



**FACULTY
OF MATHEMATICS
AND PHYSICS**
Charles University

BACHELOR THESIS

František Trebuňa

Generating text from structured data

Institute of Formal and Applied Linguistics (ÚFAL)

Supervisor of the bachelor thesis: Mgr. Rudolf Rosa, Ph.D.

Study programme: Computer Science (B1801)

Study branch: General Computer Science Bc. R9
(NIOI9B)

Prague 2021

I declare that I carried out this bachelor thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In date

Author's signature

Dedication.

Title: Generating text from structured data

Author: František Trebuňa

Institute: Institute of Formal and Applied Linguistics (ÚFAL)

Supervisor: Mgr. Rudolf Rosa, Ph.D., Institute of Formal and Applied Linguistics (ÚFAL)

Abstract: Abstract.

Keywords: text generation structured data natural language processing neural networks

Contents

Introduction	3
1 Problem statement	4
1.1 Data to text generation	4
1.2 Fantasy sports	4
1.3 My goal	4
2 Data	5
2.1 General description	5
2.2 WikiBIO dataset	5
2.2.1 Structured Data	5
2.2.2 Summaries	5
2.3 Rotowire dataset	6
2.3.1 Structured Data	6
2.3.2 Summaries	8
2.3.3 Relation of summaries and tables	8
3 Preprocessing and Statistics of the Datasets	10
3.1 Transforming Tables to Records	10
3.1.1 Notation	10
3.2 WikiBIO	10
3.2.1 Transformation of Infoboxes	10
3.2.2 Dataset statistics	11
3.2.3 Preprocessing	11
3.3 RotoWire	11
3.3.1 Transformations of Input Tables	11
3.3.2 Transformations of Summaries	12
3.3.3 Dataset Statistics	13
3.3.4 Byte Pair Encoding	13
4 Neural Network Architectures	14
4.1 RNN	14
4.2 LSTM	14
4.3 Attention	14
4.4 Copy mechanism	14
4.5 Truncated Backpropagation Through Time	14
4.6 Beam search	14
4.7 Transformers	14
5 Models	15
5.1 Tensorflow	15
5.2 Sequence to Sequence architecture	15
5.3 Encoder	15
5.4 Decoder	15
5.5 Base Model	15
5.6 Joint Copy model	15

6 Experiments	16
6.1 BLEU	16
6.2 Manual evaluation	16
6.3 Other evaluation approaches	16
6.4 Results of the baseline model	16
6.5 Dropout	16
6.6 Scheduled Sampling	16
6.7 Copy methods	16
Conclusion	17
Bibliography	18
List of Tables	19

Introduction

1. Problem statement

In september 2020 I read a blog by [Karpathy, 2015]. He created a neural network consisting of only one LSTM cell and trained it to predict a character based on all the previous ones. The network was trained on a corpus of all the plays by Shakespeare. During inference the last predicted character was fed as the input to the network and this way it could create a really good looking Shakespeare-like text. Then I began to explore the possibilities of generating a text conditioned on some input parameters. How to construct a network that could be told to generate a sad, happy, or sarcastic sounding text?

1.1 Data to text generation

Known datasets (WIKIBIO, WeatherGov, RoboCup) -> short description, the generated summaries are one-two sentences long. Rotowire -> really long summaries although only a fraction of the number of unique tokens from the WIKIBIO dataset. Only short description of the dataset, the statistics and observations are in the second chapter.

1.2 Fantasy sports

What it is, where it originates, the relation to basic optimisation problems, why NLGenerated summaries could be added value.

1.3 My goal

Fluent text which captures the important statistics from the table

2. Data

One needs a lot of data if he wants to train his neural network. E.g. [Sennrich et al., 2016] trains the neural machine translation system on 4.2 million English-German sequence pairs. The Data-to-Text generation task má vyššie nároky na kvalitu datasetu. Potrebujeme, aby boli vstupné dáta štandardizované a aby výstupné texty zodpovedali vstupným dátam. Existuje viacero datasetov, ktoré spĺňajú túto podmienku. V tejto kapitole predstavím datasety WikiBIO a Rotowire.

2.1 General description

Obidva tieto datasety používajú notáciu, ktorá bola predstavená v článku od [Liang et al., 2009], preto ju najprv aj tu zadefinujeme.

