

**УНИВЕРЗИТЕТ У БЕОГРАДУ**  
**ФАКУЛТЕТ ОРГАНИЗЦИОНИХ НАУКА**

**ЗАВРШНИ РАД**

**Примена латентне Дирихлеове анализе за  
моделовање тема у рецензијама хотелских  
услуга**

Ментор:

др Сандро Радовановић

Студент:

Нађа Булајић 2017/0372

Београд, 2022. године

## АПСТРАКТ

У овом раду је спроведено истраживање примене моделовања тема уз помоћ латентне Дирихлеове алокације на примеру рецензија о хотелским услугама спроведених на скупу података прикупљеном са *Tripadvisor* апликације. Спроведено истраживање има за циљ добијање листе тема, као излаз из направљеног модела, које је потребно именовати како би се лакше разумела суштина рецензија. Сагледавањем рецензија уз помоћ написаних тема могуће је, на лакши начин, идентификовати проблеме који се јављају у пословању хотела и истакнути његове добре стране. Истраживање је спроведено кроз неколико корака почевши од припреме скупа података у виду елиминисања неинформативних карактера, токенизације и лематизације скупа података. У наставку је уследило прављење корпуса који се заснива на моделу под називом „врећа речи“, који је затим коришћен као улазни елемент за прављење модела латентне Дирихлеове алокације. Након што је направљени модел као излаз дао листу тема, теме су именоване и искоришћене за детаљну анализу услуга које хотел пружа.

**Кључне речи:** Латентна Дирихлеова алокација, Моделовање тема, Токенизација, Лематизација, Корпус података

## САДРЖАЈ

<b>1. УВОД .....</b>	<b>4</b>
<b>2. ТЕОРИЈСКЕ ОСНОВЕ .....</b>	<b>7</b>
2.1 МОДЕЛОВАЊЕ ТЕМА .....	7
2.2 КОРИШЋЕНИ КОНЦЕПТИ.....	10
2.2.1 Модел врећа речи .....	10
2.2.2 Модел фреквенција појмова – инверзна фреквенција докумената.....	11
2.3 МОДЕЛОВАЊЕ ТЕМА ПРЕ ПОЈАВЕ ЛАТЕНТНЕ ДИРИХЛЕОВЕ АЛОКАЦИЈЕ.....	12
2.3.1 Историјат .....	12
2.3.2 Латентна семантичка анализа .....	13
2.3.2 Пробабилистички латентна семантичка анализа .....	14
2.4 МОДЕЛ ЛАТЕНТНЕ ДИРИХЛЕОВЕ АЛОКАЦИЈЕ .....	15
2.4.1 Почетна дефиниција и основна терминологија.....	15
2.4.2 Латентна Дирихлеова алокација .....	16
<b>3. МЕТОДОЛОГИЈА ИСТРАЖИВАЊА .....</b>	<b>21</b>
3.1 ОПИС ПОДАТАКА .....	21
3.1 ПОСТАВКА ЕКСПЕРИМЕНТА .....	22
<b>4. РЕЗУЛТАТИ ИСТРАЖИВАЊА .....</b>	<b>31</b>
<b>5. ДИСКУСИЈА РЕЗУЛТАТА.....</b>	<b>38</b>
<b>6. ЗАКЉУЧАК.....</b>	<b>50</b>
<b>ЛИТЕРАТУРА.....</b>	<b>52</b>

## СПИСАК СЛИКА

Слика 1: Матрица документ-појам (Pascual, 2019).....	13
Слика 2: Графички приказ ПЛСА модели (Pasupat, 2021).....	14
Слика 3: Изглед тема и докумената (Pascual, 2019) .....	16
Слика 4: Моделовање докумената (Pascual, 2019) .....	16
Слика 5: Графички приказ ЛДА модели (Pasupat, 2021).....	17
Слика 6: Сливовит приказ формуле за ЛДА модел (Serrano, 2020).....	18
Слика 7: Повезивање ЛДА модели са примером (Serrano, 2020).....	18
Слика 8: Расподела докумената по темама у односу на вредност хипер-параметра $\alpha$ (Serrano, 2020) .....	19
Слика 9: Расподела речи и тема уз хипер-параметар $\beta$ (Serrano, 2020).....	19
Слика 10: Кораци изградње ЛДА модели (Ghanoum, 2021).....	22
Слика 11: Различити POS тагови у NLTK библиотеци (Pythonspot, 2016) .....	23
Слика 12: Пример одређивања оптималног броја тема помоћу мере кохерентности приказано графички (Ghanoum, 2021).....	26
Слика 13: Дефинисање функције за израчунавање Џакардове сличности.....	27
Слика 14: Пример одређивања оптималног броја тема помоћу мере кохерентности и Џакардове сличности, приказано графички.....	27
Слика 15: Пример изгледа pyLDavis визуелног модели (Ghanoum, 2021) .....	29
Слика 16: Изглед почетних опсервација скупа података.....	31
Слика 17: Изглед скупа података након додавања нове колоне.....	32
Слика 18: Графички приказ одређивања оптималног броја тема помоћу мере кохерентности .....	33
Слика 19: Визуелни приказ модели.....	34
Слика 20: Графички приказ одређивања оптималног броја тема помоћу мере кохерентности и Џакардијеве сличности.....	35
Слика 21: Визуелни приказ модели.....	36
Слика 22: Графички приказ одређивања оптималног броја тема помоћу мере кохерентности и Џакардијеве сличности након поновног подешавања параметара.....	37
Слика 23: Визуелни приказ модели након поновног одређивања оптималног броја тема.....	37
Слика 24: Примери лоших модели коришћењем визуелизације .....	38
Слика 25: Визуелизација коначног оптималног модели .....	41
Слика 26: Листа доминантних речи за теме број 11 и 17.....	42
Слика 27: Расподела речи 'city' по темама.....	42
Слика 28: Листа најдоминантнијих речи за теме 5, 10 и 19.....	44

## СПИСАК ТАБЕЛА

Табела 1: Приказ вектора речи.....	8
Табела 2: Одређивање значења речи "X" .....	8
Табела 3: Пример изгледа опсервације пре и након припреме податак .....	32
Табела 4: Табеларни приказ првих десет најдоминантнијих речи за сваку тему.....	39
Табела 5: Именоване теме .....	45
Табела 6: Додељивање сваке рецензије најдоминантнијој теми.....	46
Табела 7: Пример расподеле тема по документима.....	47

## 1. УВОД

Машинско учење је процес који се изводи на рачунарима са циљем да се на основу података из прошлости изгради модел који може да унапреди функционисање система као и да, у што већој мери, искористи способност машина да обраде велике количине података за кратак временски периоду уз могућност доношења боље одлуке (Делибашић, Сукновић, Јовановић, 2009).

