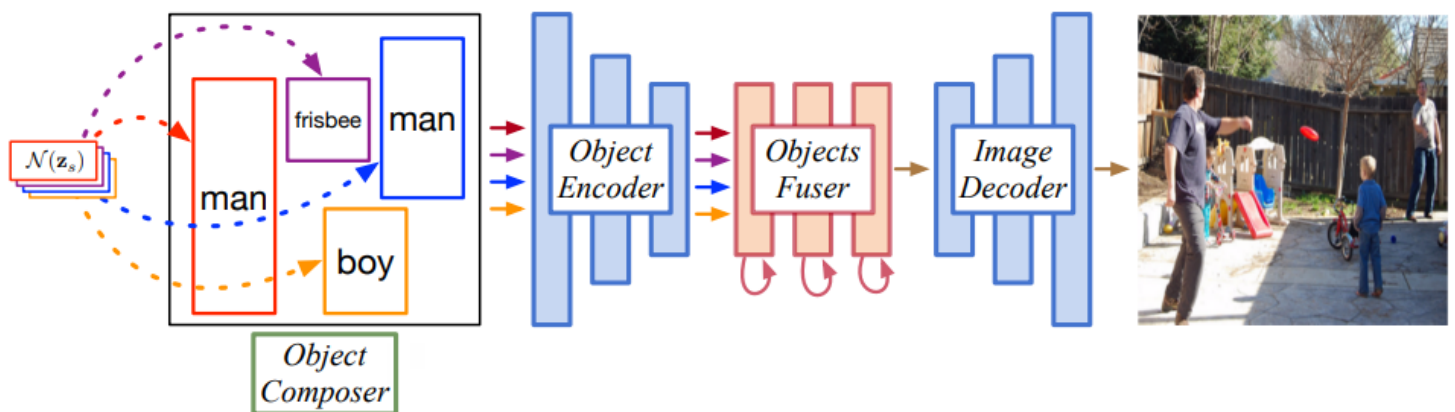
Generowanie text-to-image

Zadanie generowania obrazów z opisów tekstowych (*text-to-image-generation*, T2I) ¹⁰⁰, łączące ze sobą elementy NLP i rozpoznawania obrazów, ma na celu stworzenie semantycznie spójnych i wizualnie realistycznych obrazów uwarunkowanych opisem tekstowym. Problem T2I może być traktowany jako zadanie odwrotne do tworzenia reprezentacji słownych obrazów (*image captioning*). Wymaga nie tylko potężnego modelu generatywnego, ale również zrozumienia i posługiwania się intermodalną percepcją, czyli tworzenia powiązań między różnymi danymi percepcyjnymi. Aby przekształcić tekst na obraz, konieczne jest zrozumienie tekstu wejściowego, dopasowanie obiektów i cech do odpowiednich słów i ich synonimów, a także uchwycenie złożoności kształtów, kolorów, gestów itp. za pomocą pikseli oraz umiejętnego nakładania się lub złożenia różnych obiektów i cech. To wszystko wymaga wyższego poziomu zdolności poznawczych i może być zadaniem nietrywialnym nawet dla człowieka. Dlatego zbudowanie systemu, który rozumie związek między językiem a wizją i który może generować obrazy właściwie wyrażające znaczenie opisów tekstowych, będzie istotnym krokiem w kierunku stworzenia ogólnej sztucznej inteligencji (*artificial general intelligence*, AGI).

¹⁰⁰ Frolov, S., Hinz, T., Raue, F., Hees, J., Dengel, A. R. (2021). *Adversarial Text-to-Image Synthesis: A Review*. <https://arxiv.org/pdf/2101.09983.pdf>

Generowanie layout-to-image

Celem generowania obrazu na podstawie jego układu (*layout-to-image generation*, L2I) jest stworzenie obrazu przedstawiającego wiele różnych obiektów na wybranym tle, których rozmieszczenie jest uwarunkowane układem określonym przez zbiór ramek ograniczających (*bounding box*) o określonej wielkości, położeniu i kategoriach. Zadanie L2I jest trudnym zagadnieniem w związku z problemem modelowania relacji jeden-do-wielu. Wiele obrazów może być zgodnych z określonym układem: ten sam układ może być reprezentowany przez obiekty o różnym wyglądzie, a układ obiektów może, choć nie musi, uwzględniać interakcje między nimi. Informacje przekazywane do modelu w postaci ramek ograniczających położenie obiektów czy przypisanych im kategorii (etykiet) mogą nie być wystarczające aby wygenerować spójny obraz. Wykorzystanie układu do tworzenia obrazu zwykle służy jako pośredni etap np. przy generowaniu text-to-image, na podstawie kontekstu lub grafów odwzorowujących relacje w danym ujęciu (*scene graphs*). W tych zadaniach układ obrazu służy jako pośrednia reprezentacja między formatem wejściowym a wyjściowym¹⁰¹.



Rysunek 19: Uproszczony schemat generowania layout-to-image¹⁰²

101 Zhao, B., Meng, L., Yin, W., Sigal, L. (2019). *Image Generation from Layout*. <https://arxiv.org/pdf/1811.11389.pdf>

102 Tamże.

Inpainting

Rekonstrukcję brakujących części obrazu lub usuwanie obecnego na nim obiektu tak, aby wyjściowy obraz wyglądał realistycznie umożliwia zadanie *image inpainting*. Jego celem może być restaurowanie starych obrazów lub zdjęć (np. usuwanie pęknięć, rys, zagnieceń), odzyskiwanie brakujących części, kodowanie obrazów do transmisji oraz usuwanie z obrazów niechcianych obiektów. Ponadto jest to obecnie jedno z kluczowych narzędzi do zapewnienia ochrony prywatności osób fizycznych i obiektów, które można z nimi powiązać¹⁰³.

Najprostszym sposobem uzupełniania brakujących elementów obrazu jest zastosowanie metody kopiuj-wklej. Polega ona na wyszukiwaniu najbardziej podobnych pikseli dostępnych na obrazie lub w repozytorium danych obrazowych, a następnie bezpośrednim wklejeniu wybranych części w brakujące miejsca. Jednak algorytm wyszukiwania jest mało wydajny i ciężki do uogólnienia. Nie radzi sobie również z wypełnianiem większych i złożonych fragmentów obrazu. Obecnie najczęściej stosuje się metody oparte na uczeniu głębokim, które wykazały doskonałe wyniki w zakresie inpaintingu obrazów, umożliwiające generowanie brakujących pikseli globalnie z odpowiednim, pasującym do reszty zdjęcia stylem, a lokalnie dobrze odwzorowujących drobne tekstury.

Tłumaczenie image-to-image

Coraz większe zainteresowanie wzbudza tłumaczenie obrazu na obraz (*image-to-image translation*, I2I). Zadanie I2I zapewnia szeroki zakres zastosowań w wielu problemach związanych z rozpoznawaniem i przetwarzaniem obrazów, które można przedstawić jako tłumaczenie obrazu wejściowego na pewien obraz wyjściowy. Podobnie jak tekst może być wyrażony w języku polskim lub angielskim, tak obraz może być renderowany w postaci obrazu RGB, map krawędzi (*edge map*), semantycznej mapy etykiet (*semantic label map*) itp. Zadanie I2I polega na przeniesieniu (przetłumaczeniu) obrazu z domeny źródłowej na inną tak, aby posiadał styl lub cechy charakterystyczne dla domeny docelowej, przy jednoczesnym zachowaniu reprezentowanych przez

103 Elharrouss, O., Almaadeed, N., Al-Maadeed, S.A., Akbari, Y. (2019). *Image Inpainting: A Review*. <https://arxiv.org/ftp/arxiv/papers/1909/1909.06399.pdf>

obraz treści. Powszechnie znanym przykładem zastosowania I2I są aplikacje oferujące edycję zdjęć za pomocą różnych filtrów: zdjęcie oryginalne można przetłumaczyć na zdjęcie w stylu artystycznym, osiągając np. efekt "zdjęcia z kreskówki". I2I znalazło zastosowanie m.in. w syntezie semantycznej obrazów (maska segmentacji semantycznej przegenerowana w rzeczywiste obrazy), segmentacji obrazów, inpaintingu obrazów, transferze stylu, adaptacji domenowej, tworzeniu szkiców¹⁰⁴. Problem I2I rozwiązuje się najczęściej za pomocą modeli GAN (np. CycleGAN) i VAE.

Generowanie obrazów medycznych

Obrazy medyczne uzyskiwane są za pomocą rezonansu magnetycznego (MRI), tomografii komputerowej (CT), pozytonowej tomografii emisyjnej (PET) czy badania ultrasonograficznego (USG). Uwzględniając informacje specyficzne dla danej metody obrazowania, nie tylko pomagają one w lepszej diagnostyce, ale służą również za cenne źródło danych do prowadzenia badań naukowych. Niestety obawy społeczeństwa oraz surowe przepisy dotyczące ochrony osób fizycznych w związku z przetwarzaniem danych osobowych istotnie hamują możliwości rozwoju w tym obszarze. Jednak w wielu przypadkach badaczy nie interesują informacje o konkretnych osobach fizycznych, ponieważ w badaniach medycznych informacje są zwykle analizowane na poziomie kohort (grup osób wybieranych na podstawie pewnych wspólnych cech)¹⁰⁵.

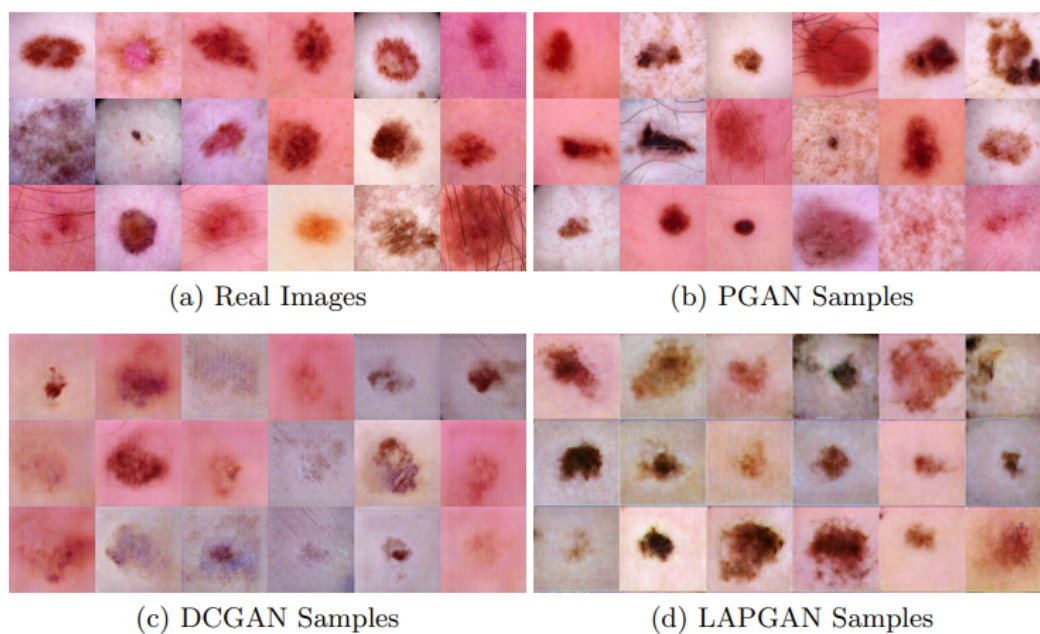
Obiecujące osiągnięcia w dziedzinie rozpoznawania i przetwarzania obrazów zyskały duże zainteresowanie badaczy i wzbudziły nadzieje na wykorzystanie tych wyników w dziedzinie medycyny. Tworzenie odpowiednich zbiorów danych jest kosztowne i trudne ze względu na wrażliwy charakter danych i wysoce szczegółową wiedzę dziedzinową wymaganą do ich rzetelnego opisanie. W rezultacie manualnie tworzone zbiory danych są znacznie mniejsze od tych wykorzystywanych w innych zadaniach przetwarzania obrazów. Biorąc pod uwagę te ograniczenia, metody syntezy obrazów medycznych stały

104 Isola, P., Zhu, J., Zhou, T., Efros, A. A. (2017). *Image-to-Image Translation with Conditional Adversarial Networks*. <https://arxiv.org/pdf/1611.07004.pdf>; Pang, Y., Lin, J., Qin, T., Chen, Z. (2022). *Image-to-Image Translation: Methods and Applications*. <https://arxiv.org/pdf/2101.08629.pdf>