Ako vstup používame postupnosť záznamov (recordov) $\mathbf{s} = \{r_i\}_{i=1}^J$. Každý record r má svoj typ $r.t \in \mathcal{T}$. Množina typov \mathcal{T} je dopredu definovaná. Ďalej má typ, množinu hodnôt $r.v = \{r.v_1, \dots, r.v_m\}$. Napríklad v datasete WeatherGOV: $r.t == windSpeed$, $r.v = \{time, min, mean, max, mode\}$. Na základe týchto záznamov následne predpovedáme výstupný text $\mathbf{w} = \{w_i\}_{i=1}^{|\mathbf{w}|}$

2.2 WikiBIO dataset

Na to, aby som vedel rozoznať, čím je generovanie popisu športového zápasu náročné, rozhodol som sa pracovať ešte s jednou úlohou a to s generovaním krátkych životopisov na základe infoboxu z wikipédie. Touto úlohou sa zaoberali tvorcovia datasetu WikiBIO [Lebret et al., 2016].

2.2.1 Structured Data

Vstupné štruktúrované dáta sú vo forme infoboxu z Wikipédie. Infobox je tabuľka faktov opisujúca život osoby. Charakteristickou vlastnosťou infoboxov je ich rôznorodosť. Vyplýva to z toho, že osoby s inou kariérou majú výrazne iné charakteristiky. U niekoho hovoríme o titule, u iného o povolání, u niekoho spomenieme významné práce, u niekoho tímy, v ktorých hraje. Zatiaľ čo zrejme pri každej osobe sa vyskytne napríklad položka "dátum narodenia", pri žijúcich sa zrejme nevyskytne "dátum úmrtia". Dá sa teda povedať, že štruktúrované dáta z tohto datasetu sú veľmi ťažké, pretože generačný systém nemôže predpokladať štruktúru, ale musí sa naučiť rozoznať ju. Príkladom môže byť infobox pre T.G. Masaryka 2.1.

2.2.2 Summaries

Ako sumár infoboxu slúži prvá veta z článku na wikipedii, ku ktorému prislúcha daná tabuľka. Z pozorovaní vstupných dát vyplýva, že prvá veta väčšinou obsahuje meno, dátum narodenia, prípadne dátum úmrtia a najdôležitejšie činnosti

a úspechy osoby, ktorej sa infobox týka. Už z pozorovaní tvorcov datasetu [Lebret et al., 2016] vyplýva, že okolo tretiny tokenov nachádzajúcich sa v prvej vete pochádza z tabuľky. Generačný systém pracujúci s týmto datasetom teda bude musieť správne určiť, kedy sa spoľahnúť na jazykové zručnosti a kedy kopírovať. Ako príklad možno použiť sumár prislúchajúci k infoboxu 2.1.

2.3 Rotowire dataset

Pre samotnú úlohu generovania textového popisu zápasu zo štruktúrovaných dát som si vybral dataset RotoWire [Wiseman et al., 2017]. Štruktúrované dáta sú vo forme tabuľky. Tabuľka obsahuje jednak hodnoty popisujúce celkové tímové štatistiky, jednak hodnoty, popisujúce štatistiky hráčov. Ako výstupné dáta slúžia sumáre jednotlivých zápasov z portálu venujúceho sa real-time fantasy sports news <https://www.rotowire.com/basketball/> (see below).

2.3.1 Structured Data

V tejto podsekcii si rozoberieme, aké dáta sa v tabuľke nachádzajú, a čo z nich sa dá využiť pri vytváraní sumáru. Pre prehľadnosť ich rozdelím na tri časti, kontextové údaje, tímové údaje a štatistiky a hráčske údaje a štatistiky. Keďže

tomáš garrigue masaryk , sometimes called thomas masaryk in english (7 march 1850 - 14 september 1937) , was a czechoslovak politician , sociologist and philosopher , who as an eager advocate of czechoslovak independence during world war i became the founder and first president of czechoslovakia .