Као област почело је да се изучава 50-их година прошлог века и у наредних 40 година развијале су се редом: вештачке неуронске мреже и модел перцептрона, експертни системи, затим је уследио период интензивног проучавања алгоритама машинског учења када су настали алгоритми стабла одлучивања. Године 1990. дошло је до формирања области откривања законитости у подацима која обједињује све алгоритме и методе из области машинског учења и статистике који служе за откривање законитости и правила у базама података. У том периоду почиње и развој анализе за откривање законитости у документима и тексту (енг. *Text Mining*) (Делибашић, Сукновић, Јовановић, 2009).

Један од облика откривање законитости у текстуалним подацима је моделовање тема, које се укратко може дефинисати као метод за проналажење група речи (тј. тема) из колекција докумената које најбоље описују информације у колекцији (Nair, 2016).

Најједноставнији метод који је осмишљен за моделовање тема је употреба јединичног кодирања (енг. *one-hot encoder*) чији је главни проблем то што вектор не поседује довољно информација о речи коју представља. Додељивањем контекста у којем може да се појави реч коју представља дати вектор, поменути проблем је решен. Векторима је додељен контекст употребом модела који припадају методама пребројавања (енг. *Count-Based methods*). Почетни корак је да се направи матрица која ће садржати могуће контексте сваке речи. Тиме се наилази на доста неинформативних елемената. Затим је потребно редуковање матрице, одређивање формуле за рачунање елемената матрице и то се постиже применом латентне семантичке анализе (Voita, 2022).

Латентна семантичка анализа (ЛСА) проширује значај контекста. Контекст више није важан само како би приближо вектору значење речи већ и како би се проширило знање о самом документу, а након тога и креирао вектор за сваки документ (Voita, 2022).

Унапређење ЛСА модела је пробабилистичка латентна семантичка анализа (пЛСА), након које се појављује Латентна Дирихлеова алокација (ЛДА).

У овом раду ће бити обрађено моделовање тема, преко ЛДА алгоритама на примеру који се односи на идентификовање тема у скупу рецензија корисника хотелских услуга откривањем образаца (патерна) и понављајућих речи.

Процес се простим објашњењем своди на то да се идентификују речи које се највише појављују у свакој рецензији засебно, на основу којих се долази до закључка које рецензије говоре о сличним стварима и међусобно се групишу (Pascual, 2019).

Израз направљеног модела ће бити листа тема која описује целокупан скуп података и које помажу да се на једноставан начин разуме шта то корисници апликације истичу

као позитивне стране хотела, а шта као негативна искуства и проблеме које би менаџмент хотела требало да реши како би унапредио пословање објекта којим управља.

Целокупна теоријска основа потребна за боље разумевање овог рада описана је у другом поглављу. Како је ЛДА модел део области која се бави моделовањем тема, сам рад је започет проласком кроз област анализе текстуалних података, затим су објашњени ЛСА и пЛСА модели чије проблеме решава ЛДА модел, након чега је детаљно објашњен ЛДА модел. У оквиру другог поглавља представљени су и концепти који нису главна тема овог рада, али их је битно поменути како би се обрађени модели боље разумели.

Треће поглавље се бави методологијом резултата. Читав процес израде модела је подељен на пет корака и детаљним проласком кроз сваки корак појединачно објашњен је редослед за добијање најбољег модела. У току израде су употребљени и одређени концепти, који су такође поменути у овом поглављу, како би се повећао квалитет модела.

Четврто поглавље приказује примену свих концепата објашњених на конкретном проблему, који се односи на моделовање тема у рецензијама хотелских услуга. Приказује детаљан процес припреме података који доводи до израде коначног модела.

У петом поглављу је акценат на анализи модела који се показао као оптималан. У оквиру анализе се теме именују, упоређују једна са другом и на крају се дискутује о примени модела у конкретним ситуацијама и користима које модел пружа.

## 2. ТЕОРИЈСКЕ ОСНОВЕ

### 2.1 МОДЕЛОВАЊЕ ТЕМА

Моделовање тема је техника ненадгледаног машинског учења која је способна да скенира скуп докумената, открије обрасце речи и фраза у њима и аутоматски групише речи и сличне изразе који заједно могу да окарактеришу скуп докумената. Анализа текста која је заснована на вештачкој интелигенцији користи широк спектар метода и алгоритама за обраду природног језика. Једна од њих је анализа тема (енг. *topic analysis*) која се користи за аутоматско откривање тема из текстова. Алгоритми за моделовање тема стварају колекције изрази и речи за које мисле да су повезане, док на доносиоцу одлуке остављају да схвате шта ти односи значе (Pascual, 2019).

Моделовање тема укључује пребројавање речи и груписање сличних образаца речи да би се закључиле теме унутар неструктурираних података. Откривањем образаца као што су учесталост речи и удаљеност између речи, модел теме групише повратне информације које су сличне и речи и изрази који се најчешће појављују. Коришћењем ових информација могуће је да се брзо и лако закључи о чему сваки скуп текстова говори. Како би се то постогло потребни су подаци високог квалитета и велике количине података до којих није увек лако доћи. На крају анализе добија се колекција докумената које је алгоритам груписао заједно (тј. они спадају у једну тему), а тема ће бити представљена кроз групу речи и изрази које је користио да закључи релације (Pascual, 2019).

Моделовање тема има за задатак да додели теме неозначеним текстовима. На пример, ако постоји велика колекција новинских чланака, додељивање ознака категорија тим чланцима, односно навођење од којих се категорија састоји чланак, помаже у бржем разумевању на шта се чланак односи или да се избором жељене категорије, лакше пронађе оно што је од интереса у датом тренутку (Malik, Goldwasser, Johnston, 2020).

Основна претпоставка је да сваки документ садржи статистичку мешавину тема тј. статистичку расподелу тема која се може добити сабирањем свих расподела за све обухваћене теме. Методе за моделовање тема покушавају да открију које су теме присутне у документима колекција и колики је проценат њиховог присуства (Pascual, 2019).

