TIETO- JA SÄHKÖTEKNIIKAN TIEDEKUNTA

Mikael Malmi

LUONNOLLISEN KIELEN KÄSITTELY LÄÄKETIETEELLISEN DIAGNOSTIIKAN KESKUSTELUJÄRJESTESTELMISSÄ

Kandidaatintyö
Tietotekniikan tutkinto-ohjelma
Tammikuu 2020

Malmi M. (2020) Luonnollisen kielen käsittely lääketieteellisen diagnostiikan keskustelujärjestestelmissä. Oulun yliopisto, Tietotekniikan tutkinto-ohjelma, 24 s.

TIIVISTELMÄ

Tämä tutkielma käy läpi luonnollisen kielen käsittelyä lääketieteellisen diagnosoinnin keskustelujärjestelmissä. Keskustelujärjestelmistä esitetään yleinen rakenne, jotta voidaan ymmärtää miten järjestelmä tuottaa vastauksen käyttäjän kysymykseen sääntöpohjaisilla tai koneoppivilla tekniikoilla. Keskustelujärjestelmät voidaan jakaa monella tapaa riippuen kielen tuottamisen tekniikoista tai järjestelmän tarkoituksesta. Luonnollisen kielen käsittely keskittyy käyttäjän lausahduksen ymmärtämiseen ja vastausten tuottamiseen. Luonnollisen kielen käsittely ei ole yksittäinen tekniikka, vaan se sisältää monia eri alitehtäviä. Keskustelujärjestelmien ongelmaksi muodostuu se, miten tekoäly saadaan ymmärtämään luonnollista kieltä huolimatta sanojen moniselitteisyydestä sekä muista kieleen liittyvistä tekijöistä. Tämän työn tuloksista saa kattavan ymmärryksen keskustelujärjestelmien yleisestä rakenteesta, sekä luonnollisen kielen käsitteelyyn liittyvistä tekniikoista etenkin lääketieteellisiin kysymyksiin vastaavissa keskustelujärjestelmissä.

Avainsanat: Luonnollisen kielen käsittely, keskustelujärjestelmä, lääketieteellinen kysymys-vastaus-järjestelmä

Malmi M. (2020) Natural Language Processing in Diagnostic Medical Systems. University of Oulu, Degree Programme in Computer Science and Engineering, 24 p.

ABSTRACT

The purpose of this thesis is investigate natural language processing in conversational systems especially on the medical domain. The overall structure of a conversational system is first introduced, to provide an understanding of how the system produces an answer to an user utterance using rule-based or machine learning methods. Conversational systems can be divided depending on the language generation technique or the purpose of the system. Natural language processing focuses on understanding the user utterance, as well as generating an answer. Natural language processing is not a single technique; it consists of many sub tasks. The main issue with conversational systems is how artificial intelligence can understand natural language, despite ambiguity in speech and language processing. The results of this thesis provide a comprehensive understanding of overall structure of conversational systems, as well as techniques related to natural language processing in medical domain question answering systems.

Keywords: Natural language processing, conversational system, medical domain question answering system

SISÄLLYSLUETTELO

ALKULAUSE

Haluaisin kiittää perhettäni opintojeni tukemisesta, sekä TkT Simo Hosiota työni ohjauksesta.

Oulussa 20. tammikuuta 2020

Mikael Malmi

LYHENTEIDEN JA MERKKIEN SELITYKSET

NLP	Natural Languge Processing
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
SVM	Support-Vector Machine
UML	Unified Modeling Language
POS	Part-of-Speech
NLG	Natural Language Generation
CRF	Conditional Random Field
HMM	Hidden Markov Model
RDF	Resource Description Framework
W3C	World Wide Web Consortium
SBD	Sentence Boundary Detection
UMLS	Unified Medical Language System
EAT	Expected Answer Type
IR	Information Retrieval

1. JOHDANTO

Luonnollisen kielen käsittely (natural language processing, NLP) alkoi lingvistiikan ja tekoälyn yhdistävällä tutkimuksella. NLP keskittyy luonnollisen kielen automaattiseen analysointiin ja tuottamiseen. Eräs NLP:n moderni tutkimusalue ovat keskustelujärjestelmät, joita on käytetty moniin eri tarkoituksiin vuosikymmenten aikana. Jotta voidaan ymmärtää miten keskustelujärjestelmät käsittelevät käyttäjän lausahduksen ja tuottavat vastauksen, täytyy ymmärtää keinoja joilla luonnollista kieltä käsitellään. [1, 2]

Ensimmäinen keskustelujärjestelmänä pidetty ohjelmisto ELIZA tehtiin jo 1960-luvulla [3]. 2000-luvun alussa keskustelujärjestelmät (esim. JUPITER, puhelinpohjainen keskustelujärjestelmä säätiedoille [4]) yleistyivät kun teknologia puheentunnistuksen ja luonnollisen kielen käsittelyn kanssa yleistyi. Uusimpana alana keskustelujärjestelmissä ovat välittömän viestinnän applikaatioiden sisäiset keskusteluagentit sekä henkilökohtaiset virtuaaliavustajat (esim. "Siri"Applen tuotteilla ja Amazonin kehittämä "Alexa") [5, 6, 1].

Keskustelujärjestelmien tarkoitus on mahdollistaa ihmisen ja tietokoneen välinen kommunikaatio luonnollisella kielellä. Käyttöliittymät ovat helppouden, nopeuden ja mukavuuden vuoksi siirtymässä luonnolliseen kieleen pohjautuviin käyttöliittymiin. Monien tehtävien, kuten komentojen antaminen ja tiedon hankkiminen on nopeaa luonnollista kieltä käyttämällä erityisesti kädessä pidettävillä laitteilla, joilla kirjoittaminen voi olla epäkäytännöllistä. Henkilökohtaiset avustajat eivät ole sidottuja yhteen tehtävään, vaan niiden avulla voi olla vuorovaikutuksessa monien applikaatioiden kanssa. Edistykset tekoälyssä ja laitteistossa, sekä suurten teknologiayritysten kiinnostus ovat tuoneet henkilökohtaiset virtuaaliavustajat suosioon. [6, 7]

Tietokoneiden, mobiililaitteiden ja internetin yleistyminen ovat luoneet nopean pääsyn suurten informaatiomäärien luo. Amerikkalaisen kyselyn[1] mukaan 35% aikuisista ovat etsineet internetistä tietoa sairaudesta itseään tai tuntemaansa henkilöä varten. Hakukoneiden käyttäminen tiedon hankkimiseen on aikaa kuluttavaa, kun tarvittua tietoa joudutaan haravoimaan loputtomista hakutuloksista. Ajan kulutus on erityisesti ongelma lääketetieteen ammattilaiselle, joka tarvitsee tietoa hyvin nopeasti. Covell, Uman ja Manning:n tutkimuksessa lääkärit eivät työssään voineet käyttää kirjapohjaisia lähteitä tiedon hankkimiseen mm. vanhentuneen tai puuttuvan tiedon, kirjojen puutteellisen indeksoinnin tai tiedon hankkimisen hitauden vuoksi. [8]