Tomáš Garrigue Masaryk


1st President of Czechoslovakia

In office
14 November 1918 – 14 December 1935

Preceded by Position established

Succeeded by [Edvard Beneš](#)

Personal details

Born 7 March 1850
[Hodonín, Austrian Empire](#)
(now [Czech Republic](#))

Died 14 September 1937
[Lány, Czechoslovakia](#)
(now [Czech Republic](#))

Spouse(s) [Charlotte Garrigue](#)
[Alice](#) (1879–1966)
[Herbert](#) (1880–1915)

Children [Jan](#) (1886–1948)
Eleonor (1890–1890)
Olga (1891–1978)

Profession [Philosopher](#)

Signature 

Figure 2.1: Príklad z tréningových dát datasetu WikiBIO

kontextové údaje sú vo vstupnej tabuľke zastúpené len ako informácia v ktorý deň sa zápas konal a vo väčšine sumárov táto informácia nie je použitá, rozhodol som sa ju tiež úplne vynechať z tréningových dát. Všetky číselné údaje v tabuľkách sú tvorcami datasetu upravené na celočíselné hodnoty.

Team Statistics

Medzi tímové údaje rátam jednak mená klubov, mestá, kde kluby sídlia. Štatistiky sú jednak kontextové (počet predchádzajúcich zápasov, ktoré klub vyhral, resp. prehral) jednak zápasové (napríklad počet získaných bodov, percentuálna úspešnosť strelby z poľa atp.). Tímové údaje a štatistiky obsahujú celkovo 15 položiek pre každý tím, tvorcovia datasetu túto časť označujú ako *line score* a ich celý výčet sa dá nájsť na githube tvorcov datasetu ¹ Tabuľka 2.1 z validačného datasetu môže poslúžiť ako dobrý príklad.

Name	City	PTS ₁	AST ₂	REB ₃	TOV ₄	Wins	Losses ...
Raptors	Toronto	122	22	42	12	11	6 ...
76ers	Philadelphia	95	27	38	14	4	14 ...

Note: The statistics are accumulated across all the team players

¹ Points; ² Assists; ³ Rebounds; ⁴ Turnovers

Table 2.1: Príklad tímových štatistík z datasetu Rotowire

Player statistics

U každého hráča je údaj, za ktorý tím hrá, či je daný tím domáci, alebo či hráč patril do starting roster, alebo nastúpil z lavičky. Štatistiky sú len zápasové a medzi ne patria napríklad počet nahraných bodov, počet minút, ktoré hráč hral atp. Hráčske údaje a štatistiky čítajú 24 položiek, tvorcovia datasetu ich nazývajú *box score* a čitateľa, ktorého zaujíma kompletný výpis položiek opäť odporúčame na github tvorcov datasetu ¹ a uvádzame aj príklad tabuľky z validačného datasetu 2.2.

Name	Team City	S_POS ₁	PTS ₂	STL ₃	BLK ₄ ...
Kyle Lowry	Toronto	G	24	1	0 ...
Terrence Ross	Toronto	N/A	22	0	0 ...
Robert Covington	Philadelphia	G	20	2	0 ...
Jahlil Okafor	Philadelphia	C	15	0	1 ...

Note: N/A means that the statistic couldn't be collected because it is undefined

(e.g. player didn't appear on starting roster therefore his starting position is undefined)

¹ Starting position ; ² Points; ³ Steals; ⁴ Blocks

Table 2.2: Príklad hráčskych štatistík z datasetu Rotowire

¹<https://github.com/harvardnlp/boxscore-data>

2.3.2 Summaries

Kladíme vysoké nároky na sumáre. Jednak musia súvisieť s tabuľkou, jednak nechceme neurónovú sieť učiť niečo, čo by sa dalo rýchlo napísať pomocou jednoduchšej šablóny. Preto potrebujeme sumáre, ktoré dokážu vytiahnuť z tabuľky hlbšie štatistiky a správne s nimi pracovať. Ako už vysvetľuje [Wiseman et al., 2017] obyčajné športové spravodajstvo zo stránok <https://www.sbnation.com/nba> nespĺňa tieto kritériá, keďže obsahuje priveľa informácií, ktoré sú založené na iných kontextoch, ako sú dáta v zápasovej tabuľke. Práve preto je zaujímavé sledovať fenomén fantasy športov.