За разлику од људи који лако могу да разумеју значење реченице које прочитају, машинама то није толико једноставно. Да би горе-описан процес био изводљив машинама су потребни вектори карактеристика и неки од начина за њихово проналажење су представљени у даљем тексту.

Овај рад ће се односити на моделовање текста применом латентне Дирихлеове алокације и да би се тачно разумело како је ова метода нашла своју примену у обради природног језика потребно је отићи пар корака уназад. Од најједноставнијег *one-hot* вектора речи, затим додавањем речима контекста у којем могу да се појаве и развијањем вектора докумената дошло се до латентне семантичке анализе.

Унапређење латентне семантичке анализе представља пробабилистичка латентна семантичка анализа, која нас доводи до латентне Дирихлеове алокације.

Најпростији начин је представљање речи кроз *one-hot* векторе. То значи да се  $i$ -та реч у речнику дефинише као вектор који се састоји од цифре 1 на  $i$ -тој позицији и 0 на свим осталим. Дужина вектора је једнака броју речи у речнику. Јасно је да, ако речник садржи много речи, вектор који представља сваку од њих ће бити доста дугачак, што представља први проблем на који се наилази у овом приступу. Такође, следећи проблем се односи на значење тих речи. Вектори немају никакве информације о речима које представљају. На пример, *one-hot* вектор мисли да је реч „пас“ једнако блиска речима „мачка“ и „сто“. У табели испод дато је неколико примера представљања вектора за поједине речи (Voita, 2022).

**Табела 1:** Приказ вектора речи

Реч	Вектор речи
Пас	0...0...0 <b>1</b> 0...0...0
Мачка	0...0 <b>1</b> 0...0...0...0
Сто	0...0...0...0 <b>0</b> 10...

Како би се вектору приближило значење речи коју представља, потребно је да му се додели и контекст у којем та реч може да се појави. Овде наступа хипотеза расподеле по којој речи које се често појављују у сличном контексту имају слично значење.

Ако се замисли ситуација да се у тексту наиђе на реч чије значење није познато (нека буде означена са „ $X$ “), један од начина да се схвати шта та реч значи је да се погледа контекст у којем може да се нађе:

- 1) Тањир са  $X$  је на столу.
- 2)  $X$  се углавном једе за доручак.
- 3) Домаћа  $X$  се сматрају квалитетнијим од куповних.
- 4) Из  $X$  може да се излегне пиле.

Након што се прочитају ове реченице иако првобитно није било познато значење речи „ $X$ “, сада је могуће претпоставити о чему је реч. Човек то може да закључи само читањем наведених реченица, док код машина ту наступа хипотеза расподеле. Прлазе се редови матрице који се односе на речи и упоређују се контексти који се налазе по колонама матрице са контекстима у којима се појављује непозната реч „ $X$ “. Као што и сама хипотеза каже, за реч „ $X$ “ се сматра да има слично значење са речју која се налази у реду са којим има највише преклапања по колонама, односно, највише истих контекста у којима може да се нађе. Из следеће табеле може да се закључи да је тражена реч „*јаје*“.

**Табела 2:** Одређивање значења речи " $X$ "

Реч	1)	2)	3)	4)
Јаје	1	1	1	1
месо	1	0	1	0
Столица	0	0	0	0
...	...	...	...	...
$X$	1	1	1	1



Дакле, примером горе је објашњен један од разлога зашто је битно да вектор не садржи само реч већ и да има идеју о томе шта та реч значи и један од начина да се векторима речи додели контекст у којем могу да се нађу је описан у даљем тексту.

Први корак је конструисање матрице чији ће редови да представљају речи, а колоне контекст у којем та реч може да се нађе. С обзиром на то да је оваква матрица доста велика и да се одређене речи које се у њој налазе могу појавити у свега пар контекста, она садржи доста неинформативних елемената тј. нула. То нас доводи до следећег корака који је потребно да се предузме – редуковање димензионалности матрице (Voita, 2022).

Потребно је да се дефинише могући контекст сваке речи и формула за израчунавање елемената матрице. Постоји више метода пребројавања које у овом кораку могу да се примене: *L-sized window*, *Positive Pointwise Mutual Information*, *Latent Semantic Analysis* (Voita, 2022). Имајући у виду да је Латентна Дирихлеова алокација надоградња латентне семантичке анализе (ЛСА), фокус у овом тренутку прелази на њу.

ЛСА је метод који се користи за анализу докумената. Овде се шири „интересовање“ модела, па контекст речи више не служи само како би се добио вектор речи, већ се користи и у сврху проширења знања о документу као целини што омогућава стварање вектора за сваки документ. ЛСА користи косинусну сличност између вектора докумената како би измерила сличност самих докумената (Voita, 2022). Узима велику матрицу термина (речи) и докумената и конструише семантички простор у којем су термини и документи који су међусобно повезани смештени један близу другог (Deerwester, Dumais, Landauer, Furnas, Harshman, 1990). За редуковање димензија користи сингуларну декомпозицију (енг. *Singular value decomposition SVD*).

Унапређење латентне семантичке анализе је пробабилистичка латентна семантичка анализа (пЛСА). Како ЛСА као излаз модела теме представља само као листу речи, пЛСА додаје речима и вероватноће да припадају одређеној теми. Затим нас унапређење пЛСА модела доводи до латентне Дирихлеове алокације придодавањем Дирихлеових расподела пЛСА моделу.

ПЛДА је генеративни алгоритам јер вероватноћу да се нека тема налази у неком документу –  $p(z|d)$  схвата као параметре модела и не могу се генерисати невиђени документи. Ту долази на ред ЛДА јер он ову вероватноћу третира као расподелу која зависи од неког параметра. У ЛДА тема настаје као расподела речи из унапред дефинисаног речника. Свакој теми се придодаје вероватноћа да садржи сваку реч из речника (Marasović, 2015). Сва три модела игноришу синтаксичке информације и третирају документ као врећу речи (*bag-of-words*) (Pascual, 2019).

Сада је на поједностављен начин описан пут до изградње ЛДА модела, али да би то сигурно било јасно, потребно је да се зађе у детаље неколико битних концепата, латентне семантичке анализе и пробабилистичке латентне семантичне анализе.