Näiden ongelmien ratkaisuun on kehitetty lääketieteellisiin kysymyksiin vastaavia keskustelujärjestelmiä. Keskustelukäyttöliittymät ovat hyvin monimutkaisia monien moduuliensa vuoksi. Äänen tunnistuksen ja vastauksen tuottamisen välissä on monta ongelmaa, suurimpana näistä luonnollisen kielen käsittely. Jotta keskustelujärjestelmä pystyy analysoimaan annetun syötteen automaattisesti ja tuottamaan järkevän vastauksen, täytyy annetut lausahdukset prosessoida. NLP kokonaisuutena voidaan jakaa moniin pienempiin tehväviin, esim. lauserajojen havaitseminen ja nimettyjen entiteettien tunnistus. [1, 9, 2]

Tämän työn tarkoitus on käsitellä keskustelujärjestelmiä, luonnollisen kielen käsittelyä sekä näiden sovelluksia etenkin lääketieteellisiin kysymyksiin vastaavissa

[1] Fox & Duggan (2013) Health Online 2013. http://pewinternet.org/Reports/2013/Health-online.aspx

järjestelmissä. Työ on järjestetty seuraavasti: 2. osa käsittelee keskustelujärjestelmien teknologioita ja käsittelyä. 3. osa käsittelee lääketieteellisen toimialan luonnollisen kielen käsittelyä. 4:ssä keskustellaan edellisistä osista, ja 5:ssä osassa tehdään yhteenveto.

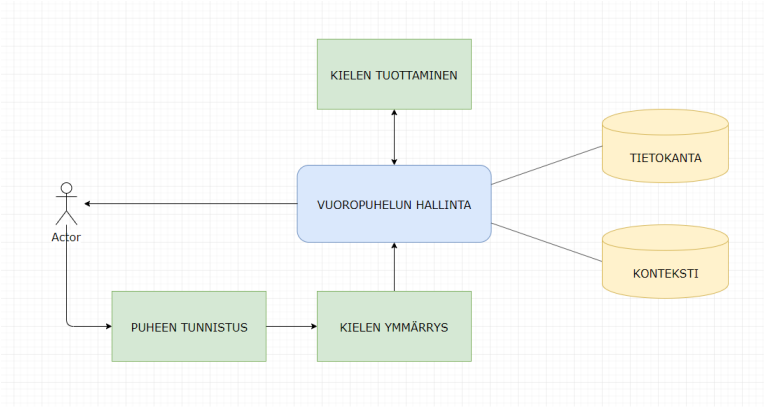
2. KESKUSTELUJÄRJESTELMIEN TEKNOLOGIAT

Keskustelujärjestelmät ovat tekoälyä hyödyntäviä järjestelmiä, jotka ymmärtävät kieltä ja pystyvät keskusteluun ihmisen kanssa teksti- tai äänipohjaisesti. Ymmärtääkseen miten rakennetaan järjestelmä ihmisen ja tietokoneen väliselle kommunikaatiolle, tutkijat ovat tarkastelleet kahden ihmisen välistä keskustelua. Järjestelmää suunniteltaessa täytyy ottaa huomioon, että ihmisten välinen keskustelu on usein täynnä epätäydellisyyksiä, esim. lauseen keskeyttämistä ja uudelleenmuotoilua, keskeytyksiä, lyhyitä kuittauksia kuten "okei"ja "aivan"sekä muita epäsujuvuuksia. Kaikkia lauseita ei voi tulkita oikein tietämättä keskustelun kontekstia. Nämä epätäydellisyydet tulisi huomioida myös keskustelukäyttöliittymissä. [7]

Keskustelujärjestelmät voidaan jakaa kahdella tavalla käyttötarkoituksen mukaan [10, 11]:

1. Keskusteluun suuntautunut
2. Tehtävään suuntautunut

Keskusteluun suuntautuneiden järjestelmien tarkoitus on viihdyttää ja saada keskustelu vaikuttamaan siltä, kuin se tapahtuisi kahden ihmisen välillä. Tehtävään suuntautuneet järjestelmät nimensä mukaan avustavat käyttäjää nimetyn ongelman ratkomisessa, kuten säätietojen kyselyssä. Tehtävään suuntautuneessa järjestelmässä on mahdollista luoda moduuleja sekä valmistella tietoa järjestelmää koskettavaan toimialaan liittyen. Täten avoimen toimialan järjestelmät ovat hankalampia toteuttaa. Perinteiset lähestymistavat ovat sääntöpohjaisia järjestelmiä. Sääntöpohjaiset keskustelujärjestelmät käyttävät käsintehtyjä sääntöjä, jotka ovat riippuvaisia yksittäisten kehittäjien taitotasosta. Kuva 1 esittää yleistettyä keskustelujärjestelmää. [7, 12]



Kuva 1. Yleistys (äänipohjaisen) keskustelujärjestelmän osista.

Kuvaa 1 tulkitessa täytyy huomioida, että kaikissa keskustelukäyttöliittymissä ei ole puheentunnistusmoduulia. Käyttäjän syöte voi tulla suoraan tekstimuodossa, jos järjestelmä on tekstipohjainen äänipohjaisen sijaan.

2.1. Puheentunnistus

Jos järjestelmä on äänipohjainen, käyttäjän lausahdukset prosessoidaan ensin tekstimuotoon käyttäen automaattista puheentunnistusta. Puheentunnistus voidaan toteuttaa syvillä takaisinkytkeytyvillä neuroverkoilla (recurrent neural network, RNN) [13]. Takaisinkytketyissä neuroverkoissa lähtökerrokset vaikuttavat omiin arvoihinsa kytkennän kautta, jolloin äänidatasta voidaan oppia ajallisesti toisistaan riippuvia piirteitä. Tämän lisäksi hyödynnetään pitkän työmuistin verkkoja (Long Short-term Memory, LSTM) jossa olennainen data tunnistetaan ja ohjataan takaisinkytkennällä muistettavaksi. Näin olennainen data auttaa ennustamaan tulevaa dataa.