Fantasy sports

Podľa [Tozzi, 1999] môžeme datovať počiatky fenoménu fantasy športu do šesťdesiatych rokov dvadsiateho storočia. Podľa stránky RotoWire² je základom fantasy športu vytvorenie tímov reálnych hráčov z ligy a získavanie bodov na základe ich výkonov v skutočných zápasoch. Bodovanie berie do úvahy, či ide o defenzívneho, alebo ofenzívneho hráča a je nutné vybrať hráča na každú pozíciu v hre. (čím sa zaručuje, že nemôžu existovať fantasy tímy, ktoré majú len point guardov). Disponujete obmedzenými zdrojmi na nákup hráčov a hráči, u ktorých je pravdepodobnejšie, že skórujú viac bodov, resp. získajú viac lôpt a tak podobne, stoja viac. Skutočné pravidlá sú jemne zložitejšie a tí, ktorých by fantasy ligy zaujímali odporúčame na dané stránky²

Fantasy sports news

Na to, aby hráč mohol uspieť vo fantasy lige, potrebuje mať dokonalý prehľad o štatistikách hráčov, o trendoch, o zraneniach, o tímoch, ktorým sa darí aj o tých, ktorým sa možno začne dariť neskôr. Práve preto už od začiatku organizovania fantasy líg existujú formy spravodajstva, ktoré sa špecializujú práve na hráčov fantasy športov. Podľa legendy popísanej [Tozzi, 1999] sa hráči jednej z prvých fantasy líg schádzali v reštaurácii La Rotisserie Francaise, a podľa tejto reštaurácie je pomenovaná aj stránka špecializujúca sa na spracovávanie štatistík pre fanúšikov fantasy líg, <https://www.rotowire.com/>. Články písané o zápasoch oveľa viac berú do úvahy, čo sa v zápase udialo a zároveň poskytujú aj hlbší náhľad do kontextu zápasu. Preto ako píše [Wiseman et al., 2017] je pre generačné systémy ideálnejšie učiť sa generovať práve články podobné tým z RotoWire.

2.3.3 Relation of summaries and tables

V tejto kapitole na jednoduchom príklade ukážem, ako sú štatistiky z tabuľky previazané so sumárom z RotoWire. Vo figure 2.2 je krásne vidieť, že väčšina údajov zo sumáru pochádza z tabuľky. Napriek tomu však text obsahuje niektoré údaje (**zvýraznené žltou farbou**), ktoré v tabuľke vôbec nie sú a niektoré údaje (**zvýraznené modrou farbou**), ktoré sú v tabuľke len implicitne a je ich potrebné odvodiť. Zatiaľ čo fakt, že keďže Terrence Ross má nedefinovanú štartovaciu

²<https://www.rotowire.com/basketball/advice/>
².

pozíciu, tak musel nastúpiť do zápasu z lavičky je celkom zrejmý, informácia o tom, že Joel Embiid nehrá je dôležitá len vtedy, pokiaľ je Joel Embiid natoľko dôležitý hráč pre tím, že stojí za zmienku ho spomenúť. To je však kontext, ktorý v tabuľke spomenutý nie je a navyše informácia o tom, prečo nehrá nemôže byť jasná ani z kontextu celého korpusu dát.

TEAM	WIN	LOSS	PTS ₁	FG_PCT ₂	REB ₃	AST ₄ ...
Raptors	11	6	122	55	42	22
76ers	4	14	95	42	38	27

PLAYER	City	PTS ₁	AST ₄	REB ₃	FG ₅	FGA ₆	S.POS ₇ ...
Kyle Lowry	Toronto	24	8	4	7	9	G
Terrence Ross	Toronto	22	0	3	8	11	N/A
Robert Covington	Philadelphia	20	2	5	7	11	G
Jahlil Okafor	Philadelphia	15	0	5	7	14	C
DeMar DeRozan	Toronto	14	5	5	4	13	G
Jonas Valanciunas	Toronto	12	0	11	6	12	C
Ersan Ilyasova	Philadelphia	11	3	6	4	8	F
Sergio Rodriguez	Philadelphia	11	7	3	4	7	G
Richaun Holmes	Philadelphia	11	1	9	4	10	N/A
Nik Stauskas	Philadelphia	11	2	0	4	9	N/A
Joel Embiid	Philadelphia	N/A	N/A	N/A	N/A	N/A	N/A
...							