## 2.2 КОРИШЋЕНИ КОНЦЕПТИ

### 2.2.1 Модел врећа речи

Врећа речи (енг. *Bag-of-words*) је модел који омогућава да се текст прикаже као нумерички вектор атрибута. Идеја овог модела може да се рашчлани на два корака. У првом се креира речник јединствених токена (на пример, речи) из целог скупа докумената. У другом се конструише вектор атрибута из сваког документа који указује на то колико се пута свака реч појављује у одређеном документу (Malik, Goldwasser, Johnston, 2020).

Имајући у виду да јединствене речи у сваком документу представљају само мали подскуп свих речи у речнику модела вреће речи, вектори атрибута се углавном састоје од нула. Редослед речи у документима није важан (Malik, Goldwasser, Johnston, 2020).

За пример могу да се узму следеће реченице:

1. The sun is shining.
2. The weather is nice.
3. The sun is shining, the weather is nice and one and one is two.

Сада се из ових реченица креира речник:

{'and':0, 'is':1, 'nice':2, 'one':3, 'shining':4, 'sun':5, 'the':6, 'two':7, 'weather':8}

Свака реч је добила свој индекс у виду јединственог броја. Следећи корак је креирање вектора атрибута за сваку од реченица:

1. The sun is shining.  
[0, 1, 0, 0, 1, 1, 1, 0 0]
2. The weather is nice.  
[0, 1, 1, 0, 0, 0, 1, 0, 1]
3. The sun is shining, the weather is nice and one and one is two.  
[2, 3, 1, 2, 1, 1, 2, 1, 1]

Број елемената у вектору атрибута је једнак броју елемената речника који је претходно дефинисан. Свака индексна позиција се односи на реч која се налази на истој позицији у речнику. У случају да реченица не садржи реч која се налази на одређеној индексној позицији, тај елемент ће у вектору атрибута бити означен са нулом, у супротном, уместо нуле писаће број појављивања елемента из речника са тим индексом. На пример, реч „and“ у речнику има индекс нула. У првој и другој реченици се уочава да се ова реч уопште не појављује, самим тим на позицијама са индексом нула, у обе реченице, вредност тог елемента ће бити нула. Са друге стране, у трећој реченици се реч „and“ појављује два пута па ће вредност елемента на позицији са нултим индексом бити два.

Горе-описане вредности, тј. вредности елемената вектора атрибута су заправо фреквенције необрађених речи, односно фреквенција појмова (енг. *term frequency*),

$tf(w, d)$  – колико пута се реч  $w$  појављује у документу  $d$  (Malik, Goldwasser, Johnston, 2020).

Потребно је напоменути да је пре почетка изградње модела вреће речи, јако важно очистити текстуалне податке уклањањем свих нежељених карактера (Malik, Goldwasser, Johnston, 2020).

## 2.2.2 Модел фреквенција појмова – инверзна фреквенција докумената

Модел фреквенција појмова – инверзна фреквенција докумената (енг. *Term frequency-inverse document frequency*,  $tf-idf$ ) је техника која може да се користи за смањивање речи које се често појављују у векторима атрибута (Malik, Goldwasser, Johnston, 2020). Овај метод израчунава фреквенцију узимајући у обзир не само колико су речи учестале у датом документу, већ и колико су учестале у целој колекцији докумената. Речи са високом фреквенцијом у целој колекцији ће бити бољи кандидати за представљање докумената од речи са ниском фреквенцијом без обзира на то колико се пута појављују у појединачном документу. Овај метод се показао као бољи од метода који узимају у обзир само фреквенције речи на нивоу документа (Pascual, 2019).  $Tf-idf$  може да се дефинише као производ фреквенције речи и инверзне фреквенције документа и израчунава се формулом: (Malik, Goldwasser, Johnston, 2020).

$$tf-idf(w, d) = tf(w, d) \times idf(w, d) \quad (1)$$

где се  $tf(w, d)$  односи на фреквенције речи, тј. колико пута се реч  $w$  појављује у документу  $d$ , док се  $idf(w, d)$  односи на инверзну фреквенцију документа и представљена је следећом формулом:

$$idf(w, d) = \log \frac{n_d}{1 + df(d, w)} \quad (2)$$

где је  $n_d$  укупан број докумената, а  $df(d, w)$  број докумената  $d$  који садрже члан  $w$  (Malik, Goldwasser, Johnston, 2020). Додавање константе један је опционо, и служи за доделу вредности која није нула за речи из речника које се не појављују у примеру (ако се на пример, скуп дели на тренинг и тест део, појавиће се речи које се не налазе ни у једном документу тренинг скупа и како дељење са нулом није дозвољено, додавањем константе један се прилагођава вредност).  $\log$  се користи како би се избегло да ниже фреквенције докумената добију превелике тежине (Malik, Goldwasser, Johnston, 2020).

Ако се погледа пример из претходног одељка, реч „is” има највећу фреквенцију речи у трећој реченици, али с обзиром да се појављује и у прве две реченице, може да се закључи да вероватно не садржи никакве корисне дискриминаторне информације и да ће њена  $tf-idf$  вредност бити ниска (Malik, Goldwasser, Johnston, 2020).

## 2.3 МОДЕЛОВАЊЕ ТЕМА ПРЕ ПОЈАВЕ ЛАТЕНТНЕ ДИРИХЛЕОВЕ АЛОКАЦИЈЕ

### 2.3.1 Историјат

Моделовање тема је врста пробабилистичког генеративног модела који се последњих година широко користи у области рачунарских наука са посебним фокусом на откривање законитости у текстуалним подацима и проналажење информација (енг. *Information Retrieval*) (Deerwester, Dumais, Landauer, Furnas, Harshman, 1990). Откривање законитости у текстуалним подацима је већ објашњено раније, док се проналажење информација најједноставније може дефинисати као активност прибављања материјала (углавном текста) који задовољава потребе за информацијама из великих колекција које се чувају на рачунарима (на пример, проналажење информација може бити када корисник унесе упит у систем). Другим речима, представља софтверски програм који се бави организацијом, складиштењем, проналажењем и евалуацијом информација, посебно текстуалних (GeeksforGeeks.org, 2022). Основна методологија коју су предложили приликом истраживања проналажења информација за колекцију података се односи на то да се сваки документ у скупу података своди на вектор реалних бројева који представљају фреквенције појављивања речи у документу (Blei, Ng, Jordan, 2003).