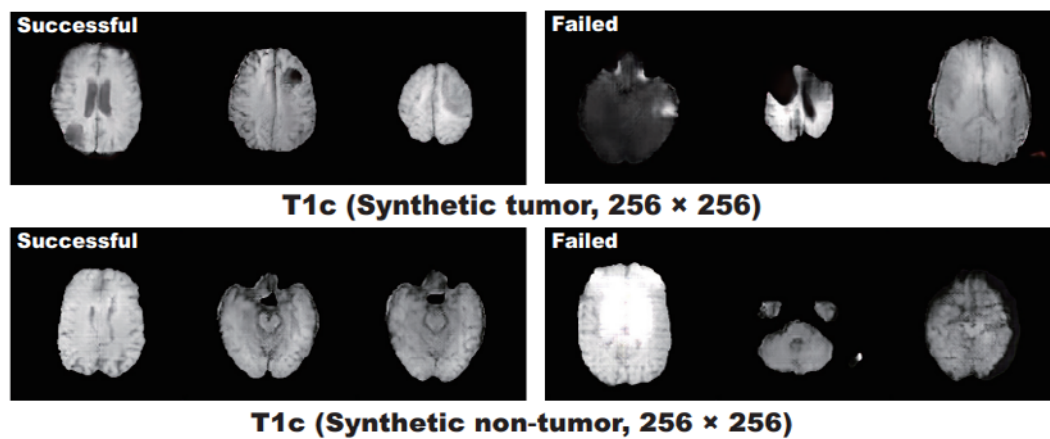
105 Yi, X., Walia, E., Babyn, P.S. (2019). *Generative Adversarial Network in Medical Imaging: A Review*. <https://arxiv.org/pdf/1809.07294.pdf>; Koshino, K., Werner, R.A., Pomper, M.G., Bundschuh, R.A., Toriumi, F., Higuchi, T., Rowe, S.P. (2021). *Narrative review of generative adversarial networks in medical and molecular imaging*. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8246192/>

się aktywnym obszarem badawczym. Wśród stosowanych metod dominują zwłaszcza sieci CNN oraz GAN. Sztucznie utworzone obrazy mogą być dodawane do istniejących zbiorów danych, zwiększając ilość i różnorodność danych treningowych i ostatecznie poprawiając algorytmy stosowane do wspomaganých przez sztuczną inteligencję systemów diagnoz. W ostatnich latach pojawiło się wiele prac przedstawiających możliwości generowania realistycznych danych syntetycznych, np. obrazy zmian skórnych w wysokiej rozdzielczości¹⁰⁶, siatkówki oka¹⁰⁷, MRI mózgu¹⁰⁸, obrazy guzków raka płuc¹⁰⁹ itp. Przy generacji tych obrazów zastosowano modele zbudowane na architekturze generatywnych sieci przeciwstawnych oraz ich odmiany: DCGAN¹¹⁰, WGAN¹¹¹, PGGAN¹¹².

-
- 106 Baur, C., Albarqouni, S., Navab, N. (2018). *Generating Highly Realistic Images of Skin Lesions with GANs*. <https://arxiv.org/pdf/1809.01410.pdf>
- 107 Costa, P., Galdran, A., Meyer, M.I., Niemeijer, M., Abràmoff, M., Mendonça, A.M., Campilho, A. (2017). *End-to-End Adversarial Retinal Image Synthesis*. http://eprints.bournemouth.ac.uk/34883/1/tmi_2017.pdf; Zhao, H., Li, H., Maurer-Stroh, S., Cheng, L. (2018). *Synthesizing retinal and neuronal images with generative adversarial nets*. <https://www.sciencedirect.com/science/article/pii/S1361841518304596>
- 108 Han, C., Rundo, L., Araki, R., Furukawa, Y., Mauri, G., Nakayama, H., Hayashi, H. (2019). *Infinite Brain MR Images: PGGAN-based Data Augmentation for Tumor Detection*. <https://arxiv.org/pdf/1903.12564.pdf>
- 109 Chuquicusma, M.M., Hussein, S., Burt, J.R., Bagci, U. (2018). *How to fool radiologists with generative adversarial networks? A visual turing test for lung cancer diagnosis*. <https://arxiv.org/pdf/1710.09762.pdf>
- 110 Radford, A., Metz, L., Chintala, S. (2016). *Unsupervised Representation Learning ...*, dz. cyt.
- 111 Arjovsky, M., Chintala, S., Bottou, L. (2017). *Wasserstein GAN*. <https://arxiv.org/pdf/1701.07875.pdf>
- 112 Karras, T., Aila, T., Laine, S., Lehtinen, J. (2018). *Progressive Growing of GANs for Improved Quality, Stability, and Variation*. <https://arxiv.org/pdf/1710.10196.pdf>



Rysunek 20: Przykłady wygenerowanych za pomocą różnych modeli obrazów zmian skórnych¹¹³



Rysunek 21: Przykłady syntetycznych obrazów MRI¹¹⁴

¹¹³ Baur, C., Albarqouni, S., Navab, N. (2018). *Generating ...*, dz. cyt.

¹¹⁴ Han, C., Rundo, L., Araki, R., Furukawa, Y., Mauri, G., Nakayama, H., Hayashi, H. (2019). *Infinite Brain MR Images ...*, dz. cyt.

Generowanie danych tabelarycznych

Podczas gdy generowanie danych nieustrukturyzowanych, takich jak dane tekstowe i obrazy, pobudza ekscytację zarówno naukowców, jak i społeczeństwa, rozwój systemów AI zależy również od ilości i jakości danych ustrukturyzowanych, np. danych tabelarycznych. Dane można przedstawić w formie tabelarycznej na różne sposoby, w zależności od celu użycia (np. zapisy badań klinicznych, dane finansowe, wyniki spisów powszechnych itd.). Jak sama nazwa wskazuje, dane tabelaryczne to dane przechowywane w tabelach, gdzie każdy wiersz odpowiada jednemu wpisowi danych. Zapisywane w plikach płaskich lub w bazie danych, zwykle składają się z prostych wartości alfanumerycznych, takich jak liczby, daty lub fragmenty tekstu.

Zadanie generowania danych tabelarycznych polega na utworzeniu modelu generatywnego, tak aby wygenerowane z niego próby utworzyły syntetyczną tabelę, spełniającą określone kryteria. Przede wszystkim model uczenia maszynowego trenowany przy użyciu syntetycznych tabel powinien wykazywać zachowanie podobne do modelu trenowanego na tabeli rzeczywistej. Standardowo zadanie polega na modelowaniu wspólnego wielowymiarowego rozkładu prawdopodobieństwa dla danego zbioru danych, a następnie próbkowaniu z tego rozkładu. Mając na celu umożliwienie przetwarzania danych przy jednoczesnej ochronie prywatności osób reprezentowanych przez dane, badacze zaczęli również wykorzystywać metody oparte na randomizacji. Obecnie popularność zyskało podejście polegające na tworzeniu modeli generatywnych opartych na głębokich sieciach neuronowych – atrakcyjne zarówno pod względem wydajności, jak i elastyczności w generowaniu i manipulowaniu reprezentacjami danych, (np. autoenkodery wariacyjne, generatywne sieci przeciwstawne i ich liczne rozszerzenia).

Syntetyczne dane tabelaryczne mogą zrewolucjonizować różne obszary rozwoju technologii, takie jak pojazdy autonomiczne, opieka zdrowotna i usługi finansowe. Możliwość generowania danych, które wyglądają i zachowują się jak dane rzeczywiste pozwoli na realistyczne testowanie i opracowywanie nowych funkcji. Ponadto w syntetycznych danych tabelarycznych nie występuje relacja jeden-do-jeden między rekordami rzeczywistymi a rekordami syntetycznymi, co minimalizuje zagrożenie ponownej identyfikacji i pozwala na udostępnianie danych bez obaw o wyciek informacji. Stwarza to ogromne możliwości w zakresie modelowania zjawisk i zachowań, dla których

dostępność danych jest ograniczona z powodu kosztów zbierania danych, nacisku na ochronę prywatności oraz kontroli sprawowanej przez podmioty, które nie mogą lub nie chcą publicznie udostępniać danych.

Metody generowania danych tabelarycznych

Możemy wyróżnić dwie główne kategorie technik generowania syntetycznych danych tabelarycznych: techniki statystyczne oraz metody uczenia maszynowego. W obu przypadkach dane syntetyczne są wynikiem statystycznego modelowania rzeczywistych zbiorów danych. Utworzony model wykorzystywany jest do generowania nowych wartości o właściwościach statystycznych zbliżonych do tych obecnych w danych rzeczywistych. Posiadanie podobnych właściwości statystycznych oznacza, że model musi odtworzyć rozkład danych w takim stopniu, aby analityk pracujący z syntetycznym zbiorem danych otrzymał wyniki zbliżone do tych, które otrzymałby, pracując z danymi rzeczywistymi, tzn. wnioski wysnute z obu wersji danych powinny być jednakowe.

W przypadku gdy dane rzeczywiste są niedostępne, ale analityk danych ma pełną wiedzę na temat tego, jak wyglądałby rozkład danych, możliwe jest wygenerowanie losowej próbki z dowolnego rozkładu prawdopodobieństwa. Jednak użyteczność takich danych będzie zależeć od wiedzy analityka o badanym zjawisku. Jeśli dane rzeczywiste są dostępne, to dane syntetyczne są generowane z rozkładów najlepiej dopasowanych do danych rzeczywistych: tworzony jest model, który wyjaśnia zaobserwowane zjawisko, a następnie, za pomocą tego samego modelu, losowo generowane są nowe wartości danych. Aby dopasować rozkład do danych, analitycy stosują szereg różnych metod, począwszy od metody Monte Carlo po modele generatywne typu GAN.

Warto podkreślić, że tworząc modele generujące dane, nie należy się skupiać na samych danych rzeczywistych, ale skierować uwagę na procesy, które doprowadziły do ich powstania. W ten sposób możliwe jest osiągnięcie probabilistycznego przybliżenia rzeczywistych danych, które nie będzie zawierało żadnych identyfikowalnych informacji, jednocześnie zachowując właściwości oryginału.

Metoda Monte Carlo

Jedną z najstarszych i najczęściej stosowanych technik statystycznych do wnioskowania o populacji przy małej próbie jest metoda Monte Carlo (MC)¹¹⁵. Określa ona klasę algorytmów obliczeniowych, które do modelowania systemów probabilistycznych wykorzystują proces wielokrotnego próbkowania losowego. Celem jest przybliżenie możliwości wystąpienia zdarzenia losowego (komponentu probabilistycznego), tj. ustalenie prawdopodobieństw wystąpienia różnych wyników tego komponentu. Symulacja MC, używając zakresu wartości określonej w dziedzinie problemu i wykorzystując dane rozkłady prawdopodobieństw, buduje model możliwych wyników dla dowolnej zmiennej losowej. Po określonej liczbie prób wyniki są ponownie przeliczane, za każdym razem przy użyciu innych losowych wartości z określonej dziedziny. Za pomocą symulowanej próby możliwe jest oszacowanie wartości oczekiwanej komponentu probabilistycznego obecnego w badanym procesie. Pozwalają na to tzw. prawa wielkich liczb, zgodnie z którymi w miarę zwiększania liczby powtórzeń danego eksperymentu średnia wartość wyników zbliża się do rzeczywistej wartości prawdopodobieństwa wystąpienia tych zdarzeń.

W symulacji MC model tworzy wiele zbiorów danych, które można postrzegać jako realistyczne wariacje zbioru pierwotnego. Sięgnięcie po MC do generowania danych jest prostym i intuicyjnym rozwiązaniem, gdy celem jest modelowanie probabilistycznej natury zjawisk, a stosowanie innych metod matematycznych jest trudne lub niemożliwe ze względu na złożoność problemu. Metoda jest jednak uważana za niedokładną w porównaniu z innymi metodami generowania danych syntetycznych, ponieważ do uzyskania realistycznych wyników symulacji potrzebne są dokładne dane empiryczne, aby jak najlepiej zdefiniować dziedzinę rozważanego problemu.

MCMC i próbkowanie Gibbsa

Klasyczne metody Monte Carlo opierają się na generowaniu próbek składających się z niezależnych obserwacji. Wariant tej metody pn. *Markov Chain Monte Carlo* (MCMC) jest stosowany do generowania sekwencji zależnych obserwacji, będących, jak wskazuje nazwa, łańcuchami Markowa. W części

¹¹⁵ Raychaudhuri, S. (2008). *Introduction to Monte Carlo simulation*. <https://www.informs-sim.org/wsc08papers/012.pdf>

dotyczącej metod generowania danych tekstowych wyjaśniono, że łańcuchy Markowa są metodą tworzenia sekwencji zmiennych losowych, w której prawdopodobieństwo wystąpienia aktualnego zdarzenia zależy jedynie od zdarzenia poprzedniego w łańcuchu. Połączenie tych dwóch metod umożliwia próbkowanie wysokowymiarowych rozkładów prawdopodobieństwa uwzględniających zależność między próbkami, tj. skonstruowanie łańcucha Markowa, który zawiera symulacje MC (następna próbka zależy od ostatniej wylosowanej próbki).