2.2. Kielen ymmärrys

Luonnollisen kielen käsittelyssä (natural language processing, NLP) pyritään automaattisesti analysoimaan ja esittämään ihmisten kieltä laskennallisin keinoin. NLP vielä nykypäivänä keskittyy tekstin syntaktiseen esitykseen, eli keinoihin, jotka ovat riippuvaisia sanojen yhteisesiintymistiheyksistä. Sanalla tai lauseella voi olla useita eri merkityksiä kontekstista riippuen. Sanojen moniselitteisyys täytyy ottaa huomioon suunnitellessa keskustelujärjestelmää. Tutkijat ovat jakaneet moniselitteisyyden kahteen luokkaan:

1. Sanastollinen moniselitteisyys
2. Syntaktinen moniselitteisyys

Sanastollisessa moniselitteisyydessä merkillä tai merkkijonolla on kontekstin mukaan useita tulkintoja. Esim. Suomen kielessä sana "kuusi"voi tarkoittaa numeroa tai havupuuta. Syntaktisessa moniselitteisyydessä lauseella voi olla kontekstista riippuen useita tulkintoja. Esim. "Näitkö sen pojan kameran kanssa?"voisi tarkoittaa "Näitkö sen pojan, kun sinulla oli kamera?"tai "Näitkö sen pojan jolla oli kamera?". Moniselitteisyyttä käsitellään annotoinnilla (part-of-speech (POS) tagging). [14]

Lingvistiikassa suurta ja jäsenneltyä joukkoa tekstejä kutsutaan korpukseksi. Korpuksia on käytetty luomaan kielimalleja ja opettamaan koneoppivia kieltä ymmärtäviä systeemejä. Koneoppivia luokittelijoita opetetaan suurilla tekstimäärillä. Korpuspohjaiset toteutukset ovat yleinen ratkaisu luonnollisen kielen moduuleissa. Esim. annotoija voidaan opettaa PoS-merkityllä korpuksilla. Korpus voidaan käytännössä valita minkälaisesta tekstimuodosta tahansa, kuten twiiteista tai liiketoimintaan liittyvistä kirjeistä. Korpuksen rajaaminen on tärkeää oikeanlaisten vastausten tuottamiseksi ja sen tulisi sisältää laajalta alueelta sellaisia tekstejä, joita järjestelmän odotetaan tuottavan. [15, 16]

Puheentunnistusmoduuli usein kommunikoi kielenymmärryskomponentin kanssa N-paras liitännän kautta, jotta lausahdus ymmärretään oikein. N-parhaassa liitännässä puheentunnistuskomponentti ehdottaa N parasta lausehypoteesia, joista kielenymmärryskomponentti valitsee suurimmalla todennäköisyydellä oikein tunnistetun lauseen. Usein lausahduksille ei suoriteta täyttä syntaktista analyysia, vaan keskitytään havaitsemaan lausahduksesta avainsanoja. Uusimmissa järjestelmissä on useita eri moduuleja, joilla lausahduksista otetaan talteen tietoa. Tyypillisiä

kielenymmärrysmoduuleja ovat kysymyksen tyypin luokittelu ja keskeisten sanojen tunnistus. Lääketieteellisissä sovelluksissa keskitytään erityisesti tunnistamaan lääketieteellisiä entiteettejä sekä ottamaan talteen sanayhteyksiä. Luonnollisen kielen käsittelyä käsitellään tarkemmin osiossa 2.6. [7, 12, 14, 17]

2.3. Vuoropuhelun hallinta

Vuoropuhelun hallintamoduuli (Dialog Manager, DM) on vastuussa käyttäjän ja järjestelmän välisestä kommunikaatiosta. Tähän liittyy annetun informaation selventäminen sekä mahdollinen lisäinformaation kysyminen. Tämä on tärkeää jotta oikea kysely tietokantaan osataan rakentaa vastauksen saamiseksi. Aktiivisen roolin ottaminen keskustelussa on tärkeää, jotta vuoropuhelu saadaan ohjattua järjestelmän perimmäistä tarkoitusta kohden käyttäjän palvelussa. Onnistunut toteutus keskustelujärjestelmästä ohjaa käyttäjän tehtävän läpi. DM myös huolehtii vuoropuhelun sujuvuudesta olemalla vuorovaikutuksessa muihin moduuleihin. DM:ia voidaan verrata järjestelmän ohjausyksikköön, joka huolehtii kokonaisuuden toimimisesta. [6, 7, 18]

Toteutukset DM:sta usein käyttävät toista kahdesta tavasta, joilla hallitaan vuoropuhelun sujuvuutta: skriptikielellä tai graafeilla. Eräässä toteutuksessa [12] tutkijat opettivat tukivektorikonepohjaisen (Support vector machine, SVM) estimaattorin, joka pyrkii ennustamaan systeemin seuraavaa lausahdusta opetetusta datasta. Nämä arviot vaikuttavat luonnollisen kielen tuottamiseen. Kielentuottajamoduuli ehdottaa lausahduksia systeemin vastaukseksi, ja näistä vaihtoehdoista valitaan yhtenäisin perustuen kriteereihin. Systeemin ulostuloksi valitaan korkeimmalle arvosteltu vastaus useiden lausahdusominaisuuksien mukaan, esim. POS-tagit ja semanttiset kategoriat viimeisimmän ja aikaisempien lausahdusten välillä. [7, 12, 18]

Esimerkkinä lääketieteen puolen toteutuksesta toimii Knowledge-Graph -pohjainen järjestelmä, jossa on käytetty graafipohjaista toteutusta [18]. Hallintamoduuli on jaettu kolmeen tilaan: "Idle", "EvaluatingKGPattern"ja "CollectingKnowledgeForKGPattern". Systeemin tuottama vastaus r_t sekä tila s_t lasketaan aikaisemmasta tilasta sekä käyttäjän viimeisimmän lausahduksen avainsanoista sekä tarkoituksesta (parista (s_{t-1}, m_t).

2.4. Kielen tuottaminen

Luonnollisen kielen tuottamisessa (Natural Language Generation, NLG) pyritään tuottamaan ymmärrettävää tekstiä Englanniksi tai muuksi kieleksi [15]. Suurin ero luonnollisen kielen ymmärtämisen ja tuottamisen välillä on päätösten tekeminen. Päätökset NLG:ssä liittyvät korkean ja matalan tason kysymyksiin tuotettavista lausahduksista. Matalan tason päätökset koskevat yksittäisiä sanoja, ja korkean tason päätökset eri lausevaihtoehtoja kontekstista riippuen. Esimerkiksi 2 vaihtoehdosta:

1. "Minä ostin päärynän. Minä söin sen."
2. "Minä ostin päärynän. Minä soin päärynän."

molemmat ovat valideja vaihtoehtoja. Järjestelmän päätös perustuu kriteereihin, usein luettavuuteen, lukijan kielitaitoihin tai tottumuksiin, tai tuotettavan tekstilajin genren rajoitteisiin. Ensimmäinen vaihtoehto voitaisiin valita sillä perusteella, että se on nopeammin luettavissa. Toinen vaihtoehto voitaisiin valita perustuen helpompaan luettavuuteen, esimerkiksi lukijalle, joka ei puhu suomea äidinkielenään. [19]

Keskustelujärjestemät voidaan jakaa kielen tuottamisen perusteella kahteen luokkaan [12, 14]:

1. Sääntöpohjainen järjestelmä
2. Koneoppiva järjestelmä

Sääntöpohjaisissa järjestelmissä käytetään XML-pohjaisia dialekteja, kuten AIML. AIML-kielellä tuotetaan käsin sääntöjä, joilla tuotetaan järjestelmän vastaus. Järjestelmän toimivuus vastausten osalta on siis täysin riippuvainen kehittäjien taitotasosta. Sääntöjen käsin tuottaminen saattaa kestää jopa kuukausia. Koneoppivat lähestymistavat kiinnittävät painoarvon jokaiselle käyttäjän lausahduksen ominaisuudelle, joista tuotetaan todennäköisyyksiin pohjautuvia päätöksiä. Hallitsevia tekniikoita koneoppivissa järjestelmissä ovat ohjatun oppimisen tekniikat, esim. conditional random fields (CRF), support vector machines (SVM) ja hidden Markov model (HMM). [12, 14]

Monimallisissa systeemeissä [11] vastauksen tuottamiseen käytetään useampaa kuin yhtä moduulia. Moduulilla tarkoitetaan järjestelmän jakamista itsenäisiin, vaihdettaviin ja järjestelmästä erottuviin osiin, joilla suoritetaan haluttu toiminnallisuus. Monen moduulin käyttäminen vastauksen tuottamiseen auttaa järjestelmää tuottamaan olennaisia vastauksia. Jokainen moduuli vastaa oman toimialueen aihealueesta. Kielen tuottamiseen liittyviä moduuleja ovat esim. kysymyksiin vastaava moduuli ja uutta aihealuetta ehdottava moduuli. [12]

2.5. Tyypillinen tietomalli

Keskustelujärjestelmässä käytetty tietokanta riippuu järjestelmän tarkoituksesta. Säätiedusteluihin vastaavan järjestelmän JUPITER:in [4] tietokanta perustuu uusimpiin säätietoihin. Lääketiteteellisissä sovelluksissa tietomallina käytetään Resource Description Framework (RDF). RDF perustuu subjekti-predikaatti-objekti kolmikkoon, jossa Subjektilla S on predikaatti P arvolla objekti O. RDF on World Wide Web Consortium (W3C):n standardoima ja sitä käytetään Web-pohjaisten resurssien määrittelyssä, mutta sitä voidaan myös käyttää normaalin datan esitykseen. [18, 20]

Potilaat sekä lääketieteen ammattilaiset usein eivät hallitse RDF:n rakennetta, saati syntaktisia ja semanttisia vaatimuksia tehdäkseen kyselyn SPARQL-kielellä. SPARQL on W3C:n suosittelema standardi RDF-muotoisen datan kyselyyn. Lääketieteen sovelluksissa ongelmaksi muodostuu käyttäjän antamasta lausahduksesta SPARQL-kyselyiden muodostaminen, jolla päästään RDF:n dataan käsiksi. [8, 17, 18]

2.6. Luonnollisen kielen käsittely

Luonnollisen kielen käsittelyssä (natural language processing, NLP) pyritään automaattisesti analysoimaan ja esittämään ihmisten kieltä laskennallisin keinoin. NLP ei ole yksinäinen tekniikka, vaan se sisältää monia eri tekniikoita, joilla on sama päämäärä. Kaikista suosituimmatkin lähestymistavat näkevät vieläkin tekstin analyysin sana- tai kuvionsovitusongelmana. Älykkäimmätkään rakenteet eivät vielä itse ymmärrä mitä ne tekevät. NLP:tä rajoittaa syntaktisten lähestymistapojen prosessoinnin keskittyminen vain siihen mitä algoritmit pystyvät "näkemään". Algoritmeilta puuttuu ymmärrys semantiikasta, eli terveestä järjestä sekä informaatiosta, joka liittyy todellisen maailman kokonaisuuksiin. Ihmiselle ei tarvitse erikseen kertoa, että tuoli on huonekalutyyppi, tai että kirjaa voi normaalin käyttötarkoituksensa lisäksi käyttää myös esim. paperipainona. Jotta tulevaisuudessa pystytään käsittelemään tekstiä järkevästi ja tarkasti, täytyy laskennallisten mallien kyetä ymmärtämään semantiikkaa ja sentiikkaa. [1, 2, 21]

Tutkijat ovat jakanet syntaksikeskeisen NLP:n kolmeen kategoriaan:

1. Avainsanojen paikannus
2. Sanastollinen affiniteetti
3. Tilastolliset menetelmät

Avainsanojen paikannuksessa teksti jaetaan kategorioihin yksiselitteisten sanojen mukaan. Sanastollisessa affiniteetissa sanoihin lisätään todennäköisyys kuulua tiettyyn kategoriaan. Molemmat tavat ovat riippuvaisia havaittavien sanojen esiintymisestä tekstissä. Tilastolliset menetelmät käsittävät koneoppimisen algoritmeja kuten SVM ja CRF. Tilastollisissa menetelmissä algoritmille syötetään suuria korpuksia opetusdatana. Koneoppivat algoritmit oppivat myös muita ominaisuuksia korpusten avulla, kuten sanojen yhteisesiintymisen taajuuden. [1, 2, 14]

Luonnollisen kielen käsittely voidaan jakaa moneen alitehtävään. Alla oleva osio käsittelee lauserajojen havaitsemisen alitehtävää ja sen erästä tekniikkaa. Matalan tason luonnollisen kielen käsittelyn alitehtävistä tämä on keskeinen koska ylemmän tason alitehtävät, ja tässä työssä esiteltävät muut alitehtävät nojaavat lauserajojen oikeelliseen havainnointiin.

2.7. Lauserajojen havaitseminen

Lauserajojen havaitsemisessa (sentence boundary detection, SBD) tunnistetaan lauserajojen lisäksi myös välimerkkejä. Välimerkkien oikeellinen tunnistus on tärkeää keskustelujärjestelmän muiden osien suorituksen kannalta. Piste voidaan tulkita lause-erottimen lisäksi järjestysnumeroksi, alkukirjaimeksi, lyhenteeksi ja kolmeksi pisteeksi. Virheet lauserajojen havainnoinnissa leviävät myöhempiin NLP:n tehtäviin. Välimerkkien tunnistus on täten hyvin tärkeää, jotta käyttäjän lausahdukset voidaan analysoida oikeellisesti. [22]

SBD:n ongelmat johtuvat lause-erottimista, joita ei käytetä lauserajojen merkitsemiseen, eli esim. lyhenteisiin ja alkukirjaimiin. Eritoten lyhenteet aiheuttavat suuren epävarmuuden lauserajojen havainnoinnissa. Vaikka lyhenteet päättyvätkin

aina pisteeseen, lyhenteitä ei voida havaita vain listaamalla. Muiden kuin lausetta erottavien pisteiden havaitseminen johtaa lauseita päättävien pisteiden päättelyyn, koska jokainen piste, joka ei ole järjestysnumeron yms. perässä, voidaan merkata lauseen päättäväksi pisteeksi. Tämän lisäksi täytyy myös tarkastella, päättääkö jokin lyhenne ja sen perässä oleva piste lauseen.