Латентну Дирихлеову алокацију су у машинско учење увели Дејвид Блај, Ендрју Нг и Мајкл Џордан 2003. године. Настао је као унапређење пробабилистичке латентне семантичке анализе (пЛСА). Творац пЛСА је Томас Хофман 2001. године и настао је као унапређење латентне семантичке анализе (ЛСА) (Deerwester, Dumais, Landauer, Furnas, Harshman, 1990). Алтернативни назив за латентну семантичку анализу је латентно семантичко индексирање (ЛСИ) и тај термин се користи када се ради на проблемима везаним за проналажење информација. У случају када се ради на проблемима обраде природног језика, користи се термин латентна семантичка анализа (Savev, 2015).

Као основна техника за изградњу ЛСА/ЛСИ модела коришћена је декомпозиција сингуларних вредности. Поменути статистичка техника, која се већ дуго користи у статистици и линеарној алгебри, је место у обради података нашла као техника која се користи за редуковање димензија матрица података углавном појам-документ (енг. *term-document*) матрица (Savev, 2015).

С обзиром да је, као што је поменуто у тексту изнад, ЛДА надоградња модела пЛСА и ЛСА, они ће бити детаљније објашњени у наредним одељцима.

Такође, да би се избегло понављање, битно је напоменути да се сва три модела заснивају на хипотези расподеле и хипотези статистичке мешавине (један документ може да говори о више тема) (Pascual, 2019).

## 2.3.2 Латентна семантичка анализа

Латентна семантичка анализа (ЛСА) изучава колико се често речи појављују у документима, као и у читавом корпусу података и претпоставља да ће слични документи садржати приближно исту расподелу фреквенције речи за одређене речи. У овом случају, синтаксичке информације (нпр. редослед речи) и семантичке информације (нпр. мноштво значења дате речи) се занемарују и сваки документ се третира као модел вреће речи (Pascual, 2019).

ЛСА карактерише нискодимензионално представљање докумената и речи. Потребно је скренути пажњу да се димензионалност односи на број карактеристика које постоје за опсервације које се користе у моделу. Када је модел нискодимензионалан то значи да је број опсервација већи од броја карактеристика (Pasupat, 2021).

Уз дату колекцију докумената  $d_1, d_2, \dots, d_m$  и речник са речима  $w_1, w_2, \dots, w_n$ , може да се конструише матрицу појмова документа  $X \in \mathbb{R}^{n \times m}$ , где  $x_{ij}$  описује фреквенцију појављивања речи  $w_j$  у документу  $d_i$  (Pasupat, 2021). Метод који се углавном користи за израчунавање фреквенција речи је *tf-idf* (Pascual, 2019) и обрађен је у одељку 2.2.2. Ова матрица ће садржати редове за сваки документ у колекцији и колоне за сваки разматрани контекст (Pascual, 2019).

Као што је већ напоменуто, за редуковање димензија матрице  $X$  користи се сингуларна декомпозиција.

Укратко, за матрицу  $X$ , редуковање димензија помоћу технике сингуларне декомпозиције, може да се изведе уз помоћ формуле:

$$X = UV^T \quad (3)$$

где  $U$  матрица представља документ-тема (енг. *document-topic*) матрицу, а  $V$  матрица појам-тема (енг. *term-topic*) матрицу. По правилима линеарне алгебре,  $\Sigma$  матрица је дијагонална и ЛСА ће сматрати сваку сингуларну вредност, односно сваки број на главној дијагонали ове матрице (свака колона матрица  $U$  и  $V$ ) као потенцијалну тему која се налази у документима (Pascual, 2019).

Document-Term Matrix	Document 1	Document 2	Document 2	Document 2
Lebron	0.4	0	0	0
Senate	0.01	0.9	0	0
Celtics	0.2	0	0	0
Sprain	0	0	0.2	0.2
Cancer	0	0.02	0.3	0.3

Document-Topic	T1	T2	T3	T4
D1	0.8	0.2	0	0
D2	0	0.7	0	0
D3	0.1	0	0	0
D4	0.6	0	0.2	0.2

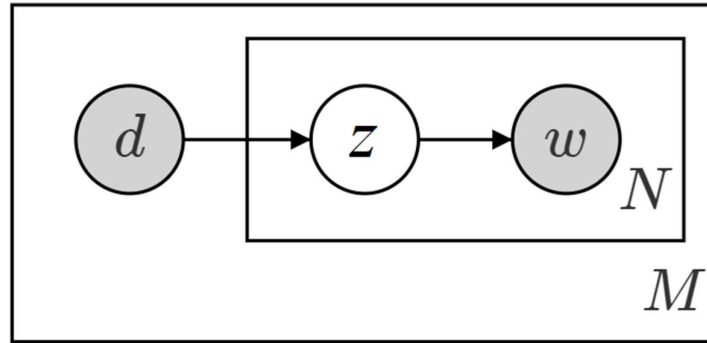
Term-Topics	T1	T2	T3
Lebron	0.8	0.1	0.1
Senate	0	0.7	0.3
Celtics	0.9	0.1	0
Sprain	0.6	0	0.4
Cancer	0		0.7

Слика 1: Матрица документ-појам (Pascual, 2019)

### 2.3.2 Пробабилистички латентна семантичка анализа

За разлику од ЛСА, пробабилистички латентна семантичка анализа (пЛСА), проналази теме помоћу пробабилистичког модела уместо да користи сингуларну декомпозицију (Albanese, 2022).

Графички приказ овог модела, као и формула за његово израчунавање, приказани су у наставку.



Слика 2: Графички приказ пЛСА модела (Pasupat, 2021)

$$P(w|d) = P(d) \sum_z P(z|d)P(w|z) \quad (4)$$

где је:

- $d$  – ознака за документ;
- $z$  – ознака за тему;
- $w$  – ознака за реч;
- $N$  – број речи у документу;
- $M$  – број докумената у читавој колекцији;
- $P(z|d)$  означава вероватноћу да је тема  $z$  садржана у документу  $d$ ;
- $P(w|z)$  означава вероватноћу да се реч  $w$  садржана у теми  $z$ .

На слици изнад, оквири означавају садржај који се понавља. Бројеви у доњем десном углу означавају број понављања. Дакле, формула ће се применити на све речи у документу и на све документе у колекцији. Сиви чворови представљају инстанце у скупу података, док се бели односе на скривене случајне применљиве или параметре. На крају, стрелице означавају зависност (Deerwester, Dumais, Landauer, Furnas, Harshman, 1990).