The host Toronto Raptors defeated the Philadelphia 76ers , 122 - 95 , **at Air Canada Center on Monday** . **The Raptors came into this game as a monster favorite** and they did n't leave any doubt with this result . Toronto just continuously piled it on , as they won each quarter by at least four points . The Raptors were lights - out shooting , as they went 55 percent from the field and 68 percent from three - point range . They also held the Sixers to just 42 percent from the field and dominated the defensive rebounding , 34 - 26 . Fastbreak points was a huge difference as well , with Toronto winning that battle , 21 - 6 . **Philadelphia (4 - 14) had to play this game without Joel Embiid (rest)** and they clearly did n't have enough to compete with a potent Raptors squad . Robert Covington **had one of his best games of the season though** , tallying 20 points , five rebounds , two assists and two steals on 7 - of - 11 shooting . Jahlil Okafor **got the start for Embiid** and finished with 15 points and five rebounds . Sergio Rodriguez , Ersan Ilyasova , Nik Stauskas and Richaun Holmes all finished with 11 points a piece . **The Sixers will return to action on Wednesday , when they host the Sacramento Kings for their next game** . Toronto (11 - 6) left very little doubt in this game who the more superior team is . Kyle Lowry carried the load for the Raptors , accumulating 24 points , four rebounds and eight assists . **Terrence Ross was great off the bench , scoring 22 points on 8 - of - 11 shooting** . DeMar DeRozan finished with 14 points , five rebounds and five assists . Jonas Valanciunas recorded a double - double , totaling 12 points and 11 rebounds . **The Raptors next game will be on Wednesday , when they host the defensively - sound Memphis Grizzlies** .

Note: ₁ Points; ₂ Field Goal Percentage; ₃ Rebounds; ₄ Assists; ₅ Field Goals; ₆ Field Goals Attempted; ₇ Starting Position; N/A means undefined value

Figure 2.2: Príklad vstupných tabuliek a zlatého sumáru z datasetu. Žltou sú zvýraznené informácie nenachádzajúce sa v tabuľke, zatiaľ čo modrou sú zvýraznené informácie, ktoré z tabuľky a celého korpusu vyplývajú len implicitne.

3. Preprocessing and Statistics of the Datasets

Ako hlavný spôsob riešenia problému generovania prirodzeného textu zo štruktúrovaných dát volíme RNN. RNN sú uspôsobené na spracúvanie sekvenčných, 1D dát, avšak my potrebujeme spracovať 2D tabuľky. V tejto kapitole popíšem spôsob, ako sa s týmto problémom vyrovnávam, pričom sa pokúsim vysvetliť, prečo je postup rozdielny pre dataset WikiBIO a RotoWire. Ďalej budem rozprávať o tom, ako som ďalej upravoval vstupné a výstupné dáta a dôvody pre každú z aplikovaných zmien.

3.1 Transforming Tables to Records

Najprv sa pokúsim ukázať aký cieľ chcem naplniť pri transformácii tabuľky na sekvenčný vstup. Zadefinujeme si tabuľku, s ktorou budeme pracovať. Povedzme, že stĺpce budú označovať typy hodnôt a riadky budú označovať entity. (ako v príklade - referencia na tabuľku). Chceme, aby čo najviac informácií ostalo v dátach. Konkrétne to znamená, že v sekvenčnom vstupe by malo zostať zachované, jednak ku ktorej entite daný vstup patrí, jednak aký typ hodnoty prislúchajúci k danej entite vyjadruje.

3.1.1 Notation

Podľa [Liang et al., 2009] zavádzam notáciu, ktorú budem ďalej používať. Tabuľku \mathcal{T} transformujeme na postupnosť záznamov $\mathbf{s} = \{r_i\}_{i=1}^J$, kde r_i označuje i -ty záznam. Vzhľadom na ciele stanovené vyššie, každý záznam položku $r.f$ označujúcu typ hodnoty, položku $r.v$ označujúcu hodnotu daného záznamu, prípadne položku $r.e$, označujúcu entitu, ktorej prislúcha.

3.2 WikiBIO

V tejto sekcii najprv ukážem, na aké záznamy sa transformuje infobox z WikiBIO datasetu. Následne predstavím štatistiky a preprocessing, ktorý používam.

3.2.1 Transformation of Infoboxes

Každý infobox sa týka práve jednej entity. Jeden typ však môže mať rôznorodý počet hodnôt. Napríklad pre typ definujúci počet rokov v úrade *office* je možnosťou *from 9.3.1999 to 27.5.2021, 1999 - 2021, march 1999 - may 2021* ...