Kissin ja Strunkin (2006) esittämässä mallissa [22] lauserajoja havaitaan perustuen todennäköisyyksiin. Nollahypoteesilla H_0 (1) oletetaan että pisteen (.) esiintyminen ei ole riippuvainen edeltävästä sanasta (w), kun taas vastahypoteesilla H_A (2) oletetaan että piste esiintyy melkein aina katkaistun sanan jälkeen.

$$H_0 : P(\cdot|w) = P_{\text{MLE}}(\cdot) = \frac{C(\cdot)}{N} \tag{1}$$

$$H_{\text{A}} : P(\cdot|w) = 0{,}99 \tag{2}$$

N on korpuksessa esiintyvien merkkien määrä, ja C(.) on summa kerroista, jolloin merkki ja lauseen päättävä piste esiintyvät peräten korpuksessa. Käyttäen kaavoja (1) ja (2) voidaan laskea logaritminen todennäköisyys log λ käyttäen binomijakaumaa:

$$log\lambda = -2log\frac{P_{\text{binom}}(H_0)}{P_{\text{binom}}(H_{\text{A}})}. \tag{3}$$

Pistettä reunustavia sanoja w_1 ja w_2 tarkastelemalla selvitetään, esiintyykö niiden välillä kollokationaalinen side. Sanojen välillä on kollokaatio jos yhtälöt (4) ja (5) toteutuvat.

$$Log\lambda > raja - arvo \tag{4}$$

$$\frac{C(w_1, w_2)}{C(w_1)} > \frac{C(w_2)}{N} \tag{5}$$

Lyhenteet ovat nimensä mukaisesti lyhyitä mitattuna merkkijonon pituudella. Tämä huomioidaan todennäköisyydessä kaavalla:

$$F_{\text{pituus}} = \frac{1}{e^{\text{pituus(w)}}} \tag{6}$$

jossa pituus(w) on merkkijonon viimeistä pistettä edeltävien merkkien summa, pois lukien välipisteet. Lyhenteissä välipisteet otetaan huomioon kaavalla:

$$F_{\text{pisteet}} = \text{W:N SISÄISET PISTEET} + 1 \tag{7}$$

joka antaa vakioarvon kaikille merkkijonoille joissa ei ole välipisteitä, kun taas välipisteelliset merkkijonot saavat suuremman todennäköisyyden olla lyhenne. Kielissä, joissa lauserakenne päättyy verbiin, logaritminen todennäköisyys erheellisesti olettaa verbin, joka esiintyy ilman lauserajaa ilmoittavaa pistettä poikkeukseksi. Tämä otetaan huomioon kaavalla:

$$F_{\text{rangaistus}} = \frac{1}{pituus(w)^{\text{C(w,-.)}}} \tag{8}$$

Edelliset kaavat yhdistämällä, sanalla w päästään tulokseen:

$$log\lambda(w) \times F_{\text{pituus}} \times F_{\text{pisteet}} \times F_{\text{rangaistus}} >= 0{,}3 \tag{9}$$

$$log\lambda(w) \times F_{\text{pituus}} \times F_{\text{pisteet}} \times F_{\text{rangaistus}} < 0{,}3 \tag{10}$$

jossa (9) tarkoittaa että sana w on lyhenne, ja (10) että sana ei ole lyhenne. Tätä vaihetta kutsutaan tyyppiin perustuvaksi luokitteluksi. Tyyppiluokittelun jälkeen suoritetaan merkkiluokittelu. Jokaiselle pisteeseen päättyvälle sanalle päätetään, täytyykö tyyppiluokittelun päätöstä vielä muuttaa.

Merkkiluokittelussa on kolme vaihetta. Ensimmäisessä vaiheessa tarkistetaan lyhenteen tai kolmen pisteen jälkeinen sana. Jos tämä sana on kirjoitettu isolla alkukirjaimella, tarkistetaan että esiintyykö se pienellä alkukirjaimella tekstissä tai isolla alkukirjaimella keskellä lausetta. Jos sana ei esiinny isolla alkukirjaimella keskellä lausetta ja ainakin kerran pienellä alkukirjaimella, järjestelmä päättelee lauserajan.

Toisessa vaiheessa tarkastellaan, että esiintyykö pistettä reunustavien sanojen välillä kollokationaalinen side. Jos log λ todennäköisyys kahden sanan välillä on suurempaa kuin 7.88, voidaan 99,5% todennäköisyydellä olettaa että sanojen välillä on kollokaatio.

Kolmannessa vaiheessa tarkastellaan sanoja, jotka usein aloittavat lauseen. Jos log λ todennäköisyys on suurempaa kuin raja-arvo 30, lisätään sana todennäköisten lauseidenaloittajien listaan. Kolmas vaihe tasapainottaa toista vaihetta. Jos sana löytyy todennäköisistä lauseenaloittajista, ei tälle sanalle voida päätellä kollokaatiota.

3. LUONNOLLISEN KIELEN KÄSITTELY LÄÄKETIETEEN PIIRISSÄ

Luonnollisen kielen käsittelyn ongelmat liittyvät lääketieteellisiin kysymyksiin vastaavissa keskustelujärjestelmissä eritoten entiteettien tunnistukseen ja tiedonhakuun (information retrieval, IR). Perinteinen järjestelmä perustuu kysymyksen analysointiin, avainsanojen tunnistukseen, vastausten hakuun ja vastauksen tuottamiseen. Luonnollisen kielen käsittelyn perusongelmien lisäksi haasteeksi muodostuvat spesifin toimialan ongelmat. Seuraavissa alakappaleissa käsitellään näitä ongelmia. [23]

3.1. Lääketieteellisten entiteettien tunnistus

Nimettyjen entiteettien tunnistusta on avoimen toimialan toteutuksissa käytetty mm. luokittelemaan entiteettejä tekstistä [9]. Lääketieteellisellä toimialalla NLP:tä käytetään yhdistämään sanoja konsepteihin. Lääketieteellisten entiteettien tunnistuksessa (medical entity recognition, MER) havaitaan ja rajataan käyttäjän lausahduksesta tieto joka liityy lääketieteellisiin entiteetteihin. Havaitut entiteetit luokitellaan ennalta määrättyihin joukkoihin. Tutkijat ovat käyttäneet esim. 7 eri luokkaa: ongelma, hoito, merkki tai oire, lääke, ruoka, potilas ja testi. [8, 17]

Tässä työssä käsitellään kahta yleisesti käytössä olevaa tekniikkaa, joita käytetään luokkien tunnistukseen. Ensimmäisenä MetaMap Plus (MM+) jota käytetään kartoittamaan käyttäjän lausahduksesta substantiivilauseita (noun phrase, NP) ja yhdistämään Unified Medical Language System (UMLS) -konsepteihin vastaavan pisteytyksen mukaisesti. Samasta konseptista voidaan puhua monella eri nimellä, joka osoittautuu haasteelliseksi ongelmaksi kielen ymmärrykselle. UMLS-konsepteja käytetään yhdistämään saman konseptin eri terminologioita. [8, 17, 21]