Istnieje wiele algorytmów MCMC, które definiują różne sposoby konstruowania łańcuchów Markowa. Jednym z nich jest tzw. próbkowanie Gibbsa (*Gibbs sampling*), opierające się na możliwości próbkowania z rozkładów warunkowych zmiennych. Jest to korzystne zwłaszcza wtedy, gdy rozkład łączny zmiennych nie jest znany lub trudno na nim przeprowadzić próbkowanie, ale znane są rozkłady warunkowe każdej ze zmiennych. Iteracyjne próbkowanie z rozkładów warunkowych, uzależnionych od bieżących wartości innych zmiennych, może ostatecznie doprowadzić do uzyskania przybliżonej próby z rozkładu łącznego, czyli jego oszacowania¹¹⁶.

Sieci bayesowskie

W sieci bayesowskiej (*bayesian network*, BN), czyli probabilistycznym modelu grafowym łącznego rozkładu prawdopodobieństwa dla zbioru zmiennych, każdy węzeł reprezentuje zmienną losową, a krawędzie między węzłami reprezentują probabilistyczne zależności między nimi. Zasadniczo sieć bayesowska składa się z dwóch części: struktury sieciowej, w postaci skierowanego grafu acyklicznego, która wyraża wszystkie parami warunkowe zależności między zmiennymi, oraz zbioru lokalnych rozkładów prawdopodobieństwa, wyrażającego warunkowe rozkłady dla tych par (prawdopodobieństwa skutku ze względu na przyczynę). Łączny rozkład prawdopodobieństwa w sieci bayesowskiej można wyznaczyć, stosując regułę łańcuchową, za pomocą lokalnych rozkładów prawdopodobieństwa, analitycznie zdefiniowanych przez dany model BN. Dzięki temu, zamiast bezpośredniego próbkowania z wielowymiarowego rozkładu prawdopodobieństwa, możliwe jest generowanie próbek z czynników jego produktu, czyli z rozkładów warunkowych, co jest wydajne obliczeniowo i ułatwia skalowanie wielowymiarowe. Sieci

116 Integration, M.C. (2002). *Markov Chain Monte Carlo and Gibbs Sampling*. <https://cis.temple.edu/~latecki/Courses/CIS2033-Spring12/GreatLectures/mcmc-gibbs-intro.pdf>

bayesowskie przekazują również dobrą intuicję do modelowania i syntezy danych populacji¹¹⁷.

W przypadku generowania danych syntetycznych struktura wykresu i warunkowe rozkłady prawdopodobieństwa są wnioskowane z danych rzeczywistych. Chociaż ogranicza to ich zastosowanie w niektórych dziedzinach, sieci bayesowskie są nadal popularną drogą do tworzenia zbiorów syntetycznych.

Generatywne sieci przeciwstawne (GAN)

Tworzenie modeli generatywnych z wykorzystaniem sieci neuronowych, takich jak generatywne sieci przeciwstawne GAN, jest atrakcyjne zarówno pod względem wydajności, jak i elastyczności w reprezentowaniu i tworzeniu realistycznych danych syntetycznych o wysokiej jakości. Jak już omówiono w części poświęconej generowaniu danych obrazowych, sieci GAN składają się z dwóch komponentów: generatora i dyskryminatora. Pierwszy służy do tworzenia fałszywych danych z wejściowego losowego szumu, a drugi do klasyfikowania próbek jako prawdziwych lub fałszywych (wytworzonych przez generator), a jego wydajność jest wykorzystywana w procesie uczenia do aktualizacji i optymalizacji generatora i dyskryminatora. Zdecydowana większość badań w dziedzinie GAN dotyczy danych obrazowych, ale coraz częściej eksplorowane są również inne obszary, gdzie zastosowanie koncepcji generatora i dyskryminatora wydaje się być obiecującym kierunkiem. Przykładem są właśnie dane tabelaryczne.

Mogłoby się wydawać, że uczenie modeli na podstawie danych nieustrukturyzowanych, takich jak obrazy lub tekst, jest znacznie trudniejsze od uczenia na danych ustrukturyzowanych. Jednak algorytmy wykorzystywane do generowania danych tabelarycznych mogą bardzo szybko stać się bardzo złożone. Ta złożoność wynika przede wszystkim z różnych typów danych obecnych w danych tabelarycznych. Mogą to być dane numeryczne, kategoryczne, tekstowe, szeregi czasowe i inne, przybierające różnorodne rozkłady i charakterystyki. Aby wygenerować dane syntetyczne, modele generatywne są szkolone na rzeczywistych próbach i uczą się je aproksymować. Wyzwanie polega na

117 Young, J., Graham, P., Penny, R. (2009). *Using Bayesian Networks to Create Synthetic Data*. <https://www.scb.se/contentassets/ca21efb41fee47d293bbee5bf7be7fb3/using-bayesian-networks-to-create-synthetic-data.pdf>; Gogoshin, G., Branciamore, S., Rodin, A. S. (2021). *Synthetic data generation with probabilistic Bayesian Networks*. <https://www.aimspress.com/article/doi/10.3934/mbe.2021426>

wygenerowaniu danych, które wystarczająco dobrze odwzorowują zarówno strukturalne, jak i statystyczne właściwości rzeczywistych danych, ale których wartości nie uzyskuje się przez bezpośrednią obserwację rzeczywistego procesu generatywnego. To z kolei wiąże się z jeszcze jedną trudnością: syntetyczne dane tabelaryczne są trudne do oceny jakości.

Niemniej jednak wyniki eksperymentalne pokazują, że stosując sieci GAN, można wygenerować realistyczne syntetyczne dane tabelaryczne możliwe do wykorzystania w wybranych przypadkach biznesowych. W literaturze istnieje wiele modeli GAN zaprojektowanych do konkretnych zastosowań. Przyjrzyjmy się niektórym z nich.

table-GAN

Model table-GAN¹¹⁸ jest zbudowaną na architekturze GAN metodą syntezy danych tabelarycznych, zdolną do generowania danych o podobnych właściwościach statystycznych do rzeczywistych. table-GAN pozwala na synteze tabel zawierających wartości dyskretne, ciągłe i kategoryczne, a zbudowany został głównie z myślą o ochronie danych i ograniczeniu ryzyka związanego z ich możliwym wyciekiem. Do stałych elementów GAN, tj. generatora i dyskryminatora, autorzy dołączyli trzeci element, którym jest klasyfikator. Dyskryminator próbuje odróżnić rekordy rzeczywiste od syntetycznych, a generator utrudnia zadanie dyskryminatora, tworząc coraz bardziej realistyczne rekordy. W procesie uczenia udział bierze również klasyfikator, a jego celem jest zwiększenie integralności semantycznej generowanych rekordów. Klasyfikator jest szkolony z wykorzystaniem rzeczywistych danych, tak aby nauczył się prawdziwych korelacji między etykietami i innymi atrybutami tabeli. Na podstawie zdobytej wiedzy klasyfikator sprawdza, czy wygenerowane rekordy są semantycznie poprawne. Zasadniczo sam dyskryminator może do pewnego stopnia ocenić integralność semantyczną generowanych rekordów, jednak może dojść do sytuacji, w której zostanie on oszukany i utworzona tabela będzie wyglądała na sfabrykowaną. Dodanie klasyfikatora w formie nadzorcy generatora temu zapobiega.

Autorzy modelu wykazali, że modele uczenia maszynowego trenowane na tabelach syntetycznych na zbiorze testowym wykazują *performance* podobny

118 Park, N., Mohammadi, M., Gorde, K., Jajodia, S., Park, H., Kim, Y. (2018). *Data Synthesis based on Generative Adversarial Networks*. <https://arxiv.org/pdf/1806.03384.pdf>

do modeli wytrenowanych przy użyciu tabel rzeczywistych, co jest obiecującą perspektywą dla rozwoju technologii w świetle przepisów o ochronie prywatności.

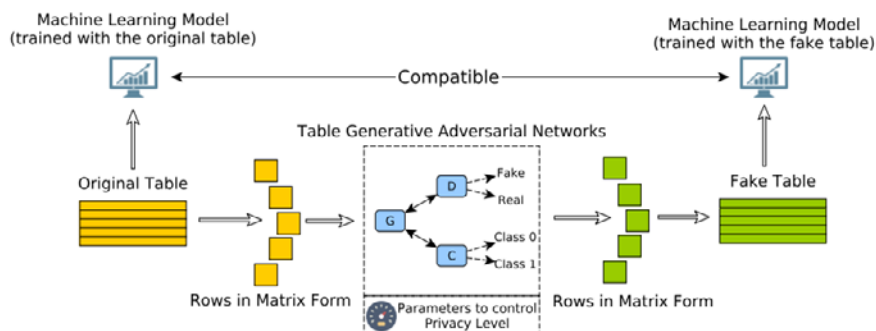


Figure 1: The overall workflow of the proposed method. The fake table (marked in green) is generated by the proposed table-GAN trained using the original table (marked in yellow). Machine learning models trained using the fake table should show the same behaviors as models trained using the original table (i.e., model compatibility). Our goal is to achieve general model compatibility regardless of machine learning algorithms and tasks.

Rysunek 22: Uproszczony schemat działania table-GAN ¹¹⁹

CTGAN

Kolejnym modelem opartym na architekturze GAN służącym do modelowania danych tabelarycznych i warunkowego próbkowania rekordów w celu tworzenia syntetycznych tabel jest CTGAN¹²⁰ (*conditional tabular generative adversarial network*). Metoda wprowadza kilka ulepszeń, aby przezwyciężyć wyzwania związane z modelowaniem danych tabelarycznych przy użyciu GAN, np. obecność różnych typów danych, występowanie rozkładów nie-gaussowskich i multimodalnych, niezrównoważenie atrybutów kategoryjnych. W strukturze CTGAN proces treningowy został rozszerzony o *mode-specific normalization*, nowatorski rodzaj normalizacji umożliwiający przekształcanie wartości ciągłych o dowolnym zakresie i rozkładzie na ograniczoną reprezentację wektorową, odpowiednią dla sieci neuronowych. Dodatkowo zastosowano generator warunkowy i trenowanie przez próbkowanie (*training-by-sampling*), co ma pozwolić na przezwyciężenie problemu niezrównoważonych danych treningowych. W szczególności celem jest to, aby wszystkie kategorie atrybutów podczas procesu uczenia były próbkowane równomiernie

¹¹⁹ Tamże.

¹²⁰ Xu, L., Skoularidou, M., Cuesta-Infante, A., Veeramachaneni, K. (2019). *Modeling Tabular data using Conditional GAN*. <https://arxiv.org/pdf/1907.00503.pdf>

(ale niekoniecznie w sposób jednolity) i żeby możliwe było odzyskanie rzeczywistego rozkładu danych podczas testów. Wprowadzone udoskonalenia powodują, że CTGAN może sprostać typowym wyzwaniom związanym z generowaniem danych, tworząc wysokiej jakości syntetyczne tabele.

CTAB-GAN

Warunkowy syntezytor danych tabelarycznych CTAB-GAN¹²¹ może modelować różne typy danych o złożonych rozkładach. Autorzy modelu wzięli pod uwagę ograniczenia wcześniejszych modeli, takie jak kodowanie mieszanych typów danych (zmiennych ciągłych i kategoriowych), efektywne modelowanie rozkładów z długimi ogonami oraz zwiększoną odporność na niezrównoważone zmienne kategoryjne i skośność rozkładów. W CTAB-GAN do struktury warunkowego GAN dodano dodatkowy klasyfikator (równoległe do dyskryminatora) wraz z funkcją straty dla klasyfikacji (*classification loss*), której zadaniem jest obliczanie rozbieżności między klasą generowaną i prognozowaną, co pomaga zwiększyć semantyczną integralność generowanych rekordów. Wprowadzono również nowy sposób kodowania warunkowego (tzw. *mixed-type encoder*), pozwalający na efektywne kodowanie zmiennych mieszanych i radzenie sobie z wysoce skośnymi rozkładami dla zmiennych ciągłych.

CTAB-GAN składa się z trzech komponentów: generatora, dyskryminatora oraz klasyfikatora, który, podobnie jak w table-GAN, ma na celu zwiększenie integralności semantycznej tworzonych rekordów. CTAB-GAN przewyższa wcześniejsze metody dzięki modelowaniu zmiennych mieszanych i zapewnia możliwość generowania niezrównoważonych zmiennych kategoryjnych oraz zmiennych ciągłych o złożonych rozkładach. Autorzy ocenili model w trzech wymiarach: użyteczności analizy opartej na uczeniu maszynowym z wykorzystaniem danych syntetycznych, statystycznego podobieństwa do danych rzeczywistych i zapewnienia ochrony prywatności, porównując go m.in. z modelami table-GAN i CTGAN. Wyniki osiągnięte przez CTAB-GAN przewyższyły pozostałe modele w każdym z wymienionych obszarów.