Existujú minimálne dve možnosti ako sa s tým vysporiadať. Prvou je vyhlásiť hodnotu prislúchajúcu jednomu typu za jeden token a prípadne vytvoriť viacero typov lepšie reprezentujúcich to, čo sa v tabuľke nachádza. To je však vzhľadom na veľkosť datasetu (okolo 730 000 párov infobox-sumár, viac ako 7000 rôznych typov) nevhodný prístup. Druhou možnosťou je pridať do záznamu aj pozičnú

informáciu. To je prístup, ktorý zvolili aj tvorcovia datasetu, [Lebret et al., 2016], či autori state of the art riešenia [Liu et al., 2017].

Konkrétne, ku každému záznamu pridáme hodnotu $r.pos$, ktorá vyjadruje poradie vrámci hodnôt prislúchajúcich typu, číslované od 1. Jednoduché pozorovanie ukazuje, že je vhodné pridať ešte aj informáciu o poradí od konca. Napríklad pokiaľ chceme povedať, že niekto skončil svoju činnosť v 2021, je jednoduchšie informáciu vybrať z recordu s typom $r.f == office$, ako určiť, že je potrebné vybrať record s $r.pos == 3$ (1. a 2. príklad), prípadne $r.pos == 5$ (3. príklad). Preto je ku každému záznamu pridaná ešte pozícia od konca $r.rpos$. Takto stačí pre všetky tri prípady vybrať record, kde $r.f == office$ a $r.rpos == 1$.

3.2.2 Dataset statistics

Pre tréning neurónovej siete používam origiálny train-valid-test split od autorov, teda 582 659 trénovacích infobox-sumár párov, 72 831 validačných a 72 831 testovacích.

Sumáre obsahujú celkovo 493 878 unikátnych tokenov, celkovo 18 981 222 tokenov. Tabuľky obsahujú 7 200 unikátnych typov.

3.2.3 Preprocessing

Jednotlivé sety (train, valid, test) ešte prefiltrujem tak, aby žiadna tabuľka nebola dlhšia ako 100 recordov a žiadny sumár nebol dlhší ako 75 tokenov. (urobené na základe štatistík v tabuľke *tu bude odkaz na tabuľku*) Keďže tento dataset nie je nosným datasetom tejto práce, rozhodol som sa neexperimentovať s preprocessingom a použil som hodnoty hyperparametrov, ktoré zvolili [Liu et al., 2017].

V krátkosti zhrniem, čo konkrétne používam. Celý dataset je lowercaseovaný, čiarky, zátvorky, bodky ... sú považované za samostatné tokeny. Všetky okrem najčastejších 20 000 tokenov vybraných [Liu et al., 2017] v sumároch a hodnotách tabuliek nahradím špeciálnymi UNK tokenmi. Rovnako všetky, okrem najčastejších 1480 typov nahradím UNK tokenmi.

3.3 RotoWire

Na rozdiel od datasetu WikiBIO, v tabuľke charakterizujúcej jeden zápas NBA (teda vo vstupe pre generačný model) sa vyskytuje viacero entít a dokonca viacero druhov entít (tímy, hráči). Tieto fakty je potrebné zobrať do úvahy pri spracúvaní tabuliek a ich transformácii na sekvenčný vstup pre RNN.

3.3.1 Transformations of Input Tables

Ako je spomenuté v sekcii 2.3.1, hodnoty v tabuľke môžu byť buď mená tímov, hráčov, miest, alebo celé číslo oznamujúce percentuálnu, či absolútnu hodnotu nejakej štatistiky.

Preto má zmysel použiť rozdielny prístup ku spracovaniu hodnôt ako pri datasete WikiBIO. Konkrétne budeme každú hodnotu považovať za jeden token. Väčšina hodnôt je číselná (konkrétne $\frac{13}{15}$ z typov charakterizujúcich tímy

a $\frac{19}{25}$ z typov charakterizujúcich hráčov. Následne mená tímov sú väčšinou dlhé práve jeden token, až na jednu výnimku, z ktorej sa vyrobí jeden token (*Trail Blazers* \rightarrow *Trail_Blazers*). Podobne pre mená tímov a výnimky (*Oklahoma City* \rightarrow *Oklahoma_City*, *Golden State* \rightarrow *Golden_State*, ...).