За сваки документ  $d \in \{1 \dots N\}$  и сваку реч у документу  $d$ , пЛСА прати следећу генеративну процедуру (Deerwester, Dumais, Landauer, Furnas, Harshman, 1990):

- Изабрати  $z \sim P(z|d)$  (насумично бира тему);
- Изабрати  $w \sim P(w|z)$  (насумично бира реч).

Модел може да се прилагоди коришћењем *ЕМ* (енг. *expectation-maximization algorithm*), који врши процену максималне вероватноће у присуству латентних варијабли, у овом случају тема (Albanese, 2022).

Проблем који се јавља је недостатак параметара за вероватноћу  $P(d)$ , тако да није познато како да се додели вероватноћа новом документу. Друга врста проблема која се појављује је проблем претренираности, с обзиром да број параметара за  $P(z|d)$  расте линеарно са бројем докумената (Pasupat, 2021). Ови проблеми су настали услед тога што ЛСА не пружа пробабилистички модел на нивоу документа (Albanese, 2022).

## 2.4 МОДЕЛ ЛАТЕНТНЕ ДИРИХЛЕОВЕ АЛОКАЦИЈЕ

### 2.4.1 Почетна дефиниција и основна терминологија

Латентна Дирихлеова алокација је генеративни пробабилистички модел чији је главни улазни елемент корпус. Основна идеја је да су документи представљени као насумична комбинација латентних тема где је свака тема категорисана преко расподеле речи (Blei, Ng, Jordan, 2003).

Битно је напоменути да ЛДА није нужно везан за текст, већ има примену и на друге проблеме као што је прикупљање података, нпр. подаци везани за колаборативно филтрирање, проналажење слика засновано на садржају и биоинформатика (Blei, Ng, Jordan, 2003).

Пре самог почетка залажења у детаљније објашњавање ЛДА модела потребно је да се представи терминологија која ће бити коришћена. Кључни термини који ће бити помињани у наставку су (Blei, Ng, Jordan, 2003):

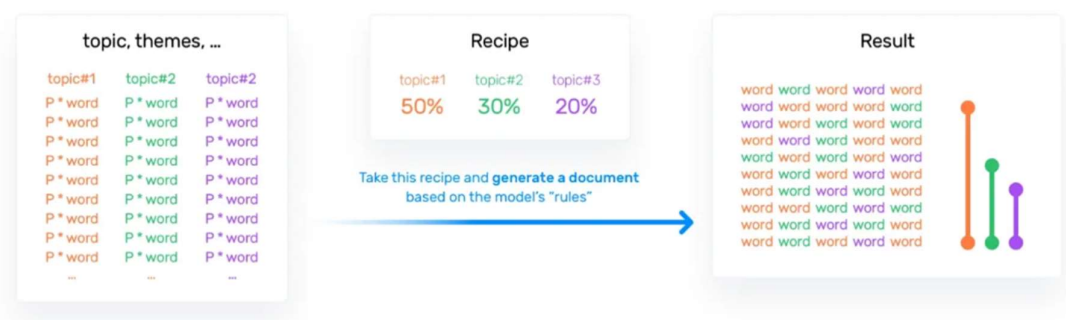
- Реч је основна јединица дискретних података, дефинисана као ставка из речника индексираног са  $\{1, \dots, V\}$ . Представља речи користећи векторе базиране на јединици који имају једну компоненту једнаку јединици и све остале компоненте једнаке нули. Дакле  $v$ -та реч у речнику је представљена са  $V$  димензионим вектором  $w$ , таквим да је  $w^v = 1$  и  $w^u = 0$  и мора да буде испуњен услов:  $u \neq v$ ;
- Документ је основна низ од  $N$  речи означених са  $\mathbf{w} = (w_1, w_2, \dots, w_N)$ , где је  $w_n$   $n$ -та реч у документу;
- Корпус је колекција од  $M$  докумената означених са  $D = \{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_M\}$ .

## 2.4.2 Латентна Дирихлеова алокација

ЛДА модел покушава да пронађе групе речи које се често појављују заједно у различитим документима. Ове речи представљају тему под претпоставком да је сваки документ комбинација различитих речи (Malik, Goldwasser, Johnston, 2020). Такође, претпоставља се и да се свим речима у документу може доделити вероватноћа да припадају некој теми и да је сваки документ написан одређеним распоредом речи (Pascual, 2019).

За ЛДА је карактеристично да се један документ може повезати са више тема (Blei, Ng, Jordan, 2003). Самим тим, сврха ЛДА модела је мапирање сваког документа у колекцији на скуп тема које покривају добар део речи у документу, односно, да утврди коју мешавину тема садржи документ (Pascual, 2019).

Другим речима, ЛДА претпоставља да теме и документи изгледају овако (Pascual, 2019):



Слика 3: Изглед тема и докумената (Pascual, 2019)

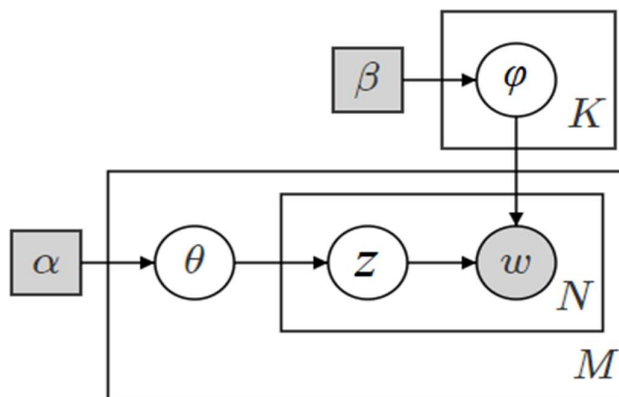
Начин на који ЛДА моделује нови документ изгледа овако (Pascual, 2019):



Слика 4: Моделовање докумената (Pascual, 2019)

ЛДА ће прво бити представљен сликовито, а затим ће бити наведена главна формула пропраћена детаљним објашњењем.