Alkuperäistä MetaMap-tekniikkaa käytetään kartoittamaan biolääketieteellisiä tekstejä UMLS-konsepteihin [24]. Parannetussa versiossa suorituskykyä on parannettu seuraavin keinoin: NP:lle annotoidaan tunnisteet TreeTagger-chunker-pohjaisesti, NP:tä suodatetaan lopetus-sana -listalla, vaihtoehtoisia termejä etsitään spesialisoidusta listasta, NP:t kommentoidaan MetaMapin avulla UMLS-konsepteihin, sekä suodatetaan vastauksia listalla yleisistä virheistä. [8, 17, 21]

Toisena tekniikkana käytetään BIO-CRF-H:ta, jolla tunnistetaan entiteettien rajat ja kategoriat. BIO-CRF-H koostuu CRF-luokittelijasta (Conditional Random Field, CRF), jossa seuraava tila riippuu nykyisestä tilasta, sekä B-I-O formaatista. Formaatin mukaan käyttäjän lausahdus luokitellaan sanojen mukaan i) B: Beginning, entiteetin alku, ii) I: inside, entiteetin jatko, iii) O: outside, entiteetin ulkopuoliset sanat. [1, 8, 17]

CRF-luokittelijan opetukseen ja testaukseen tarvitaan korpus. Tutkijat ovat käyttäneet korpuksena muun muassa annotoitua i2b2 2010 korpusta. Seuraavia ominaisuuksia käytetään luokittelijan opetukseen ja testaukseen:

1. Morphosyntaktiset ominaisuudet: sanoille kommentoidut POS-tagit
2. Sanaominaisuudet: sana itsessään, sekä 2 edeltävää sanaa ja 3 sanaa sen jälkeen
3. Semanttiset ominaisuudet: sanan semanttinen kategoria
4. Ortografiset ominaisuudet: esim. sanaan sisältyvä tavuviiva tai muu ominaisuus

3.2. Sanayhteyksien määritys

Lääketieteellisten entiteettien tunnistuksen myötä voidaan tunnistaa näiden sanojen välisiä suhteita. Kysymyksiin vastaavien järjestelmien täytyy käsitellä oikeellisesti käyttäjän kysymys. Sanayhteyksien määrittämistä on käytetty mm. helpottamaan kysymyksiin vastaamista sekä informaation talteen ottamista. [25]

Lääketieteellisissä sovelluksissa [8, 17] on keskitytty havaitsemaan 7 eri semanttista kategoriaa entiteettien välillä:

1. Treats (hoitaa): hoito kohentaa tai parantaa lääketieteellistä ongelmaa
2. Prevents (ehkäisee): hoito ehkäisee lääketieteellisen ongelman
3. Causes (aiheuttaa): hoito aiheuttaa lääketieteellisen ongelman
4. Complicates (hankaloittaa): hoito pahentaa lääketieteellistä ongelmaa
5. Diagnoses (diagnosoi): testi havaitsee, arvioi tai diagnosoi lääketieteellisen ongelman
6. DhD (drug has dose): lääkeannoksen koon havaitseminen
7. P_hSS (problem has signs or symptoms): lääketieteellisellä ongelmalla on merkki tai oire

Näitä sanayhteyksiä havaitaan kahden eri metodin yhdistelmällä. Ensimmäisenä kaavapohjainen lähestymistapa, jossa kaava on vakinainen lausahdus joka sisältää lääketieteellisiä entiteettejä tietyssä kohtaa lausahdusta. Kaavat muodostetaan käsin toimialueeseen liittyvistä teksteistä, ja ne järjestetään hierarkkisesti perustuen tarkkuuteen. Kaikista tarkimmat kaavat ovat ’lehti’-kaavoja, jotka juontuvat geneerisimmistä kaavoista. Kaavoille annetaan painoarvo niiden spesifiyden mukaan. Painoarvoa käytetään kaikista spesifeimmän sanayhteyden valitsemiseen laskettaessa luottamusindeksiä I:

$$I(R) = \frac{W(P)}{e^{\mathrm{N\ N\ (S,E1,E2)}}} \tag{11}$$

jossa W(P) on kaavan P painoarvo, R havaittu sanayhteys ja neperin luvun potenssi on lauseen S sisältämät NP:t lääketieteellisten entiteettien E1 ja E2 välissä. [25]

Myös Minutolo, Esposito & De Pietroa (2017) [18] toteutus käyttää kaavapohjaista lähestymistapaa. Järjestelmässä käyttäjän lausahduksesta havaittuja avainsanoja ja niiden synonyymejä verrataan aikomusvaraston esimerkkeihin ja konsepteihin.

Toisena keinona on käytetty ohjattua oppimista SVM:lla. SVM on tilastollinen keino, jonka tavoitteena on opettaa luokittelija, jolla automaattisesti suoritetaan sanojen luokittelu. Automaattisen luokittelijan etuna verrattuna kaavapohjaiseen on se, että vältytään tekemästä kaavoja manuaalisesti. Luokittelijaa opetettavasta datasta (korpus) jokainen selvästi erottuva sanavarsi vastaa yhtä ominaisuutta vektorissa. Ominaisuuden arvona käytetään sanavarren esiintymisen lukumäärää opetusdatassa. Sanavarsi lasketaan ominaisuudeksi vain, jos se esiintyy opetusdatassa vähintään 3 kertaa, ja se ei ole ’pysäytys-sana’ (esim. ’ja’). Vektori voi sisältää jopa yli 10000 ulottuvuutta. Luokittelija käyttää samoja morphosyntaktisia, semanttisia ja sanallisia ominaisuuksia kuten osiossa 4.1, sekä verbejä lääketieteellisten entiteettien E1 ja E2 välissä. [8, 17, 25, 26]

Hybridimetodissa [8, 25] yhdistetään kaavapohjainen sekä tilastollinen lähestymistapa. Kummallekin aiemmalle lähestymistavalle annetaan painoarvo.

SVM-pohjaisessa metodissa painoarvo perustuu opetusdatasta löytyvien esimerkkien lukumäärään sanayhteydellä R.

3.3. Kysymysten analysointi ja SPARQL-kyselyiden muodostus

Kun käyttäjän lausahduksesta on tunnistettu lääketieteelliset entiteetit sekä näiden väliset sanayhteydet, tiedetään kysymyksen painopiste. Vastaus käyttäjän kysymykseen saadaan muodostamalla SPARQL-muotoinen kysely RDF-tietolähteelle.