Transformations of Player Names

Mená hráčov sú všetky, až na jednu výnimku (*Nene*) viac-tokenové a autori, ktorých prístup ku danému datasetu som bral za referenciu ([Wiseman et al., 2017], [Puduppully et al., 2019]) zvolili odlišný prístup ako ten, ktorý používam ja.

Referenčný prístup používa 2 špeciálne typy, *first_name* a *last_name*. Vďaka nim sa úloha pochopiť, či token *James* odkazuje na krstné meno hviezdneho *James Harden*, alebo na priezvisko legendy *LeBron James* necháva na generálny systém. (Tu by malo význam pridať porovnanie toho, ako to vyzerá u mňa a u nich)

Môj prístup je založený na myšlienke, že už na tak veľmi ťažkom datasete je dôležité vytvoriť čo najjednoduchšiu úlohu pre neurónovú sieť. Preto transformujem sumáre tak, aby meno každého hráča bolo reprezentované práve jedným tokenom. Práve preto strácajú typy *first_name* a *last_name* zmysel a vo vstupnej tabuľke ich nemám.

Entities

Keďže vstupná tabuľka obsahuje informácie o obidvoch hrajúcich tímoch a všetkých hráčoch na súpiskách, každý záznam musí obsahovať aj hodnotu *r.e* popisujúcu, ktorú entitu záznam charakterizuje. Vďaka transformácii mien hráčov a tímov, môžeme ako entitu použiť názov tímu, resp. meno hráča. Okrem toho však ku každému záznamu pridáme špeciálnu položku *r.ha*, ktorá symbolizuje, či sa záznam vzťahuje ku domácemu, alebo hosťujúcemu tímu.

Record Format

Nakoniec teda používam záznamy, ktoré obsahujú položky *r.f* - typ záznamu, *r.e* - entita, *r.v* - hodnota a *r.ha* - domáci/hostia. (tu by mal byť pridaný príklad, ako to vyzerá)

3.3.2 Transformations of Summaries

Pri transformácii sumárov vychádzame z dvoch predpokladov:

- Predpokladáme, že neurónová sieť sa môže naučiť generovať určitý token, pokiaľ sa v tréningových dátach aspoň päťkrát.
- Predpokladáme, že je jednoduchšou úlohou kopírovať dáta z tabuľky, ako generovať zo skrytého stavu.

Kvôli tomu vykonávame také transformácie, aby sa jednak vyskytovalo čo najviac tokenov čo najčastejšie, a aby v sumároch boli používané rovnaké tokeny ako v tabuľke.

Number Transformations

Rovnako ako [Wiseman et al., 2017] a [Puduppully et al., 2019], reprezentujeme čísla len pomocou číslic. Táto transformácia je motivovaná druhým predpokladom a teda tým, že dúfame, že väčšinu čísel bude neurónová sieť kopírovať a nie generovať. To znamená, že napríklad token *five* transformujeme na 5. Používame na to knižnicu `text2num`¹. Niektoré čísla však majú v basketbalovej terminológii špeciálny význam. Špecificky teda netransformujeme číslo *three*, pretože sa často vyskytuje ako súčasť slovných spojení *three pointer*, *three pt range* ...

Player name transformations

Ako už vyplýva z 3.3.1 a z predpokladov, ktoré sme si stanovili, chceme v texte reprezentovať jedného hráča práve jedným tokenom. To platí aj pre extrémne prípady ako *Luc Richard Mbah a Moute*, ktorý bol v datasete reprezentovaný šiestimi rôznymi kombináciami elíps v jeho mene. Najčastejšie však vychádza, že v texte sa najprv hráč spomenie a následne sa už oslovuje len priezviskom a až na 17 prípadov (čo činí 2.4% zo 668 hráčov spomenutých aspoň v jednej tabuľke) je meno hráča vždy zložené z 2 tokenov. (jeden prípad *Nene* je jednotokenový, ostatných 16 je 3 a viac tokenových). (tu pôjde algoritmus, ako vyhľadávam meno hráča)

3.3.3 Dataset Statistics

Keďže transformácie sumárov sú výrazne ovplyvnené štatistikami datasetu, najprv si dovoľím predstaviť to, ako vyzerajú tieto štatistiky.

3.3.4 Byte Pair Encoding

Tu by som chcel predstaviť, o čo ide, ako vyzerá algoritmus, koho implementáciu používam, a ako mi to pomôže v úlohách, ktoré som si stanovil.

