Kanditutkielma

Tietojenkäsittelytieteen kandiohjelma

# NLP-hyökkäysten käyttökohteet

Akira Taguchi

4.3.2022

Matemaattis-luonnontieteellinen tiedekunta

Helsingin yliopisto

**Ohjaaja(t)**

FT Nikolaj Tatti

**Yhteystiedot**

PL 68 (Pietari Kalmin katu 5)
00014 Helsingin yliopisto

Sähkopostiosoite: info@cs.helsinki.fi
URL: http://www.cs.helsinki.fi/

HELSINGIN YLIOPISTO – HELSINGFORS UNIVERSITET – UNIVERSITY OF HELSINKI

| Tiedekunta — Fakultet — Faculty | Koulutusohjelma — Utbildningsprogram — Study programme |
|---|---|
| Matemaattis-luonnontieteellinen tiedekunta | Tietojenkäsittelytieteen kandiohjelma |

Tekijä — Författare — Author

Akira Taguchi

Työn nimi — Arbetets titel — Title

NLP-hyökkäysten käyttökohteet

Ohjaajat — Handledare — Supervisors

FT Nikolaj Tatti

| Työn laji — Arbetets art — Level | Aika — Datum — Month and year | Sivumäärä — Sidoantal — Number of pages |
|---|---|---|
| Kanditutkielma | 4.3.2022 | 7 sivua |

Tiivistelmä — Referat — Abstract

Luonnolisen kielen käsittely on kätevä työkalu käsittelemään ihmisten puhumaa kieltä tietokonemaailmassa.

Luonnollisen kielen käsittely on kuitenkin sellaisenaan haavoittuvainen erillaisille tekstipohjaisille hyökkäyksille.

Tässä tutkielmassa tutustutaan näiden luonnollisen kielen käsittely historiaan, aikaisemmin mainittujen hyökkäysten mahdollistajiin sekä näiden torjuntametodeihin.

**ACM Computing Classification System (CCS)**
Security and privacy
Computing methodologies → Artificial Intelligence → Natural language processing

Avainsanat — Nyckelord — Keywords

nlp, unicode, nlp attack, machine learning, Natural Language Processing, cyber security

Säilytyspaikka — Förvaringsställe — Where deposited

Helsingin yliopiston kirjasto

Muita tietoja — övriga uppgifter — Additional information

# Sisältö

# 1 Johdanto

Ohjelmistojen hyökkäysrajapinta-ala kasvaa jatkuvasti. Osa haavoittuvaisuuksista korjataan heti havainnoinnin jälkeen, osa mitigoidaan ja osan vaikutusalue on näyttäytyy vasta tulevaisuudessa . Luonnollisen kielen käsittely (eng. Natural Language Processing, NLP) on osoittautunut hyväksi hyökkäysrajapinnaksi tätä teknologiaa hyödyntäviä osapuolia vastaan (Boucher et al., 2021). NLP-järjestelmät on tehty tulkitsemaan ihmisen luonnollista kieltä. Tämän kielen konekääntäminen aloitettiin jo vuonna 1949.

Tässä tutkielmassa tarkastellaan NLP-hyökkäysten käyttökohteita. Tähän kuuluu oleellisen historian esittely, hyökkäystaksonomia sekä puolustusmetodit. On tärkeää ymmärtää luonnollisen kielen prosessoinnin tarkoitus, jotta voidaan syventyä hyökkäysiä mahdollistaviin ongelmiin sekä näiden ratkaisemiseen (Yang et al., 2021).

# 2 Tarve luonnollisen kielen käsittelylle

Ehdotukset kielten välisten sanojen välittämisestä koodeilla esitettiin 1700-luvulla Leibnizin ja Descartesin johdolla. Vuonna 1957 Georgetown-IBM-kokeen tekijät väittivät 3-5 vuoden jälkeen konekääntämisen olevan ratkaistu ongelma.

Tarve konekääntämiselle kumpuaa tietokoneen vajeesta ymmärtää ihmisen puhumaa kieltä. Ohjelmoinnissa tämän käännöksen tekee ihminen kutsuessaan esimerkiksi python-tulkilla "print("Hello world")". Käännöksen tapahtuminen tietokonepuolella tuottaa mielekkäämpiä ongelmia. Komento "Tulosta syntymäpäiväni"saattaa koneoppimismallista riippuen tulostaa näytölle merkkijonon "1.1.1970"tai fyysisen kuvan syntymästäsi lähikirjastosi tulostimeen.

NLP-hyökkäyksissä keskiössä on juuri tämän tulkitsemisen vaikeuden hyväksikäyttäminen pahansuopiin tarkoituksiin.

# 3 Hyökkäystaksonomia

Käymme seuraavaksi läpi neljä erilaista huomaamatonta hyökkäysmetodia. Nämä hyökkäykset eivät siis näy visuaalisesti ihmiskäyttäjälle näyttöpäätteellä erilaisina verrattuna viattomaan tekstiin. Tarkemmin keskitymme Unicoden ja muiden enkoodausmenetelmien hyväksikäyttämiseen NLP-malleja vastaan.

## 3.1 Näkymättömät merkit

Näkymättömät merkit vaikuttavat tietokoneen NLP-mallin ymmärtämään kontekstiin. Esimerkki tällaisesta on nollatilavuuden välilyönti -merkki, jonka Unicode merkintä on `U+200B`. Tällä merkillä voimme esimerkiksi vaikuttaa pelichattiin lähetettävän toksissuodatettavan merkkijonoon "olet huono"niin, että merkkijono menisi NLP-mallin läpi chätistä. Merkkijono `olU+200Bet huU+200Bono` saattaisi mennä läpi chatin suodattimesta, mutta vastapuolelle viesti olisi edelleen `olet huono`.