Dalsze badania nad modelem doprowadziły do jego rozszerzenia: CTAB-GAN+¹²². Celem nowego algorytmu jest dalsza poprawa jakości

121 Zhao, Z., Kunar, A., Scheer, H.V., Birke, R., Chen, L.Y. (2021). CTAB-GAN: Effective Table Data Synthesizing. <https://arxiv.org/pdf/2102.08369.pdf>

122 Zhao, Z., Kunar, A., Birke, R., Chen, L.Y. (2022). CTAB-GAN+: Enhancing Tabular Data Synthesis. <https://arxiv.org/pdf/2204.00401.pdf>

danych syntetycznych pod względem użyteczności uczenia maszynowego i podobieństwa statystycznego oraz efektywne wdrożenie prywatności różnicowej do szkolenia GAN, aby móc kontrolować jego wydajność w ramach różnych poziomów prywatności (*privacy budgets*).

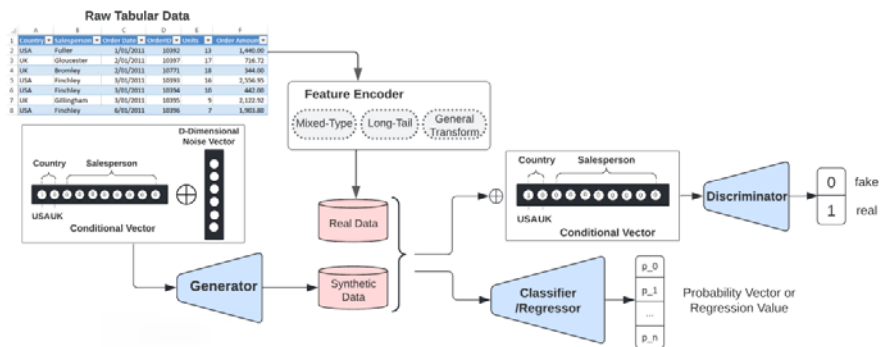


Figure 2: Synthetic Tabular Data Generation via CTAB-GAN+

Rysunek 23: Uproszczony schemat działania CTAB-GAN+¹²³

TimeGAN

Szereg czasowy to ciąg obserwacji uporządkowanych w czasie. Aby skutecznie modelować szeregi czasowe, należy uchwycić nie tylko rozkłady zmiennych w każdym punkcie, ale również ich złożoną dynamikę w czasie. TimeGAN¹²⁴ (*time-series generative adversarial network*) jest modelem generatywnym opracowanym w celu zachowania dynamiki czasowej w danych syntetycznych. Model jest w stanie tworzyć realistyczne dane w postaci szeregów czasowych z różnych dziedzin, np. ceny akcji czy energii elektrycznej. Model wykorzystuje cztery komponenty: funkcję osadzania (*embedding function*), funkcję odzyskiwania (*recovery function*), generator sekwencji i dyskryminator sekwencji. Wszystkie komponenty są trenowane wspólnie, tak, aby TimeGAN jednocześnie uczył się kodować cechy, generować reprezentacje i iterować w czasie. Funkcje osadzania i odzyskiwania zapewniają odwzorowanie między cechami i przestrzenią ukrytą, umożliwiając sieci przeciwstawnej poznanie podstawowej dynamiki czasowej danych za pomocą reprezentacji o niższych wymiarach. Generator i dyskryminator działają w przestrzeni ukrytej, a ukryta dynamika danych rzeczywistych i syntetycznych jest synchronizowana przez nadzorowaną

¹²³ Tamże.

¹²⁴ Yoon, J., Jarrett, D., Schaar, M.V. (2019). *Time-series Generative Adversarial Networks*. <https://www.damtp.cam.ac.uk/user/dkj25/pdf/yoony2019time.pdf>

funkcję straty (*supervised loss*): model jest naprowadzany do uchwycenia warunkowego rozkładu czasowego w danych poprzez wykorzystanie danych rzeczywistych jako nadzoru. Dodatkowo TimeGAN jest uogólniany tak, aby mógł w tym samym czasie generować dane mieszane, tj. zarówno dane statyczne, jak i dane szeregów czasowych.

TimeGAN wykazuje spójne i znaczące ulepszenia w tworzeniu realistycznych szeregów czasowych w porównaniu z innymi metodami. W celu generowania wysokiej jakości danych szeregów czasowych dalsze prace nad rozwojem TimeGAN mają dotyczyć włączenia prywatności różnicowej do jego struktury.

Wybrane narzędzia do generowania danych syntetycznych

Oprócz omówionych wyżej metod generowania danych tabelarycznych w ostatnich latach pojawiło się wiele bibliotek dla języka programowania Python, nie tylko dostarczających gotowe dane syntetyczne, ale również zaopatrujących użytkowników w narzędzia umożliwiające ich tworzenie. W dalszej części zaprezentowano wybór takich narzędzi, które mogą być używane jako samodzielne biblioteki lub elementy bardziej wszechstronnych platform służących do udostępniania danych.

pydbgen

Jeśli wygenerowane dane syntetyczne mają zawierać kilka wspólnych zmiennych, z pewnym stopniem dostosowania, ale bez odzwierciedlenia poważnych zależności między nimi, to dobrym wyborem jest pydbgen¹²⁵. To proste narzędzie umożliwia losowe generowanie wskazanych przez użytkownika różnych typów danych (imię i nazwisko, adres, numer karty kredytowej, data, godzina, nazwa firmy, stanowisko, numer tablicy rejestracyjnej itp.) i zapisanie ich w obiekcie DataFrame Pandas lub jako tabela SQLite w pliku bazy danych bądź w pliku MS Excel. Za pomocą pydbgen i kilku linijek kodu można wygenerować zbiory danych o dowolnym rozmiarze z wieloma tabelami wypełnionymi uwarunkowanymi przez użytkownika losowymi danymi. Jako domyślnego generatora losowych danych pydbgen używa biblioteki Faker.

¹²⁵ Sarkar, T. *Random database/dataframe generator*. GitHub. Pobrano 16.09.2022 z <https://github.com/tirthajyoti/pydbgen>

Faker

Biblioteka Faker¹²⁶ służy do tworzenia sztucznych danych, które mogą posłużyć do testowania aplikacji, uruchamiania baz danych i utrzymywania anonimowości użytkowników. Został opracowany w celu uproszczenia generowania danych. W swoich strukturach wykorzystuje go wiele narzędzi z syntetycznym generatorem danych. Faker udostępnia wiele predefiniowanych metod, dzięki którym w krótkim czasie i bez większego wysiłku możliwe jest generowanie danych zgodnie z określonymi wymaganiami. Dodatkowo możliwe jest tworzenie różnych typów informacji specyficznych dla konkretnych krajów.

Mimesis

Biblioteka Mimesis¹²⁷ jest podobna do pydbgen i Fakera, ale jest bardziej kompletnym rozwiązaniem. To wysokowydajny generator sztucznych danych, zdecydowanie przewyższający Fakera pod względem szybkości działania, umożliwiając generowanie danych związanych z ludźmi, żywnością, transportem, adresami, sprzętem komputerowym i nie tylko. Oferuje różne lokalizacje językowe o wysokim stopniu specjalizacji, pozwalając na generowanie różnych typów informacji specyficznych dla konkretnych krajów. Biblioteka obsługuje szeroką gamę generatorów i obejmuje metody tworzenia kolumn kontekstowych, co czyni ją dobrym narzędziem do generowania poprawnych i różnorodnych zbiorów danych syntetycznych.

Mesa

Przy użyciu narzędzia Mesa¹²⁸ możliwe jest szybkie tworzenie modeli agentowych (*agent-based modeling*, ABM). Modelowanie agentowe polega na symulowaniu działań i interakcji agentów w celu oceny ich wpływu na system. Mesa pozwala na generowanie syntetycznych danych na podstawie złożonych scenariuszy: zapewnia sztuczne środowisko, w którym agenci mogą

126 Welcome to Faker's documentation! Pobrano 16.09.2022 z <https://faker.readthedocs.io/en/master/>

127 Mimesis: Fake Data Generator. Pobrano 16.09.2022 z <https://mimesis.name/en/latest/index.html>

128 Mesa: Agent-based modeling in Python 3+. Pobrano 16.09.2022 z <https://mesa.readthedocs.io/en/latest/index.html>

wchodzić w interakcje ze sobą i środowiskiem. Agenci mogą reprezentować żywe komórki, zachowanie zwierząt, a także pojedynczych osób czy nawet całych organizacji lub abstrakcyjnych jednostek. Głównym celem jest uzyskanie wyjaśniającego wglądu w zachowanie agentów przy określonym zestawie reguł, w szczególności zrozumienie, jak zachowują się poszczególne elementy systemu oraz jakie zachowania i efekty na poziomie systemu wynikają z ich interakcji. Przy użyciu wbudowanych podstawowych komponentów Mesa, lub niestandardowych implementacji, możliwe jest szybkie tworzenie modeli ABM, ich wizualizacja oraz analiza wyników za pomocą narzędzi dostępnych w Pythonie. Komponent DataCollector z modułu Data Collection zapewnia prosty sposób zbierania danych generowanych przez utworzone modele, w szczególności umożliwia zapis w postaci tabel przedstawiających zmienne gromadzone na poziomie modelu (*model-level variables*) i agentów (*agent-level variables*), czyli wartości obliczane z modelu lub każdego agenta w ich aktualnym stanie.

DataSynthesizer

Narzędzie DataSynthesizer¹²⁹ przekształca wejściowy zbiór danych i generuje strukturalnie i statystycznie podobny zbiór syntetyczny z silną gwarancją prywatności. System składa się z trzech modułów: DataDescriber, DataGenerator, i ModelInspector. Zbiór wejściowy jest najpierw przetwarzany przez moduł DataDescriber. Dziedziny i estymowane rozkłady atrybutów są wnioskowane z danych rzeczywistych i zapisywane w osobnym pliku, z którego korzysta DataGenerator. Dla każdego atrybutu o wartościach kategoryalnych DataDescriber oblicza rozkład częstotliwości wartości reprezentowanych w postaci wykresu słupkowego, z którego DataGenerator pobiera próbki podczas tworzenia zbioru syntetycznego. W przypadku atrybutów o wartościach liczbowych i zawierających daty/godziny DataDescriber reprezentuje rozkłady w postaci histogramów, z których DataGenerator pobiera próbki podczas generowania danych. Dodatkowo DataGenerator oferuje trzy różne sposoby generowania danych: losowe, niezależne lub skorelowane. Z kolei moduł ModelInspector zawiera kilka wbudowanych funkcji do sprawdzania podobieństwa między zbiorem wejściowym (rzeczywistym) a wyjściowym (syntetycznym). Rozmiar wyjściowego zbioru danych określa użytkownik.

129 Ping, H., Stoyanovich, J., Howe, B. (2017). *DataSynthesizer: Privacy-Preserving Synthetic Datasets*. <https://dl.acm.org/doi/pdf/10.1145/3085504.3091117>

Synthetic Data Vault

Środowisko Synthetic Data Vault (SDV)¹³⁰ to zbiór bibliotek do generowania danych syntetycznych. Pozwala użytkownikom modelować zbiory danych jednotabelowych, wielotabelowych, a nawet szeregów czasowych, aby później generować dane syntetyczne o tym samym formacie i właściwościach statystycznych, co zbiory rzeczywiste. SDV tworzy dane syntetyczne, stosując techniki matematyczne i modele głębokiego uczenia, takie jak dystrybuanty wielowymiarowych rozkładów prawdopodobieństwa lub sieci GAN, w szczególności CTGAN. Środowisko radzi sobie nawet ze zbiorami zawierającymi mieszane typy danych i brakujące wartości, więc wystarczy podać dane, dopasować do nich model, a następnie wygenerować dane syntetyczne. Ponadto SDV oferuje zestaw narzędzi do uruchamiania generatorów dla wielu zbiorów danych i zastosowania dedykowanych metryk, aby ocenić jakość wygenerowanych danych.

¹³⁰ Patki, N., Wedge, R., Veeramachaneni, K. (2016). *The Synthetic Data Vault*. <https://dai.lids.mit.edu/wp-content/uploads/2018/03/SDV.pdf>

Większość opisanych w niniejszym dokumencie modeli służących do generowania danych syntetycznych wykorzystuje metody oparte na sieciach neuronowych, a w szczególności na głębokim uczeniu. Metody te są również używane w procesie wykrywania danych podlegających anonimizacji – podczas rozpoznawania jednostek nazwanych (*named entity recognition*), czy też wykrywania twarzy na obrazach.