¹<https://github.com/allo-media/text2num>

4. Neural Network Architectures

Why only neural network approach is used - mention many authors and papers which approach the problem of NLG by making use of deep neural networks.

4.1 RNN

What it is, creating the representation of sequence, etc. etc. Gradient vanishing and gradient explosion problems.

4.2 LSTM

What it is, how it solves the mentioned problems, cite the paper Massive Exploration of Neural Machine Translation Architectures which experiments with LSTMs, GRUs and vanilla RNNs and shows that LSTMs are the most promising option for the sequence to sequence tasks.

4.3 Attention

What it is, cite Bahdanau, Luong, possible subsection about the input feeding approach. It should select the most relevant entry from the database of match statistics.

4.4 Copy mechanism

What it is, right now I've implemented only Joint Copy mechanism, possibly add Conditional Copy

4.5 Truncated Backpropagation Through Time

Why it is infeasible to generate sequences of average 350 tokens with full back-propagation. Which types of truncated BPTT exist and which I've chosen.

4.6 Beam search

Why greedy search isn't enough, what is beam search, when is it used.

4.7 Transformers

Right now I don't think I'll get this far in my exploration and implementation of DNN architectures.

5. Models

The main goal is to explore the architectures. Therefore each model is manually implemented making use of the tensorflow library

5.1 Tensorflow

What is it, mention the paper introducing it, mention other frameworks and motivation why this is the one which is used.

5.2 Sequence to Sequence architecture

Encoder, decoder. Encoder creates the representation of the input in some meta language, decoder creates the output from the representation.

5.3 Encoder

Mention the embedding and MLP encoding (the main approach used in the ro-towire paper) - MLP is used instead of LSTM in the encoding process, then the 2 initial decoder states are obtained by mean pooling over the MLP encodings of the embedded source records.

5.4 Decoder

2 layer LSTM, embeddings, dimensionality.

5.5 Base Model

Seq2Seq architecture with attention, both Luong style Dot Attention and Bah-danau style Concat Attention are used, input feeding approach. Maybe some pictures.

5.6 Joint Copy model

Uses 2 attention mechanisms. Definitely some pictures.

6. Experiments

This chapter should present the observations about the generated data and the steps taken to improve the generations.

6.1 BLEU

What it is, why do I use such a metric for evaluating the data.

6.2 Manual evaluation

How the summaries for manual evaluation are chosen, how many people do evaluate the predicted summaries.

6.3 Other evaluation approaches

Which approaches are presented in the read papers, which improvements should be made.

6.4 Results of the baseline model

Learned which teams play, what are the greatest stars of each team, although the summaries diverge, only first few sentences from the generated summaries are relevant.

6.5 Dropout

What it is, where I apply the dropout - on the decoder LSTM cells.

6.6 Scheduled Sampling

What it is, how it solves the divergence of the summaries.

6.7 Copy methods

How do they help the model to choose more relevant data from the table, how do they fare in the concurrence of the baseline model.

Conclusion

Bibliography

- Andrej Karpathy. The unreasonable effectiveness of recurrent neural networks, 2015. URL <http://karpathy.github.io/2015/05/21/rnn-effectiveness/>.
- Remi Lebrete, David Grangier, and Michael Auli. Neural text generation from structured data with application to the biography domain, 2016.
- Percy Liang, Michael Jordan, and Dan Klein. Learning semantic correspondences with less supervision. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 91–99, Suntec, Singapore, August 2009. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/P09-1011>.
- Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. Table-to-text generation by structure-aware seq2seq learning, 2017. URL <https://arxiv.org/abs/1711.09724>.
- Ratish Puduppully, Li Dong, and Mirella Lapata. Data-to-text generation with content selection and planning, 2019. URL <https://arxiv.org/abs/1809.00582>.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units, 2016.
- Lisa Tozzi. The great pretenders, 1999. URL http://weeklywire.com/ww/07-05-99/austin_xtra_feature2.html.
- Sam Wiseman, Stuart M. Shieber, and Alexander M. Rush. Challenges in data-to-document generation, 2017. URL <https://arxiv.org/abs/1707.08052>.

List of Tables

2.1	Príklad tímových štatistík z datasetu Rotowire	7
2.2	Príklad hráčskych štatistík z datasetu Rotowire	7