Слика 5: Графички приказ ЛДА модела (Pasupat, 2021)

$$P(\varphi, \theta, w, z | \alpha, \beta) = \prod_{d=1}^M P(\theta_d | \alpha) \prod_{n=1}^N P(z_{dn} | \theta_d) P(w_{dn} | z_{dn}, \varphi) \prod_{k=1}^K P(\varphi_k | \beta) \quad (5)$$

где је:

- $K$  – број тема;
- $N$  – број речи у документу;
- $M$  – број докумената у читавој колекцији;
- $\alpha$  – хипер-параметар који означава Дирихлеову расподелу тема у документу;
- $\beta$  – хипер-параметар који означава Дирихлеову расподелу речи у теми;
- $\theta$  – параметар који означава мултиноминалну расподелу и изводи се из  $\alpha$ ;
- $\varphi$  – параметар који означава мултиноминалну расподелу и изводи се из  $\beta$ ;
- $z$  – ознака за тему;
- $w$  – ознака за реч.

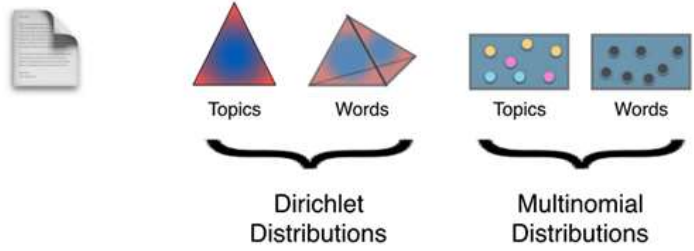
ЛДА прати следећи генеративни процес (Deerwester, Dumais, Landauer, Furnas, Harshman, 1990):

1. За сваку тему  $k \in \{1, \dots, K\}$ :
  - Изабрати  $\varphi_k \sim \text{Dir}(\beta)$ .
2. За сваки документ  $d \in \{1, \dots, M\}$ :
  - Изабрати  $\theta_d \sim \text{Dir}(\alpha)$ .
3. За сваку реч  $w$  у документу  $d$ :
  - Изабрати тему  $z_{dn} \sim \text{Multinomial}(\theta)$ ;
  - Изабрати реч  $w_{dn} \sim \text{Multinomial}(\varphi)$ .

У оквиру ЛДА модела постоје три битна хипер-параметра које је битно боље разумети. Прва два контролишу сличност докумената и тема и познати су као  $\alpha$  и  $\beta$ . Хипер-параметар  $\alpha$  контролише расподелу тема у документу и ниска вредност овог параметра, сваком документу додељује мањи број тема, док у случају да је вредност  $\alpha$  висока, сваки документ ће садржати мешавину великог броја тема. Са друге стране, хипер-параметар  $\beta$  контролише расподелу речи у теми. Његова ниска вредност значи да ће за моделовање одређене теме бити употребљен мањи број речи и обрнуто, висока вредност  $\beta$  значи да ће бити употребљена већина речи за моделовање тема, што доводи до тога да теме буду међусобно сличне. Трећи хипер-параметар се односи на број тема које ће алгоритам детектовати и он се подешава приликом

имплементације ЛДА. Нажалост, алгоритам није у могућности да сам одреди који број тема је потребан (Pascual, 2019).

Са слике 5 може да се види да постоје три нивоа ЛДА репрезентације. Параметри  $\alpha$  и  $\beta$  су параметри на нивоу корпуса, за које се претпоставља да су узорковани једном у процесу генерисања корпуса. Променљива  $\theta_d$  је променљива на нивоу корпуса, узоркована једном за сваки документ, док је  $\phi_k$  узоркована за сваку тему. Трећи ниво чине  $z_{dn}$  и  $w_{dn}$ , променљиве на нивоу речи и узорковане су једном за сваку реч у сваком документу (Blei, Ng, Jordan, 2003).

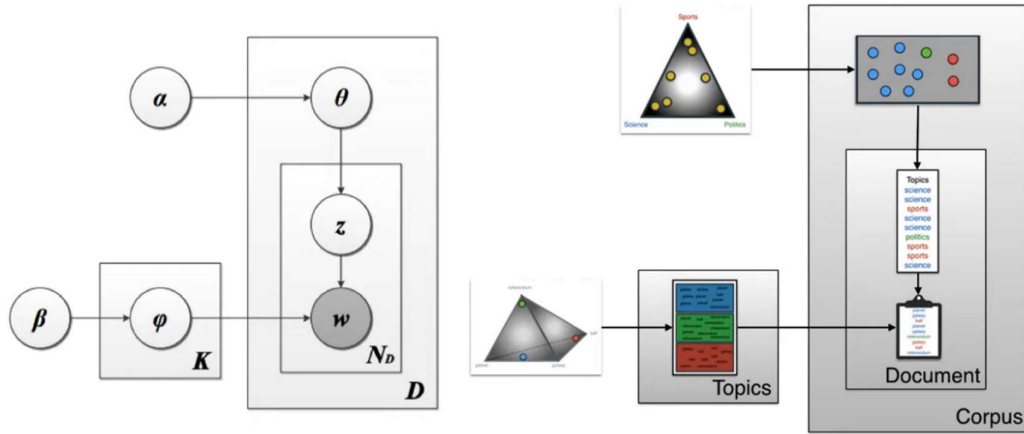
$$P(\mathbf{W}, \mathbf{Z}, \boldsymbol{\theta}, \boldsymbol{\varphi}; \alpha, \beta) = \prod_{j=1}^M P(\theta_j; \alpha) \prod_{i=1}^K P(\varphi_i; \beta) \prod_{t=1}^N P(Z_{j,t} | \theta_j) P(W_{j,t} | \varphi_{Z_{j,t}})$$


**Слика 6:** Сливовит приказ формуле за ЛДА модел (Serrano, 2020)

Ако се погледа ова слика види се да формула може да се рашчлани на неколико делова. Вероватноћа која се рачуна са леве стране је вероватноћа да документ постоји јер машина може да генерише било који документ, а она која генерише документ који је најприближнији оригиналном документу је модел који ће се користити.