Sieci neuronowe modelują problem poprzez wykorzystanie połączonych węzłów i procesów decyzyjnych, które mogą stanowić małe części większych, bardziej złożonych mechanizmów rozwiązywania problemów. To wymaga jednak odpowiedniej mocy obliczeniowej. Warto zwrócić uwagę, że dzisiejsze CPU są projektowane jako jednostki uniwersalne, z ograniczonym zestawem operacji matematycznych. Mają wykonywać dowolną liczbę zadań typowych dla komputera, takich jak dostęp do pamięci masowej, rejestrowanie danych wejściowych, przenoszenie danych z pamięci podręcznej do pamięci itp. Oznacza to, że procesory mogą szybko przetaczać się pomiędzy zadaniami. Z kolei układy GPU – jako wyspecjalizowane jednostki, z dedykowaną pamięcią do renderowania obrazów i grafiki o wysokiej rozdzielczości – w porównaniu z CPU przeznaczają więcej tranzystorów na jednostki arytmetyczno-logiczne, a mniej na pamięć podręczną i kontrolę przepływu. Układy GPU składają się z procesorów SIMD przetwarzających niezależne instrukcje. To wszystko pozwala kartom graficznym przyspieszyć uczenie sieci neuronowych, a dodatkowym czynnikiem wspomagającym ten proces jest odczyt dużych pakietów danych. Układy GPU potrafią osiągnąć w tym aspekcie wyniki lepsze od CPU o dwa rzędy wielkości.

Uczenie maszynowe wymaga użycia dużych zbiorów danych, które są niezbędne do wytrenowania modelu. Główna zasada brzmi: im więcej danych jest dostarczonych do modelu podczas jego trenowania, tym lepszą dokładność przewidywania osiągnie. Jest to szczególnie istotne w przypadku sieci neuronowych i algorytmów głębokiego uczenia, które mają ogromne ilości parametrów, np. model podstawowy (*base*) BERT ma ich 110 milionów, duży (*large*) T5 – 770 milionów, GPT-2 – 1,5 miliarda, a DALL-E – 12 miliardów. Wielkość zbiorów danych jednorazowo ładowanych podczas trenowania algorytmów AI zależy od rozmiaru pamięci układu GPU. Im więcej pamięci, tym czas trenowania będzie krótszy. Należy dodać, że modele generatywne narzucają dodatkowe wymagania na układy GPU: np. aby wygenerować obraz przy użyciu modelu *stable diffusion* z wykorzystaniem 32-bitowych liczb zmiennoprzecinkowych, potrzebne jest 10 GB pamięci na karcie graficznej. Można zatem przyjąć, że tworząc środowisko do uczenia modeli sztucznej inteligencji,

należy zapewnić przynajmniej cztery karty graficzne z minimum 24 GB pamięci, które będą dedykowane specjalnie do uczenia maszynowego.

Każdy model AI trzeba poddać ocenie. W przypadku generowania tekstu odpowiednią metryką może być np. odległość cosinusowa, Rouge lub BLEU. Potrzebny jest jednak do tego tekst referencyjny, który posłuży do obliczenia jakości otrzymanych danych z modelu generatywnego. Niestety, trudno to zautomatyzować. Z tego względu, podczas projektowania systemu do anonimizacji i generowania danych syntetycznych, należy wziąć pod uwagę nie tylko środowiska programistyczne, testowe, preprodukcyjne i produkcyjne, ale również środowisko do testów akceptacyjnych. Będzie ono służyło przede wszystkim ocenie jakości anonimizacji i wytworzonych danych syntetycznych. Dodatkowo umożliwi zbieranie ocen od testerów – zarówno liczbowych, jak i opisowych – w celu udoskonalania usług.

Glosariusz

Analiza sentymentu

Znana również jako *opinion mining*. To proces wykorzystania przetwarzania języka naturalnego, analizy tekstu i statystyki do badania wydźwięku emocjonalnego (tzw. sentymentu) tekstu lub wypowiedzi, tj. identyfikowania stanów afektywnych i informacji subiektywnych, w szczególności w celu określenia stosunku emocjonalnego autora do opisywanego tematu (np. wydźwięk pozytywny, negatywny lub neutralny).

Architektura sieci

Definiuje, w jaki sposób skonstruowany jest model uczenia głębokiego, a co ważniejsze, do czego jest przeznaczony. Wybór architektury wpływa na dokładność sieci, określa rodzaj danych wejściowych i wyjściowych, a także definiuje kombinację warstw i sposób przepływu danych przez warstwy.

Atakujący

Osoba trzecia działająca wbrew celom ochrony prywatności, której poczynania prowadzą do przypadkowego lub umyślnego ujawnienia danych lub dostępu do przesyłanych, przechowywanych lub w inny sposób przetwarzanych danych.

Atrybut

Cecha lub właściwość jakiejś rzeczy, osoby lub zjawiska, wyróżniająca je spośród innych, którą można wykorzystać do opisanego jej stanu, wyglądu lub innych cech charakterystycznych.

Atrybut wrażliwy

Rodzaj atrybutu, który pozwala na identyfikację osób lub organizacji. Podlega szczególnej ochronie.

Autoenkoder (*autoencoder*)

Rodzaj sztucznej sieci neuronowej używanej do uczenia się efektywnego kodowania danych w sposób nienadzorowany. Dane wejściowe przekształcane

są w tzw. reprezentacje ukryte (*latent representations*) lub kodowania (*codings*), a następnie sieć stara się uzyskać wynik przypominający otrzymane na wejściu dane. W swojej najprostszej postaci autoenkoder składa się z dwóch części: kodera (*encoder*) i dekodera (*decoder*).

Computer vision

Widzenie komputerowe to dziedzina zajmująca się umożliwieniem maszynom lub systemom uzyskiwanie i zrozumienie istotnych informacji z cyfrowych obrazów, filmów lub innych danych wizualnych, oraz podejmowanie działań na podstawie tych informacji.

Dane osobowe

Wszelkie informacje o zidentyfikowanej lub możliwej do zidentyfikowania osobie fizycznej („osobie, której dane dotyczą”), które niezależnie lub w połączeniu ze sobą mogą prowadzić do bezpośredniego lub pośredniego zidentyfikowania tożsamości tej osoby.

Deduplikacja

Eliminowanie powtarzających się części w zbiorze danych.

Dekoder (*decoder*)

Część autoenkodera. Zadaniem dekodera jest przekształcanie danych z postaci reprezentacji ukrytej (*latent representation*) lub kodowania (*coding*) na dane wyjściowe.

Dostrajanie sieci (*fine-tuning*)

Technika polegająca na dostosowaniu wstępnie wytrenowanego modelu do wykonywania określonego zadania za pomocą własnego zestawu danych. Zakładając, że pierwotne zadanie jest podobne do nowego, użycie modelu sieci neuronowej, który został już zaprojektowany i przeszkolony na pewnym zbiorze uczącym, pozwala wykorzystać zdobytą przez niego wiedzę, bez konieczności tworzenia jej od podstaw.

Osadzanie (*embedding*)

Reprezentacja rzeczywistych obiektów i relacji, wyrażona jako wektor

liczbowy w stosunkowo małowymiarowej przestrzeni, na którą można tłumaczyć wektory wysokowymiarowe. Wektory te zachowują część semantyki danych wejściowych, umieszczając podobne semantycznie dane wejściowe blisko siebie w przestrzeni wektorowej.

Entropia

Miara nieokreśloności i stopnia nieuporządkowania elementów i stanów znajdujących się w pewnym zbiorze.

Funkcja straty (*loss function*)

Inaczej funkcja kosztu. Jest to metoda oceny modelowania zbioru danych przez algorytm. Funkcja sprawdza, jak bardzo szacunkowa wartość różni się od wartości prawdziwej.

Hiperparametr

Parametr, którego wartość jest wykorzystywana do kontrolowania procesu uczenia się.

Klasa równoważności

Podzbiór danych zawierający elementy, które mają być przetwarzane w ten sam sposób.

Koder (*encoder*)

Część autoenkodera. Zadaniem kodera jest przekształcanie danych wejściowych do reprezentacji ukrytej (*latent representation*) lub kodowania (*coding*).

Mechanizm uwagi (*attention mechanism*)

Technika przetwarzania danych w sieciach neuronowych, polegająca na rozbiciu skomplikowanych zadań na mniejsze obszary uwagi, które są przetwarzane sekwencyjnie, co pozwala sieci selektywnie skoncentrować się na wartościowych częściach sekwencji wejściowej, a tym samym nauczyć się powiązania między nimi. Mechanizm uwagi umożliwia uwzględnienie zależności między wszystkimi elementami sekwencji, niezależnie od odległości, która je dzieli.

Mechanizm rekurencji segmentów ze schematem kodowania względnego

Techniki zawarte w architekturze Transformer-XL. Mechanizm rekurencji segmentów (*segment-level recurrence*) umożliwia modelowi ponowne wykorzystanie poprzednich stanów ukrytych w czasie trenowania, rozwiązując zarówno kwestie kontekstu o stałej długości, jak i fragmentacji kontekstu (reprezentacje obliczone dla poprzedniego segmentu są stałe i przechowywane w pamięci, aby można je było ponownie wykorzystać jako rozszerzony kontekst, gdy model przetwarza następny nowy segment). Mechanizm zwany schematem kodowania względnego (*relative positional encoding scheme*) umożliwia poprawne zastosowanie mechanizmu rekurencji segmentów, tzn. pozwala na właściwe ponowne wykorzystanie poprzednich stanów ukrytych.

Mechanizm PLM (*permutation language modeling*)

Mechanizm, który służy do przechwytywania kontekstu dwukierunkowego poprzez trenowanie modelu autoregresyjnego na wszystkich możliwych permutacjach słów w zdaniu. Zadaniem mechanizmu PLM jest przewidywanie kolejnego tokenu na podstawie poprzedzającego kontekstu, ale zamiast w kolejności sekwencyjnej (jak jest w przypadku tradycyjnych modeli językowych), przewiduje tokeny w kolejności losowej.

Performance

Zdolność modelu do dokładnego wykonywania zadania nie tylko na danych uczących, ale także na danych produkcyjnych, zwłaszcza wtedy, gdy model jest faktycznie wdrażany w środowisku produkcyjnym.

Problemy o wiedzy zerowej (*zero-knowledge problems*)

Obejmują zadania, gdzie wstępnie wytrenowany model jest używany do oceny danych testowych klas, które nie były używane podczas jego szkolenia.

Propagacja wsteczna (*backpropagation; backward propagation of errors*)

Algorytm nadzorowanego uczenia sztucznych sieci neuronowych do precyzyjnego dostosowania każdej wagi w sieci proporcjonalnie do jej wkładu do błędu ogólnego. Mając sztuczną sieć neuronową i funkcję błędu, metoda oblicza gradient funkcji błędu względem wag sieci neuronowej. Obliczanie gradientu przebiega wstecz, przy czym najpierw obliczany jest gradient ostatniej warstwy wag, a jako ostatni obliczany jest gradient pierwszej warstwy

wag. Częściowe obliczenia gradientu z jednej warstwy są ponownie wykorzystywane do obliczania gradientu dla poprzedniej warstwy. Ten wsteczny przepływ informacji o błędzie umożliwia wydajniejsze obliczenie gradientu w każdej warstwie w porównaniu z naiwnym podejściem obliczania gradientu każdej warstwy oddzielnie, co w rezultacie prowadzi do poprawy dokładności wyjść sieci neuronowej.

Przekształcenie afiniczne

Rodzaj przekształcenia geometrycznego, który zachowuje kolinearność (współliniowość) oraz stosunki odległości między punktami na prostej. Do przekształceń afinicznych należą m.in. translacja (przesunięcie), skalowanie, obrót i odbicie.

Seq2Seq (*sequence-to-sequence*)

Rodzina modeli stosowanych w zadaniach przewidywania sekwencji, takich jak modelowanie języka i tłumaczenie maszynowe. Seq2Seq zmienia jedną sekwencję w inną sekwencję za pomocą LSTM lub rzadziej, rekurencyjnej sieci neuronowej (RNN), aby uniknąć problemu zanikającego gradientu. Kontekstem dla każdego elementu są dane wyjściowe z poprzedniego kroku. Podstawowymi komponentami są koder i dekodek. Koder odczytuje sekwencję wejściową krok po kroku, aby zamienić każdy element w odpowiadający mu ukryty wektor zawierający element i jego kontekst. Dekoder odwraca ten proces, zamieniając wektor w sekwencję wyjściową, używając poprzedniego wyjścia jako kontekstu wejściowego.

Uczenie przeciwstawne (*adversarial learning*)

Technika służąca do trenowania sieci neuronowych w zakresie wykrywania intencjonalnie wprowadzających w błąd danych lub zachowań. Celem tego rodzaju uczenia jest nie tylko wykrywanie złych danych wejściowych, ale również zapobiegawcze lokalizowanie nieprawidłowości i tworzenie bardziej elastycznych algorytmów uczenia.