Jotta kysely voidaan muodostaa, täytyy määrittää käyttäjän kysymyksen tyyppi. Kysymystyyppejä ovat esimerkiksi WH-kysymys (what, how jne.) ja kyllä/ei kysymys. WH-kysymyksille määritetään myös odotettu vastauksen tyyppi (expected answer type, EAT). Jos käyttäjän kysymyksessä on monta odotettua vastausta, jokaiselle näistä määritetään oma vastaustyyppi. Esim. "Kuinka diagnosoidaan ja hoidetaan pääkipua?", jossa odotetut vastaustyypit ovat Diagnoses ja Treats. Kysymystä myös yksinkertaistetaan korvaamalla interrogatiivipronominit "ANSWER"avainsanalla. Esim. "Miten parhaiten hoidetaan pääkipua"yksinkertaistetaan muotoon "ANSWER hoidetaan pääkipua". [8, 17]

SPARQL:llä voidaan tehdä kyselyjä neljällä eri muotilla, joista tutkijat ovat käyttäneet kahta: ASK kyllä/ei-kysymyksille, sekä SELECT WH-kysymyksille. SPARQL-kysely muodostuu otsikosta (header) ja rungosta (body). Otsikko kertoo kyselyn muotin ja rungossa on kyselyn tieto. [8, 25, 27]

3.4. Vastausten haku

Vastauksia etsitään perustuen MESA-ontologiaan, jolloin voidaan ottaa huomioon lääketieteellisten entiteettien synonyymit ja morfologiset vaihtelut. Käyttäjän lausahduksen semanttisessa kommentoinnissa voidaan tehdä virheitä tai olla huomaamatta tärkeää informaatiota. Ben Abacha ja Zweigenbaum (2015) [8] käyttävät järjestelmässään kyselyiden asteittaista rentouttamista. Alkuperäisestä kyselystä muodostetaan kolmella eri tasolla kyselyitä, joissa on vähemmän rajoitteita.

Ensimmäisellä tasolla nimettyjen entiteettien arvot jätetään pois. Jos esim. alkuperäisessä kysymyksessä etsitään tietoa eritoten "terveillä aikuisilla", jätetään aikuisen arvo "terve"pois. Toisella tasolla havaitut lääketieteelliset entiteetit jätetään yksitellen pois, mutta niihin liittyvä kysymyksen painopiste ja odotettu vastaustyyppi jätetään kyselyyn. Kolmannella tasolla pääsuhde, joka määrittää vastauksen objektiksi tai subjektiksi, jätetään pois. [8]

Vastaukset arvostellaan perustuen kahteen kriteeriin: vastaus arvostetaan kuten siihen liittyvä kysely, kaikista spesifein sijoitetaan korkeimmaksi. Myös vastauksen perustelujen lukumäärä otetaan huomioon. Jos vastauksilla on sama CUI (concept unique identifier) eli uniikki konseptitunniste, katsotaan vastausten olevan identtisiä. Jokaiselle kysymykselle esitetään vastaus, perustelu, 2 vastausta edeltävää ja jälkeistä lausetta. [8]

4. KESKUSTELU

Tässä työssä käsitellyt toteutukset ovat olleet kysymyksiin vastaavia järjestelmiä tai chattibotteja. Toisin taas kuin uusimmissa toteutuksissa keskustelusuuntautuneista järjestelmistä, keskustelun ymmärrys tehtävän ulkopuolella on hyvin vähäistä. Käyttäjästä ei pyritä keräämään tietoa, eikä uusia käsitteitä opita ja yhdistetä jo tunnettuihin konsepteihin. Toteutukset ovat melko jäykkiä käyttäjän keskustelumahdollisuuksien kannalta.

Kuten luonnollisen kielen käsittelyssä muillakin toimialoilla, lääketieteellisissä sovelluksissa NLP:n keinot vielä rajoittuvat yksittäisten lauseiden käsittelyyn. Kontekstista riippuen lause "pieni jono"voidaan tulkita negatiiviseksi tai positiiviseksi. Maalaisjärki sekä semantiikka liittyen oikean maailman käytäntöihin ja asioihin jäävät uupumaan järjestelmistä. Seuraava suuri askel keskustelujärjestelmien ja luonnollisen kielen käsittelyn saralla onkin isomman kuvan ymmärtäminen kuin yksittäisten lauseiden.

Johdantokappaleessa esitetyt kysymykset saivat työn aikana vastaukset. Kaiken kaikkiaan tämä työ tuo esille lääketieteellisissä keskustelujärjestelmissä useinten käytetyt teknologiat ja keinot luonnollisen kielen käsittelyyn.

5. YHTEENVETO

Tässä työssä käytiin läpi ongelmia liittyen luonnollisen kielen käsittelyyn lääketieteellisen toimialan sovelluksissa. Keskustelujärjestelmien tai yleistyminen ovat tehneet NLP:n ongelmista entistä tutkitumman aiheen.

Keskustelujärjestelmien yleinen arkkitehtuuri käytiin läpi, ja todettiin että järjestelmiä voidaan jakaa kahteen luokkaan käyttötarkoituksen sekä kielen tuottamisen perusteella. Järjestelmiä toteutetaan monella eri tapaa, mutta niillä on sama perusperiaate. Käyttäjän lausahdus käsitellään ja mahdollisia vastauksia tuotetaan sääntöpohjaisesti tai koneoppivasti. Vastausvaihtoehdoista valitaan sopivin perustuen kontekstiin ja valittuihin kriteereihin.

Luonnollisen kielen käsittelyä käsiteltiin yleisesti sekä valitun toimialan kontekstissa. NLP:n ongelmaksi muodostuu se, miten tekoäly saadaan ymmärtämään luonnollista kieltä huolimatta sanojen moniselitteisyydestä sekä muista kieleen liittyvistä tekijöistä. Käsiteltyihin ongelmiin kuuluivat lääketieteellisten entiteettien tunnistus ja sanayhteyksien määritys. Kumpikin näistä alitehtävistä on ennakkovaatimus järjestelmän lopputavoitteen saavuttamiseksi. Jotta käyttäjän kysymykseen saadaan vastaus, täytyy kysymys analysoida, ja siitä täytyy muodostaa kysely tietolähteelle.

6. VIITTEET

[1] Nadkarni P., Ohno-Machado L. & Chapman W. (2011) Natural language processing: An introduction. Journal of the American Medical Informatics Association 18, ss. 544–551. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-80053254020&doi=10.1136%2famiajnl-2011-000464&partnerID=40&md5=2d97960a46bc446ec1c53251b6bed06e, cited By 248.

[2] Cambria E. & White B. (2014) Jumping nlp curves: A review of natural language processing research. IEEE Computational Intelligence Magazine 9, ss. 48–57. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84899050913&doi=10.1109%2fMCI.2014.2307227&partnerID=40&md5=3d8fc35df7674842e37bd904d2ca1741, cited By 272.

[3] Weizenbaum J. (1966) Eliza-a computer program for the study of natural language communication between man and machine. Communications of the ACM 9, ss. 36–45. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84911514327&doi=10.1145%2f365153.365168&partnerID=40&md5=c2b08fd4edbd80e01c32728b23779191, cited By 1343.

[4] Zue V., Seneff S., Glass J., Polifroni J., Pao C., Hazen T. & Hetherington L. (2000) Jupiter: A telephone-based conversational interface for weather information. IEEE Transactions on Speech and Audio Processing 8, ss. 85–95. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-0033878021&doi=10.1109%2f89.817460&partnerID=40&md5=4f3b07608aa592526b92346bf0911512, cited By 289.