Kontekstin poistamisen lisäksi näkymättömillä merkeillä voidaan myös tuoda ja syrjäyttää konteksteja toisilla. `Mikä pyhäinhäväistyksen rakennus! Miten onnistuit tekemään tämän näin laiskasti?` -tekstin negatiivisuus voidaan syrjäyttää positiivisuudella syöttämällä NLP-mallille sen sijaan teksti `Mikä pyU+200BhäinhävU+200BäistyU+200Bksen`U+200B`rakennus! Miten onnistuit tekemään tämän `U+200B`näU+200Bin laU+200Biskas`U+200B`ti?`.

## 3.2 Homoglyfit

Homoglyyfihyökkäykset NLP-malleja vastaan pohjautuvat pahantahtoisten merkkien virallisten esitysmuotojen näyttävän hyväntahtoisten merkkien virallisilta esityksiltä. Joissain kielissä tekstin merkitys muuttuu täysin yhden merkin vaihtuessa. Esimerkkinä homoglyyfistä on `A` $\rightarrow$ `A`, missä viimeinen kirjain on todellisuudessa kyrillinen kirjain A. Näkymättömien merkkien lailla homoglyyfihyökkäyksen toteutus riippuu ympäristön fontista.

## 3.3 Uudelleenjärjestelyt

Uudelleenjärjestelyhyökkäys pohjautuu näennäisen tekstin uudelleenjärjestämiseen pahantahtoisesti. Pankkitilinumeron `1234567` pystyy esimerkiksi vaihtamaan kaksisuuntaisella-algoritmilla tilinumeroksi `7654321` maksajan huomaamatta mitään.

## 3.4 Poistatukset

Poistatushyökkäyksiä on vaikeampi toteuttaa aikaisempiin metodeihin verrattuna. Tämä johtuu useimpien käyttöjärjestelmien estosta kopioida poistatusta sisältävää tekstiä leikepöydälle.

# 4 Puolustusmetodit

## 4.1 OCR-puolustus

NLP-hyökkäykset voidaan estää alhaisemmalla tasolla overheadilla sekä korkeammalla tasolla edistyneen teknologian turvin. Näytöltäluvun (eng. OCR, On-Screen-Reading) avulla epäselvyydet tekstin aidosta luonteesta voidaan uudelleenrenderöidä tulkitsemalla aineisto uudestaan visuaalisesti. Tämä metodi lisää overheadia huomattavasti riippuen käyttötarkoituksesta, mutta poistaa pahantahtoiset merkit ilman NLP-mallin uudelleenkoulutusta.

## 4.2 Suorituskykykeskeinen puolustus

Keskitymme seuraavaksi näkymättömiin merkkeihin, -homoglyyfeihin, uudelleenjärjestelyihin -ja poistatuksiin perustuvien hyökkyksien puolustamiseen.

Tietyt näkymättömät merkit voidaan poistaa suoraa syötteestä. Mikäli applikaatiossa näitä merkkejä ei voida poistaa, voidaan ne korvata non-<unk> upotuksilla.

Homoglyyfihyökkäysten torjuminen OCR-metodilla on ymmärrettävästi vaikeampaa verrattuina muihin merkkeihin. Paras keino torjua tällaisia hyökkäyksiä olisi mapata osa homoglyyfeistä niiden yleisemmin tunnettuihin vastineisiin. NLP-mallin ylläpitäjä joutuu tekemään tässä siis suurimman jalkatyön.

Uudelleenjärjestelyhyökkäykset voidaan torjua riisumalla bidi-ohjausmerkit syötteestä, varoittamalla käyttäjää bidi-ohjausmerkkien ilmestyessä syötteeseen tai käyttämällä bidi-algoritmia halutun syötteen selvittämiseen.

Poistatukset yleensä havaitaan NLP-mallien ulkopuolella syötteen annon ensivaihessa. NLP-mallin tasolla tähän tarvitsee harvemmin puuttua ja käyttäjälle voidaan pahimmassa tapauksessa lähettää varoitus poistatusmerkkien olemassaolosta syötteessä.

# 5 Yhteenveto

Luonnolisen kielen käsittely on kätevä työkalu käsittelemään ihmisten puhumaa kieltä tietokonemaailmassa.

Luonnollisen kielen käsittely on kuitenkin sellaisenaan haavoittuvainen erillaisille tekstipohjaisille hyökkäyksille.

Tässä tutkielmassa tutustuimme näiden luonnollisen kielen prosessoinnin historiaan, aikaisemmin mainittujen hyökkäysten mahdollistajiin sekä näiden torjuntametodeihin.

# Lähteet

Boucher, N., Shumailov, I., Anderson, R. ja Papernot, N. (2021). *Bad Characters: Imperceptible NLP Attacks*. arXiv: 2106.09898 [cs.CL].

Yang, W., Lin, Y., Li, P., Zhou, J. ja Sun, X. (elokuu 2021). "Rethinking Stealthiness of Backdoor Attack against NLP Models". Teoksessa: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, s. 5543–5557. DOI: 10.18653/v1/2021.acl-long.431. URL: https://aclanthology.org/2021.acl-long.431.