Uczenie transferowe (*transfer learning*)

Technika polegająca na ponownym wykorzystaniu części wcześniej wytrenowanego modelu do stworzenia nowej sieci realizującej inne, ale podobne zadanie. Założenie jest takie, że wiedza pozyskana z modelu użytego w jednym zadaniu może być ponownie wykorzystana jako punkt wyjścia dla innego

zadania. Stosowanie uczenia transferowego jest bardzo popularne, ponieważ znacznie skraca czas potrzebny na inżynierię cech (*feature engineering*) i umożliwia trenowanie sieci przy stosunkowo niewielkiej ilości danych.

Wyszukiwanie zachłanne (*greedy algorithm*)

Algorytm, który dokonuje lokalnie optymalnych wyborów na każdym etapie, próbując znaleźć optimum globalne, czyli optymalny sposób rozwiązania całego problemu.

Literatura

Techniki depersonalizacji danych

Grupa Robocza Artykułu 29 (przyjęta w dniu 10 kwietnia 2014 r.). *Opinia 05/2014 w sprawie technik anonimizacji*. https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2014/wp216_pl.pdf

Mivule, K. (2012). *Utilizing Noise Addition for Data Privacy, an Overview*. <https://arxiv.org/ftp/arxiv/papers/1309/1309.3958.pdf>

International Household Survey Network. (2019). *Anonymization Methods*. SDC Practice Guide. Pobrano 28.06.2022 z https://sdcpractice.readthedocs.io/en/latest/anon_methods.html#anonymization-methods

Sweeney, L. (2002). *Achieving k-anonymity privacy protection using generalization and suppression*. <https://dataprivacylab.org/dataprivacy/projects/kanonymity/kanonymity2.pdf>

Mendes, R., Vilela, J. (2017). *Privacy-Preserving Data Mining: Methods, Metrics and Applications*. <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7950921>

Ayala-Rivera, V., McDonagh, P., Cerqueus, T., Murphy, L. (2014). *A Systematic Comparison and Evaluation of k-Anonymization Algorithms for Practitioners*. <http://www.tdp.cat/issues11/tdp.a169a14.pdf>

Patel, T., Amin, K. (2019). *A Study on k-anonymity, l-diversity, and t-closeness Techniques of Privacy Preservation Data Publishing*. <http://www.ijirst.org/articles/IJIRSTV6I6015.pdf>

Machanavajjhala, A., Kifer, D., Gehrke, J., Venkatasubramanian, M. (2007). *l-Diversity: Privacy beyond k-anonymity*. https://personal.utdallas.edu/~mxk055100/courses/privacy08f_files/ldiversity.pdf

Gehrke, J., Kifer, D., Machanavajjhala, A. (2011). *l-Diversity*. Springer. Pobrano 05.07.2022 z https://link.springer.com/referenceworkentry/10.1007/978-1-4419-5906-5_899

Li, N., Li, T., Venkatasubramanian, S. (2007). *t-Closeness: Privacy Beyond k-Anonymity and l-Diversity*. https://www.cs.purdue.edu/homes/ninghui/papers/t_closeness_icde07.pdf

Dwork, C. (2006). *Differential Privacy*. <http://www.audentia-gestion.fr/MICROSOFT/dwork.pdf>

Dwork, C., Roth, A . (2014). *The Algorithmic Foundations of Differential Privacy*. <https://www.cis.upenn.edu/~aaroht/Papers/privacybook.pdf>

King, T. (2019, 28 marca). *80 Percent of Your Data Will Be Unstructured in Five Years*. Solutions Review. Pobrano 22.07.2022 z <https://solutionsreview.com/data-management/80-percent-of-your-data-will-be-unstructured-in-five-years/>

Rozporządzenie Parlamentu Europejskiego i Rady (UE) 2016/679 z dnia 27 kwietnia 2016 r. w sprawie ochrony osób fizycznych w związku z przetwarzaniem danych osobowych i w sprawie swobodnego przepływu takich danych oraz uchylenia dyrektywy 95/46/WE (ogólne rozporządzenie o ochronie danych). <https://uodo.gov.pl/pl/131/224>

Littlejohn, S., Michael C. (2008). *Chapter 12 - Understanding Cyber-crime Prevention*. <https://www.sciencedirect.com/science/article/pii/B9781597492768000121>

Bellare, M., Boldyreva, A ., O 'Neill, A . (2007). *Deterministic and Efficiently Searchable Encryption*. <https://eprint.iacr.org/2006/186.pdf>

PCI Security Standards Council. (2015). *Tokenization Product Security Guidelines*. https://listings.pcisecuritystandards.org/documents/Tokenization_Product_Security_Guidelines.pdf

Turner, C. (2020, 14 grudnia). *Tokenization vs. Encryption: Pros and Cons*. eSecurity Planet. Pobrano 13.07.2022 z <https://www.esecurityplanet.com/threats/tokenization-vs-encryption/>

Inne (brak w przypisach dolnych):

Krchova, I . (2020, 25 listopada). *3 reasons why organizations are moving away from legacy data masking*. Mostly AI. Pobrano 28.07.2022 z <https://mostly.ai/blog/3-reasons-to-drop-classic-anonymization-and-upgrade-to-synthetic-data/>

SYNTHESIZED. (2019, 28 listopada). *Three Common Misconceptions about Synthetic and Anonymised Data*. SYNTHESIZED. Pobrano 28.07.2022 z <https://www.synthesized.io/post/three-common-misconceptions-about-synthetic-and-anonymised-data>

Riemann, R. *Synthetic Data*. European Data Protection Supervisor. Pobrano 28.07.2022 z https://edps.europa.eu/press-publications/publications/techsonar/synthetic-data_en

Data Protection Commission. (2019). *Guidance on Anonymisation and Pseudonymisation*. <https://www.dataprotection.ie/sites/default/files/uploads/2019-06/190614%20Anonymisation%20and%20Pseudonymisation.pdf>

Mivule, K. (2013). *Utilizing Noise Addition for Data Privacy, an Overview*. <https://arxiv.org/ftp/arxiv/papers/1309/1309.3958.pdf>

Information Security and Privacy Office. *Guidelines for Anonymization & Pseudonymization*. Pobrano 05.07.2022 z <https://ispo.newschool.edu/guidelines/anonymization-pseudonymization/#guidelines-for-anonymization-pseudonymization>

Dwork, C. (2008). *Differential Privacy: A Survey of Results*. https://web.cs.ucdavis.edu/~franklin/ecs289/2010/dwork_2008.pdf

Near, J., Darais, D., Boeckl, K. (2020, 27 lipca). *Differential Privacy for Privacy-Preserving Data Analysis: An Introduction to our Blog Series*. NIST. Pobrano 07.07.2022 z <https://www.nist.gov/blogs/cybersecurity-insights/differential-privacy-privacy-preserving-data-analysis-introduction-our>

Information Commissioner. (2012). *Anonymisation: managing data, protection risk, code of practice*. <https://ico.org.uk/media/1061/anonymisation-code.pdf>

Personal Data Protection Commission Singapore. (2018). *Guide To Basic Data Anonymisation Techniques*. https://iapp.org/media/pdf/resource_center/Guide_to_Anonymisation.pdf

Nagaraj, N. (2020, 30 września). *Face Anonymization: A survey of what works and what doesn't*. ML6 blog. Pobrano 22.07.2022 z <https://blog.ml6.eu/face-anonymization-a-comparison-66da5088d030>

Lison, P., Pilán, I., Sánchez, D., Batet, M., Øvrelid, L. (2021). *Anonymisation Models for Text Data: State of the art, Challenges and Future Directions*. <https://aclanthology.org/2021.acl-long.323.pdf>

Münch, D., Grossefinger, A.K., Krempel, E., Hebel, M., Arens, M. (2019). *Data Anonymization for Data Protection on Publicly Recorded Data*. https://link.springer.com/chapter/10.1007/978-3-030-34995-0_23

Kuang, Z., Liu, H., Yu, J., Tian, A., Wang, L., Fan, J., Babaguchi, N. (2021). *Effective De-identification Generative Adversarial Network for Face Anonymization*. <https://dl.acm.org/doi/pdf/10.1145/3474085.3475464>

Weitzenboeck, E., Lison, P., Cyndecka, M., Langford, M. (2022). *The GDPR and unstructured data: is anonymization possible?*. <https://academic.oup.com/idpl/article/12/3/184/6552802>

Simi, Ms., Nayaki, K., Elayidom, M. (2017). *An Extensive Study on Data Anonymization Algorithms Based on K-Anonymity*. <https://iopscience.iop.org/article/10.1088/1757-899X/225/1/012279/pdf>

Slijepčević D., Henzl, M., Klausner, L.D., Dam, T., Kieseberg, P., Zeppelzauer, M. (2021). *k-Anonymity in practice: How generalisation and suppression affect machine learning classifiers*. <https://www.sciencedirect.com/science/article/pii/S0167404821003126>

Weisstein, E.W., *Hash Function*. MathWorld--A Wolfram Web Resource. Pobrano 25.07.2022 z <https://mathworld.wolfram.com/HashFunction.html>

Personal data pseudonymization: GDPR pseudonymization what and how. i-scoop. Pobrano 26.07.2022 z <https://www.i-scoop.eu/gdpr/pseudonymization/>

Agencia Española de Protección de Datos. (2019). *Introduction To The Hash Function As A Personal Data Pseudonymization Technique*. https://edps.europa.eu/sites/default/files/publication/19-10-30_aepd-edps_paper_hash_final_en.pdf

Hradec, J., Craglia, M., Di Leo, M., De Nigris, S., Ostlaender, N., Nicholson, N. (2022). *Multipurpose synthetic population for policy applications*. <https://publications.jrc.ec.europa.eu/repository/handle/JRC128595>

Dankar, F.K., Ibrahim, M. (2021). *Fake It Till You Make It: Guidelines for Effective Synthetic Data Generation*. <https://www.mdpi.com/2076-3417/11/5/2158>

Gooding, M. (2021, 28 września). Synthetic data may not be AI's privacy silver bullet. Tech Monitor. Pobrano 28.07.2022 z <https://techmonitor.ai/technology/ai-and-automation/synthetic-data-may-not-be-ais-privacy-silver-bullet>

Kearns, M., Roth, A. (2020, 13 stycznia). Ethical algorithm design should guide technology regulation. Brookings. Pobrano 28.07.2022 z <https://www.brookings.edu/research/ethical-algorithm-design-should-guide-technology-regulation/>

Generowanie danych syntetycznych

Dane tekstowe

Meister, C., Pimentel, T., Wiher, G., Cotterell, R. (2022). *Locally Typical Sampling*. <https://arxiv.org/pdf/2202.00666.pdf>

Bikel, D.M., Schwartz, R.M., Weischedel, R.M. (2004). *An Algorithm that Learns What's in a Name*. <http://people.csail.mit.edu/mcollins/6864/slides/bikel.pdf>

Sherstinsky, A. (2020). *Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network*. <https://arxiv.org/pdf/1808.03314.pdf>

Pascanu, R., Mikolov, T., Bengio, Y. (2012). *On the difficulty of training Recurrent Neural Networks*. <http://proceedings.mlr.press/v28/pascanu13.pdf>

Hochreiter, S., Schmidhuber, J. (1997). *Long Short-term Memory*. <http://www.bioinf.jku.at/publications/older/2604.pdf>

Santhanam, S. (2018). *Context based Text-generation using LSTM networks*. <https://arxiv.org/pdf/2005.00048.pdf>

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, L., Polosukhin, I. (2017). *Attention Is All You Need*. <https://arxiv.org/pdf/1706.03762.pdf>

Topal, M.O ., Bas, A ., Heerden, I .V. (2021). *Exploring Transformers in Natural Language Generation: GPT, BERT, and XLNet*. <https://arxiv.org/ftp/arxiv/papers/2102/2102.08036.pdf>

Radford, A ., Narasimhan, K., Salimans, T. Sutskever, I . (2018). *Improving language understanding by generative pre-training*. https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf

Radford, A ., Wu, J., Child, R., Luan, D., Amodei, D. Sutskever, I . (2018). *Language Models are Unsupervised Multitask Learners*. https://d4mucfpksyvv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A ., Shyam, P., Sastry, G., Askell, A ., Agarwal, S., Herbert-Voss, A ., Krueger, G., Henighan, T., Child, R., Ramesh, A ., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A ., Sutskever, I . Amodei, D. (2020). *Language Models are Few-Shot Learners*. <https://arxiv.org/pdf/2005.14165.pdf>

Devlin, J., Chang, M., Lee, K., Toutanova, K. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. <https://arxiv.org/pdf/1810.04805.pdf>

Sanh, V., Debut, L., Chaumond, J., Wolf, T. (2019). *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. <https://arxiv.org/pdf/1910.01108.pdf>