[5] Masche J. & Le N.T. (2018) A review of technologies for conversational systems. Advances in Intelligent Systems and Computing 629, ss. 212–225. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85025125649&doi=10.1007%2f978-3-319-61911-8_19&partnerID=40&md5=182f15da5f62d07487bd4eb5c1514b66, cited By 3.

[6] Klopfenstein L., Delpriori S., Malatini S. & Bogliolo A. (2017) The rise of bots: A survey of conversational interfaces, patterns, and paradigms. ss. 555–565. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85023168479&doi=10.1145%2f3064663.3064672&partnerID=40&md5=ae19f197b780e24ef14b2882e34108a3, cited By 38.

[7] Zue V.W. & Glass J.R. (2000) Conversational interfaces: advances and challenges. Proceedings of the IEEE 88, ss. 1166–1180.

[8] Ben Abacha A. & Zweigenbaum P. (2015) Means: A medical question-answering system combining nlp techniques and semantic web technologies.

Information Processing and Management 51, ss. 570–594. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84930938943&doi=10.1016%2fj.ipm.2015.04.006&partnerID=40&md5=59d6afc4a1be0ca617d3fd6a4b3bc905, cited By 62.

[9] Ritter A., Sam C., Mausam & Etzioni O. (2011) Named entity recognition in tweets: An experimental study. ss. 1524–1534. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-80053238545&partnerID=40&md5=0f86100df039a049c66c30cb67d42182, cited By 532.

[10] Banchs R.E. & Li H. (2012) Iris: A chat-oriented dialogue system based on the vector space model. Teoksessa: Proceedings of the ACL 2012 System Demonstrations, ACL '12, Association for Computational Linguistics, Stroudsburg, PA, USA, ss. 37–42. URL: http://dl.acm.org/citation.cfm?id=2390470.2390477.

[11] Ali A. & Gonzalez A. (2016) Toward designing a realistic conversational system: A survey. ss. 2–7. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85003899086&partnerID=40&md5=c5349dfb8d256ca568abb20db8fbb8e7, cited By 1.

[12] Higashinaka R., Imamura K., Meguro T., Miyazaki C., Kobayashi N., Sugiyama H., Hirano T., Makino T. & Matsuo Y. (2014) Towards an open-domain conversational system fully based on natural language processing. ss. 928–939. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84959910663&partnerID=40&md5=d908e14bb03b01e0d373dd807b554cae, cited By 64.

[13] Graves A., Mohamed A.R. & Hinton G. (2013) Speech recognition with deep recurrent neural networks. ss. 6645–6649. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84890543083&doi=10.1109%2fICASSP.2013.6638947&partnerID=40&md5=2423a6be7bdd3d0bf2c2701fa147f6b7, cited By 2310.

[14] Khan W., Daud A., Nasir J. & Amjad T. (2016) A survey on the state-of-the-art machine learning models in the context of nlp. Kuwait Journal of Science 43, ss. 95–113. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84996844019&partnerID=40&md5=f6e92851fe4b60e97cc3403b91d795c7, cited By 14.

[15] Reiter E. & Dale R. (1997) Building applied natural language generation systems. Natural Language Engineering 3, ss. 57–87. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84956732014&doi=10.1017%2fS1351324997001502&partnerID=40&md5=8900bef345302ac64cd8a07aa5400bdd, cited By 126.

[16] Abu Shawar B. & Atwell E. (2005) Using corpora in machine-learning chatbot systems. International Journal of Corpus Linguistics 10, ss. 489–516. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-34248695074&partnerID=40&md5=13c550c68ac06ebaf1cf62febb74d8ea, cited By 35.

[17] Abacha A. & Zweigenbaum P. (2012) Medical question answering: Translating medical questions into sparql queries. ss. 41–49. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84857758460&doi=10.1145%2f2110363.2110372&partnerID=40&md5=448b796c0c246b4d48ac506c2714f59b, cited By 29.

[18] Minutolo A., Esposito M. & De Pietroa G. (2017) A conversational chatbot based on kowledge-graphs for factoid medical questions. Frontiers in Artificial Intelligence and Applications 297, ss. 139–152. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85029222099&doi=10.3233%2f978-1-61499-800-6-139&partnerID=40&md5=c70de4e759b14c7d9e9b349fc3213689, cited By 0.

[19] Reiter E. (2010) Natural Language Generation. Wiley-Blackwell, 574-598 s. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84887130566&doi=10.1002%2f9781444324044.ch20&partnerID=40&md5=89a6346cb350e228dd60b0566d84ae2d, cited By 19.

[20] Broekstra J., Kampman A. & Van Harmelen F. (2002) Sesame: A generic architecture for storing and querying rdf and rdf schema. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 2342 LNCS, ss. 54–68. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84866262574&partnerID=40&md5=a645888b8a72a4569dbb093fadbcefeb, cited By 687.

[21] Chary M., Parikh S., Manini A., Boyer E. & Radeos M. (2019) A review of natural language processing in medical education. Western Journal of Emergency Medicine 20, ss. 78–86. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85059906776&doi=10.5811%2fwestjem.2018.11.39725&partnerID=40&md5=5a1859629cf68462961454139c1ba805, cited By 0.

[22] Kiss T. & Strunk J. (2006) Unsupervised multilingual sentence boundary detection. Computational Linguistics 32, ss. 485–525. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-33845487544&doi=10.1162%2fcoli.2006.32.4.485&partnerID=40&md5=ddbc17a2458c67786418173ac7ce2794, cited By 125.

[23] Jacquemart P. & Zweigenbaum P. (2003) Towards a medical question-answering system: A feasibility study. Studies in Health Technology and Informatics 95, ss. 463–468. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84887909261&doi=10.3233%2f978-1-60750-939-4-463&partnerID=40&md5=9b5de21b91fb3b1a2a7dc951eea8b87f, cited By 44.

[24] Aronson A. (2001) Effective mapping of biomedical text to the umls metathesaurus: the metamap program. Proceedings / AMIA ... Annual Symposium. AMIA Symposium , ss. 17–21URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-0035752429&partnerID=40&md5=a89973b5045d5703927165f9efa26d88, cited By 1032.

[25] Ben Abacha A. & Zweigenbaum P. (2011) A hybrid approach for the extraction of semantic relations from medline abstracts. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 6609 LNCS, ss. 139–150. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-79952260968&doi=10.1007%2f978-3-642-19437-5_11&partnerID=40&md5=d9b96c4a7ce04ff7afa7756ae310ceec, cited By 27.

[26] Joachims T. (1998) Text categorization with support vector machines: Learning with many relevant features. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 1398, ss. 137–142. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84957069814&doi=10.1007%2fs13928716&partnerID=40&md5=4839c690c7664dd0ed635f8bf15450fc, cited By 3469.

[27] Pérez J., Arenas M. & Gutierrez C. (2009) Semantics and complexity of sparql. ACM Transactions on Database Systems 34. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-70349088138&doi=10.1145%2f1567274.1567278&partnerID=40&md5=1ca39094b7dfb610ebd3098cd278c87d, cited By 470.