Lan, Z ., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R. (2020). *ALBERT: A Lite BERT for Self-supervised Learning of Language Representations*. <https://arxiv.org/pdf/1909.11942.pdf>

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O ., Lewis, M., Zettlemoyer, L., Stoyanov, V. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. <https://arxiv.org/pdf/1907.11692.pdf>

Hsu, W ., Bolte, B., Tsai, Y.H., Lakhotia, K., Salakhutdinov, R., Mohamed, A . (2021). *HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units*. <https://arxiv.org/pdf/2106.07447.pdf>

Rybak, P., Mroczkowski, R., Tracz, J., Gawlik, I. (2020). *KLEJ: Comprehensive Benchmark for Polish Language Understanding*. <https://arxiv.org/pdf/2005.00630.pdf>

Mroczkowski, R., Rybak, P., Wróblewska, A., Gawlik, I. (2021). *HerBERT: Efficiently Pretrained Transformer-based Language Model for Polish*. <https://arxiv.org/pdf/2105.01735.pdf>

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V. (2020). *Unsupervised Cross-lingual Representation Learning at Scale*. <https://arxiv.org/pdf/1911.02116.pdf>

Dadas, S., Peretkiewicz, M., Poświata, R. (2020). *Pre-training Polish Transformer-Based Language Models at Scale*. <https://arxiv.org/pdf/2006.04229.pdf>

Yang, Z., Dai, Z., Yang, Y., Carbonell, J.G., Salakhutdinov, R., Le, Q.V. (2019). *XLNet: Generalized Autoregressive Pretraining for Language Understanding*. <https://arxiv.org/pdf/1906.08237.pdf>

Dai, Z., Yang, Z., Yang, Y., Carbonell, J.G., Le, Q.V., Salakhutdinov, R. (2019). *Transformer-XL: Attentive Language Models beyond a Fixed-Length Context*. <https://arxiv.org/pdf/1901.02860.pdf>

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J. (2020). *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. <https://arxiv.org/pdf/1910.10683.pdf>

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L. (2020). *BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension*. <https://arxiv.org/pdf/1910.13461.pdf>

Radev, D.R., Zhang, R., Rau, A., Sivaprasad, A., Hsieh, C., Rajani, N.F., Tang, X., Vyas, A., Verma, N., Krishna, P., Liu, Y., Irwanto, N., Pan, J., Rahman, F., Zaidi, A., Mutuma, M., Tarabar, Y., Gupta, A., Yu, T., Tan, Y.C., Lin, X.V., Xiong, C., Socher, R. (2021). *DART: Open-Domain Structured Data Record to Text Generation*. <https://arxiv.org/pdf/2007.02871.pdf>

Keymanesh, M., Benton, A., Dredze, M. (2022). *What Makes Data-to-Text Generation Hard for Pretrained Language Models?*. <https://arxiv.org/pdf/2205.11505.pdf>

Zhang, Y., Sun, S., Galley, M., Chen, Y., Brockett, C., Gao, X., Gao, J., Liu, J., Dolan, W. B. (2020). *DIALOGPT : Large-Scale Generative Pre-training for Conversational Response Generation*. <https://arxiv.org/pdf/1911.00536.pdf>

Adiwardana, D., Luong, M.Y., So, D.R., Hall, J., Fiedel, N., Thoppilan, R., Yang, Z., Kulshreshtha, A., Nemade, G., Lu, Y., Le, Q.V. (2020). *Towards a Human-like Open-Domain Chatbot*. <https://arxiv.org/pdf/2001.09977.pdf>

Bao, S., He, H., Wang, F., Wu, H., Wang, H. (2020). *PLATO: Pre-trained Dialogue Generation Model with Discrete Latent Variable*. <https://arxiv.org/pdf/1910.07931.pdf>

Roller, S., Dinan, E., Goyal, N., Ju, D., Williamson, M., Liu, Y., Xu, J., Ott, M., Shuster, K., Smith, E.M., Boureau, Y., Weston, J. (2021). *Recipes for Building an Open-Domain Chatbot*. <https://arxiv.org/pdf/2004.13637.pdf>

Pan, L., Lei, W., Chua, T., Kan, M. (2019). *Recent Advances in Neural Question Generation*. <https://arxiv.org/pdf/1905.08949.pdf>

Liu, B., Wei, H., Niu, D., Chen, H., He, Y. (2020). *Asking Questions the Human Way: Scalable Question-Answer Generation from Text Corpus*. <https://arxiv.org/pdf/2002.00748.pdf>

Yin, J., Jiang, X., Lu, Z., Shang, L., Li, H., Li, X. (2016). *Neural Generative Question Answering*. <https://www.ijcai.org/Proceedings/16/Papers/422.pdf>

Andreas, J., Rohrbach, M., Darrell, T., Klein, D. (2016). *Learning to Compose Neural Networks for Question Answering*. <https://arxiv.org/pdf/1601.01705.pdf>

Li, Z., Jiang, X., Shang, L., Li, H. (2018). *Paraphrase Generation with Deep Reinforcement Learning*. <https://aclanthology.org/D18-1421.pdf>

Fan, A., Lewis, M., Dauphin, Y. (2018). *Hierarchical Neural Story Generation*. <https://arxiv.org/pdf/1805.04833v1.pdf>

Riedl, M. (2021, 21 sierpnia). *An Introduction to AI Story Generation*. The Gradient. Pobrano 22.08.2022 z <https://thegradient.pub/an-introduction-to-ai-story-generation/>

Inne (brak w przypisach dolnych):

Markov Model of Natural Language. Pobrano 16.08.2022 z <https://www.cs.princeton.edu/courses/archive/fall13/cos126/assignments/markov.html>

Powell, V., Lehe, L. *Markov Chains Explained Visually*. Setosa. Pobrano 16.08.2022 z <https://setosa.io/ev/markov-chains/>

Becker, C., Wagner, J., He, B. (2020). *Introduction: Transfer Learning for NLP*. Pobrano 17.08.2022 z https://slds-lmu.github.io/seminar_nlp_ss20/introduction-transfer-learning-for-nlp.html

Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., Raffel, C. (2021). *mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer*. <https://aclanthology.org/2021.naacl-main.41.pdf>

Iyyer, M., Boyd-Graber, J.L., Claudino, L.M., Socher, R., Daumé, H. (2014). *A Neural Network for Factoid Question Answering over Paragraphs*. <https://aclanthology.org/D14-1070.pdf>

Bansal, A., Eberhart, Z., Wu, L., McMillan, C. (2021). *A Neural Question Answering System for Basic Questions about Subroutines*. <https://arxiv.org/pdf/2101.03999.pdf>

Guo, Z., Huang, Z., Zhu, K.Q., Chen, G., Zhang, K., Chen, B., Huang, F. (2021). *Automatically Paraphrasing via Sentence Reconstruction and Round-trip Translation*. <https://www.ijcai.org/proceedings/2021/0525.pdf>

Aithal, S.G., Rao, A.B., Singh, S. (2021). *Automatic question-answer pairs generation and question similarity mechanism in question answering system*. <https://link.springer.com/article/10.1007/s10489-021-02348-9>

Roy, A. (2021, 23 marca). *Progress and Challenges in Long-Form Open-Domain Question Answering*. Google AI Blog. Pobrano 02.08.2022 z <https://ai.googleblog.com/2021/03/progress-and-challenges-in-long-form.html>

Chen, H., Liu, X., Yin, D., Tang, J. (2017). *A Survey on Dialogue Systems: Recent Advances and New Frontiers*. https://www.kdd.org/exploration_files/19-2-Article3.pdf

Sun, B., Li, K. (2021). *Neural Dialogue Generation Methods in Open Domain: A Survey*. <https://www.atlantis-press.com/journals/nlpr/125954217>

Garbacea, C., Mei, Q. (2020). *Neural Language Generation: Formulation, Methods, and Evaluation*. <https://arxiv.org/pdf/2007.15780.pdf>

Li, J., Tang, T., Zhao, W .X., Wen, J. (2021). *Pretrained Language Models for Text Generation: A Survey*. <https://arxiv.org/pdf/2105.10311.pdf>

Xie, Z . (2018). *Neural Text Generation: A Practical Guide*. <https://cs.stanford.edu/~zxie/textgen.pdf>

Celikyilmaz, A ., Clark, E., Gao, J. (2020). *Evaluation of Text Generation: A Survey*. <https://arxiv.org/pdf/2006.14799.pdf>

Chan, Y., Fan, Y. (2019). *A Recurrent BERT-based Model for Question Generation*. <https://aclanthology.org/D19-5821.pdf>

Ni, J., Young, T., Pandelea, V., Xue, F., Adiga, V.V., Cambria, E. (2022). *Recent Advances in Deep Learning Based Dialogue Systems: A Systematic Survey*. <https://arxiv.org/pdf/2105.04387.pdf>

Luu, A ., Shah, D.J., Barzilay, R. (2020). *Capturing Greater Context for Question Generation*. <https://arxiv.org/pdf/1910.10274.pdf>

Alberti, C., Lee, K., Collins, M. (2019). *A BERT Baseline for the Natural Questions*. <https://arxiv.org/pdf/1901.08634.pdf>

Puri, R., Spring, R., Patwary, M., Shoeybi, M., Catanzaro, B. (2020).

Training Question Answering Models From Synthetic Data. <https://arxiv.org/pdf/2002.09599.pdf>

Dane obrazowe

Kingma, D.P., Welling, M. (2014). *Auto-Encoding Variational Bayes*. <https://arxiv.org/pdf/1312.6114v10.pdf>

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y. (2014). *Generative Adversarial Networks*. <https://arxiv.org/pdf/1406.2661.pdf>

Durall, R., Chatzimichailidis, A., Labus, P., Keuper, J. (2021). *Combating Mode Collapse in GAN training: An Empirical Analysis using Hessian Eigenvalues*. <https://arxiv.org/pdf/2012.09673.pdf>

Mirza, M., Osindero, S. (2014). *Conditional Generative Adversarial Nets*. <https://arxiv.org/pdf/1411.1784.pdf>

Radford, A., Metz, L., Chintala, S. (2016). *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. <https://arxiv.org/pdf/1511.06434.pdf>

Zhu, J., Park, T., Isola, P., Efros, A. (2017). *Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks*. <https://arxiv.org/pdf/1703.10593v7.pdf>

Karras, T., Laine, S., Aila, T. (2019). *A Style-Based Generator Architecture for Generative Adversarial Networks*. <https://arxiv.org/pdf/1812.04948.pdf>

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, L., Polosukhin, I. (2017). *Attention Is All You Need*. <https://arxiv.org/pdf/1706.03762.pdf>

Chen, M., Radford, A., Wu, J., Jun, H., Dhariwal, P., Luan, D., Sutskever, I. (2020). *Generative Pretraining From Pixels*. https://cdn.openai.com/papers/Generative_Pretraining_from_Pixels_V2.pdf

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I. (2021). *Zero-Shot Text-to-Image Generation*. <https://arxiv.org/pdf/2102.12092.pdf>

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M. (2022). *Hierarchical Text-Conditional Image Generation with CLIP Latents*. <https://arxiv.org/pdf/2204.06125.pdf>

Sohl-Dickstein, J., Weiss, E.A., Maheswaranathan, N., Ganguli, S. (2015). *Deep Unsupervised Learning using Nonequilibrium Thermodynamics*. <https://arxiv.org/pdf/1503.03585.pdf>

Dhariwal, P., Nichol, A. (2021). *Diffusion Models Beat GANs on Image Synthesis*. <https://arxiv.org/pdf/2105.05233v4.pdf>

Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., Chen, M. (2021). *GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models*. <https://arxiv.org/pdf/2112.10741.pdf>

Radford, A., Kim, J., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I. (2021). *Learning Transferable Visual Models From Natural Language Supervision*. <https://arxiv.org/pdf/2103.00020.pdf>

Ramesh, A. How DALL-E 2 Works. Pobrano 06.09.2022 z <http://adityaramesh.com/posts/dalle2/dalle2.html>

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M. (2022). *Hierarchical Text-Conditional Image Generation with CLIP Latents*. <https://arxiv.org/pdf/2204.06125.pdf>

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, S.K., Ayan, B.K., Mahdavi, S.S., Lopes, R.G., Salimans, T., Ho, J., Fleet, D.J., Norouzi, M. (2022). *Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding*. <https://arxiv.org/pdf/2205.11487.pdf>

Google Research, Brain Team. *Imagen: unprecedented photorealism × deep level of language understanding*. Pobrano 07.09.2022 z <https://imagen.research.google/>

Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H. (2016). *Generative Adversarial Text to Image Synthesis*. <http://proceedings.mlr.press/v48/reed16.pdf>

Frolov, S., Hinz, T., Raue, F., Hees, J., Dengel, A.R. (2021). *Adversarial Text-to-Image Synthesis: A Review*. <https://arxiv.org/pdf/2101.09983.pdf>

Zhao, B., Meng, L., Yin, W., Sigal, L. (2019). *Image Generation from Layout*. <https://arxiv.org/pdf/1811.11389.pdf>

Charles Burlin, C., Le Calonnec, Y., Duperier, L. (2017). *Deep Image Inpainting*. <http://cs231n.stanford.edu/reports/2017/pdfs/328.pdf>

Elharrouss, O ., Almaadeed, N., Al-Maadeed, S.A ., Akbari, Y. (2019). *Image In-painting: A Review*. <https://arxiv.org/ftp/arxiv/papers/1909/1909.06399.pdf>

Isola, P., Zhu, J., Zhou, T., Efros, A .A . (2017). *Image-to-Image Translation with Conditional Adversarial Networks*. <https://arxiv.org/pdf/1611.07004.pdf>

Pang, Y., Lin, J., Qin, T., Chen, Z . (2022). *Image-to-Image Translation: Methods and Applications*. <https://arxiv.org/pdf/2101.08629.pdf>

Yi, X., Walia, E., Babyn, P.S. (2019). *Generative Adversarial Network in Medical Imaging: A Review*. <https://arxiv.org/pdf/1809.07294.pdf>

Koshino, K., Werner, R.A ., Pomper, M.G., Bundschuh, R.A ., Toriumi, F., Higu-chi, T., Rowe, S.P. (2021). *Narrative review of generative adversarial networks in medical and molecular imaging*. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8246192/>

Baur, C., Albarqouni, S., Navab, N. (2018). *Generating Highly Realistic Images of Skin Lesions with GANs*. <https://arxiv.org/pdf/1809.01410.pdf>

Costa, P., Galdran, A ., Meyer, M.I ., Niemeijer, M., Abràmoff, M., Mendonça, A .M., Campilho, A . (2017). *End-to-End Adversarial Retinal Image Synthesis*. http://eprints.bournemouth.ac.uk/34883/1/tmi_2017.pdf

Zhao,H., Li, H., Maurer-Stroh, S., Cheng, L. (2018). *Synthesizing retinal and neuronal images with generative adversarial nets*. <https://www.sciencedirect.com/science/article/pii/S1361841518304596>

Han, C., Rundo, L., Araki, R., Furukawa, Y., Mauri, G., Nakayama, H., Hayashi, H. (2019). *Infinite Brain MR Images: PGGAN-based Data Augmentation for Tumor Detection*. <https://arxiv.org/pdf/1903.12564.pdf>

Chuquicusma, M.M., Hussein, S., Burt, J.R., Bagci, U . (2018). *How to fool radiologists with generative adversarial networks? A visual turing test for lung cancer diagnosis*. <https://arxiv.org/pdf/1710.09762.pdf>

Radford, A ., Metz, L., Chintala, S. (2016). *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. <https://arxiv.org/pdf/1511.06434.pdf>

Arjovsky, M., Chintala, S., Bottou, L. (2017). Wasserstein GAN. <https://arxiv.org/pdf/1701.07875.pdf>

Karras, T., Aila, T., Laine, S., Lehtinen, J. (2018). *Progressive Growing of GANs for Improved Quality, Stability, and Variation*. <https://arxiv.org/pdf/1710.10196.pdf>

Inne (brak w przypisach dolnych):

Ramesh, A ., Pavlov, M., Goh, G., Gray, S., Misra, V., Mishkin, P., Krueger, G., Agarwal, S. Sutskever, I . (2021, 5 stycznia). *DALL-E: Creating Images from Text*. OpenAI. Pobrano 05.09.2022 z <https://openai.com/blog/dall-e/>

OpenAI. *DALL-E 2*. Pobrano 06.09.2022 z <https://openai.com/dall-e-2/>

OpenAI. *Image GPT*. Pobrano 07.09.2022 z <https://openai.com/blog/image-gpt/>

Qiao, T., Zhang, J., Xu, D., Tao, D. (2019). *Learn, Imagine and Create: Text-to-Image Generation from Prior Knowledge*. <https://proceedings.neurips.cc/paper/2019/file/d18f655c3fce66ca401d5f38b48c89af-Paper.pdf>

Singh, N.K., Raza, K. (2020). *Medical Image Generation using Generative Adversarial Networks*. <https://arxiv.org/ftp/arxiv/papers/2005/2005.10687.pdf>

Wang, T., Lei, Y., Fu, Y., Wynne, J.F., Curran, W .J., Liu, T., Yang, X. (2020). *A review on medical imaging synthesis using deep learning and its clinical applications*. <https://aapm.onlinelibrary.wiley.com/doi/10.1002/acm2.13121>

Tsirikoglou, A ., Eilertsen, G., Unger, J. (2020). *A Survey of Image Synthesis Methods for Visual Machine Learning*. <https://onlinelibrary.wiley.com/doi/full/10.1111/cgf.14047>

Skandarani, Y., Jodoin, P., Lalande, A . (2021). *GANs for Medical Image Synthesis: An Empirical Study*. <https://arxiv.org/pdf/2105.05318.pdf>

Wang, L., Chen, W ., Yang, W ., Bi, F., Yu, F. (2020). *A State-of-the-Art Review on Image Synthesis With Generative Adversarial Networks*. <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9043519>

Shamsolmoali, P., Zareapoor, M., Granger, É., Zhou, H., Wang, R., Celebi, M.E., Yang, J. (2020). *Image Synthesis with Adversarial Networks: a Comprehensive Survey and Case Studies*. <https://arxiv.org/pdf/2012.13736.pdf>

Domagała, M. (2022, 5 września). *The recent rise of diffusion-based models*. deepsense.ai. Pobrano 07.09.2022 z <https://deepsense.ai/the-recent-rise-of-diffusion-based-models/>

Gafni, O., Polyak, A., Ashual, O., Sheynin, S., Parikh, D., Taigman, Y. (2022). *Make-A-Scene: Scene-Based Text-to-Image Generation with Human Priors*. <https://arxiv.org/pdf/2203.13131v1.pdf>

Dane tabelaryczne

Raychaudhuri, S. (2008). *Introduction to Monte Carlo simulation*. <https://www.informs-sim.org/wsc08papers/012.pdf>

Integration, M.C. (2002). *Markov Chain Monte Carlo and Gibbs Sampling*. <https://cis.temple.edu/~latecki/Courses/CIS2033-Spring12/GreatLectures/mc-mc-gibbs-intro.pdf>

Young, J., Graham, P., Penny, R. (2009). *Using Bayesian Networks to Create Synthetic Data*. <https://www.scb.se/contentassets/ca21efb41fee47d293b-bee5bf7be7fb3/using-bayesian-networks-to-create-synthetic-data.pdf>

Gogoshin, G., Branciamore, S., Rodin, A.S. (2021). *Synthetic data generation with probabilistic Bayesian Networks*. <https://www.aimspress.com/article/doi/10.3934/mbe.2021426>

Park, N., Mohammadi, M., Gorde, K., Jajodia, S., Park, H., Kim, Y. (2018). *Data Synthesis based on Generative Adversarial Networks*. <https://arxiv.org/pdf/1806.03384.pdf>

Xu, L., Skoularidou, M., Cuesta-Infante, A., Veeramachaneni, K. (2019). *Modeling Tabular data using Conditional GAN*. <https://arxiv.org/pdf/1907.00503.pdf>

Zhao, Z., Kunar, A., Scheer, H.V., Birke, R., Chen, L.Y. (2021). *CTAB-GAN: Effective Table Data Synthesizing*. <https://arxiv.org/pdf/2102.08369.pdf>

Zhao, Z., Kunar, A., Birke, R., Chen, L.Y. (2022). *CTAB-GAN+: Enhancing Tabular Data Synthesis*. <https://arxiv.org/pdf/2204.00401.pdf>

Yoon, J., Jarrett, D., Schaar, M.V. (2019). *Time-series Generative Adversarial Networks*. <https://www.damtp.cam.ac.uk/user/dkj25/pdf/yoon2019time.pdf>

Sarkar, T. *Random database/dataframe generator*. GitHub. Pobrano 16.09.2022 z <https://github.com/tirthajyoti/pydbgen>

Welcome to Faker's documentation! Pobrano 16.09.2022 z <https://faker.readthedocs.io/en/master/>

Mimesis: Fake Data Generator. Pobrano 16.09.2022 z <https://mimesis.name/en/latest/index.html>

Mesa: Agent-based modeling in Python 3+. Pobrano 16.09.2022 z <https://mesa.readthedocs.io/en/latest/index.html>

Ping, H., Stoyanovich, J., Howe, B. (2017). *DataSynthesizer: Privacy-Preserving Synthetic Datasets*. <https://dl.acm.org/doi/pdf/10.1145/3085504.3091117>

Patki, N., Wedge, R., Veeramachaneni, K. (2016). *The Synthetic Data Vault*. <https://dai.lids.mit.edu/wp-content/uploads/2018/03/SDV.pdf>

Inne (brak w przypisach dolnych):

Bardenet, R. (2018). *Monte Carlo Methods*. https://www.epj-conferences.org/articles/epjconf/pdf/2013/16/epjconf_sos2012_02002.pdf

Taboga, M. (2021). *Markov Chain Monte Carlo (MCMC) methods*. <https://www.statlect.com/fundamentals-of-statistics/Markov-Chain-Monte-Carlo>

Hastings, W.K. (1970). *Monte Carlo Sampling Methods Using Markov Chains and Their Applications*. <http://www2.stat.duke.edu/~scs/Courses/Stat376/Papers/Basic/Hastings1970.pdf>

Dizikes, P. (2010, 17 maja). *Explained: Monte Carlo simulations. Mathematical technique lets scientists make estimates in a probabilistic world*. MIT News. Pobrano 14.09.2022 z <https://news.mit.edu/2010/exp-monte-carlo-0517>

Oliva, J.B., Dubey, K.A., Wilson, A.G., Póczos, B., Schneider, J.G., Xing, E.P. (2016). *Bayesian Nonparametric Kernel-Learning*. <https://arxiv.org/pdf/1506.08776.pdf>

Dong, Q., Elliott, M.R., Raghunathan, T.E. (2014). *A nonparametric method to generate synthetic populations to adjust for complex sampling design features*. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5708580/>

Sun, L., Erath, A. (2015). *A Bayesian network approach for population synthesis*. <https://lijunsun.github.io/files/papers/2015-TRC-BN-Population.pdf>

Kaur, D., Sobiesk, M., Patil, S., Liu, J., Bhagat, P., Gupta, A., Markuzon, N. (2021). *Application of Bayesian networks to generate synthetic health data*. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7973486/>

Hu, M.J., Reiter, J. (2013). *Non-parametric Bayesian Model for Generating Synthetic Household Data*. https://unece.org/fileadmin/DAM/stats/documents/ece/ces/ge.46/2013/Topic_2_Duke_University.pdf

Goncalves, A., Ray, P., Soper, B., Stevens, J., Coyle, L., Sales, A.P. (2020). *Generation and evaluation of synthetic patient data*. <https://bmcmmedresmethodol.biomedcentral.com/articles/10.1186/s12874-020-00977-1>

Mottini, A., Lhéritier, A., Acuna-Agost, R. (2018). *Airline Passenger Name Record Generation using Generative Adversarial Networks*. <https://arxiv.org/pdf/1807.06657.pdf>

Choi, E., Biswal, S., Malin, B.A., Duke, J.D., Stewart, W.F., Sun, J. (2017). *Generating Multi-label Discrete Patient Records using Generative Adversarial Networks*. <https://arxiv.org/pdf/1703.06490.pdf>

Xu, L., Veeramachaneni, K. (2018). *Synthesizing Tabular Data using Generative Adversarial Networks*. <https://arxiv.org/pdf/1811.11264.pdf>

Camino, R.D., Hammerschmidt, C.A., State, R. (2018). *Generating Multi-Categorical Samples with Generative Adversarial Networks*. <https://arxiv.org/pdf/1807.01202.pdf>

Soltana, G., Sabetzadeh, M., Briand, L.C. (2017). *Synthetic data generation for statistical testing*. <https://people.svv.lu/soltana/papers/ASE17.pdf>

Dahmen, J., Cook, D. (2019). *SynSys: A Synthetic Data Generation System for Healthcare Applications*. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6427177/>

Chulyadyo, R., Leray, P. (2018). *Using Probabilistic Relational Models to generate synthetic spatial or non-spatial databases*. <https://hal.archives-ouvertes.fr/hal-01761901/document>

Walonoski, J., Kramer, M., Nichols, J., Quina, A., Moesel, C., Hall, D., Duffett, C., Dube, K., Gallagher, T., McLachlan, S. (2018). *Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record*. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7651916/>