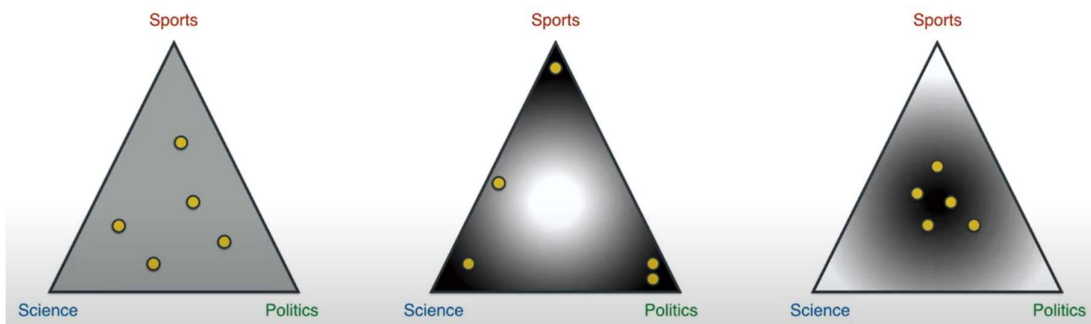
Десна страна формуле састоји се од четири дела. Први и други део се односе на подешавања параметара које се постављају пре почетка прављења модела, док се крајњи део односи на оно што се дешава у току настанка модела.

Уз помоћ  $\alpha$  и  $\beta$ , које су Дирихлеове расподеле, проналазе се теме, док се уз помоћ  $\theta$  и  $\phi$ , које су мултиноминалне расподеле, проналазе речи. Њиховим спајањем могу да се генеришу документи. Ако се ово повеже са графичким приказом модела може да се закључи следеће.



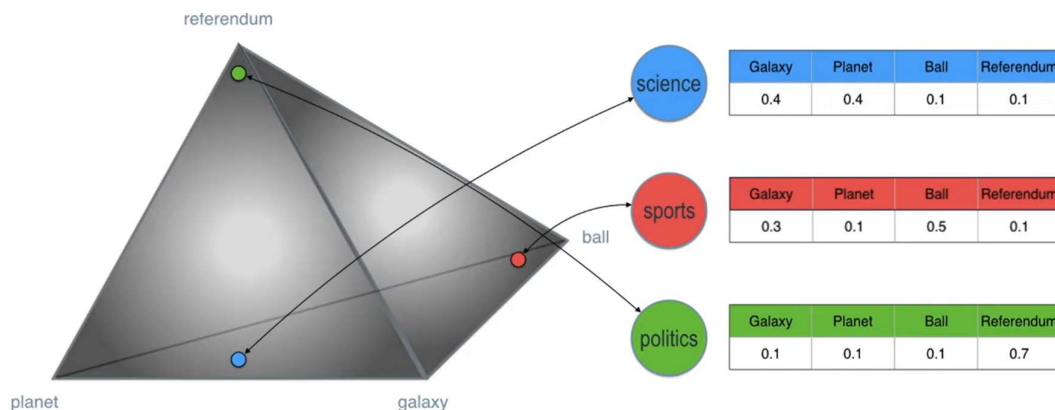
**Слика 7:** Повезивање ЛДА модела са примером (Serrano, 2020)

Хипер-параметар  $\alpha$  распоређује документе по темама. Ако се претпостави да је унапред дефинисано да постоје три теме, за графички приказ ће се користити троугао на чије ће свако теме бити постављена по једна тема. Унутар троугла се распоређују документи. Позиција коју документ заузима у троуглу зависи од вредности на коју је постављено  $\alpha$ . Када је  $\alpha = 1$  (прво троугао са слике 8), документи су униформно распоређени по троуглу, када је  $\alpha < 1$  (други троугао са слике 8), документи су распоређени близу углова, што алудира на то да се вероватно састоје од једне теме или малог броја тема, а ако је  $\alpha > 1$  (трећи троугао са слике 8), документи ће бити распоређени по центру.



**Слика 8:** Расподела докумената по темама у односу на вредност хипер-параметра  $\alpha$  (Serrano, 2020)

Хипер-параметар  $\beta$  распоређује теме између речи. Ради прегледности, ако се замисли да постоје само четири речи, добиће се тетраедар са речима на теменима и теме ће се распоређивати приближно теменима која се односе на речи које садрже.



**Слика 9:** Расподела речи и тема уз хипер-параметар  $\beta$  (Serrano, 2020)

Из  $\alpha$  се добија  $\theta$  – мултиноминална расподела за избор тема, из  $\beta$  се добија  $\phi$  – мултиноминална расподела за избор речи. Затим, се из  $\theta$  добија  $z$  – листа тема, и комбинују се  $z$  и  $\phi$  како би се добила листа речи. Листу речи и тема је могуће објединити у документ и то се понавља онолико пута колико има докумената у корпусу. Када се креира корпус са документима на овај начин, упоређује са постојећим како би се добила одговарајућа вероватноћа.

Израз алгоритма је вектор који садржи покривеност сваке теме за документ који се моделује. Изгледаће отприлике овако: [0.2, 0.5, ...], где прва вредност показује покривеност прве теме, друга друге, и тако даље. Упоредивањем описаних вектора на одговарајући начин, могуће је добити увид у актуелне карактеристике посматраног корпуса (Pascual, 2019).

Приказивање докумената у облику вектора даје могућност да се документи упоређују стандардним мерама сличности у векторском простору, једна од њих је косинусна сличност. Мера сличности може да се објасни као функција реалне вредности која квантификује сличност између два објекта (Marasović, 2015).

Косинусна сличност мери косинус угла између два вектора и с обзиром да су елементи вектора *tf-idf* вредности, које су ненеагитвне, косинусна сличност ће се кретати у опсегу од нула до један. Што је ова вредност ближа јединици, то су два вектора сличнија (Marasović, 2015).

### 3. МЕТОДОЛОГИЈА ИСТРАЖИВАЊА

Поглавље *методологија истраживања* почиње са описом скупа података над којим је истраживање спроведено, а затим ће уследити детаљно објашњење свих корака и концепата који су примењени над подацима. На ово поглавље се надовезује поглавље „*Резултати истраживања*“ које ће приказати резултате добијене применом свих објашњених концепата.

#### 3.1 ОПИС ПОДАТАКА

Како време одмиче и технологија константно напредује, људи се све више ослањају на њу када је потребно да се информишу о одређеној области или донесу одлуке на свим пољима. У пар кликова могуће је добити све врсте података, о било којој теми. Како је остављање рецензија постало све популарније, могуће је да се открију искуства других корисника и често на основу тога и донесе коначна одлука.

*Tripadvisor* је апликација која омогућава да се испланира поподне, викенд, па и дужи одмор за јако кратак временски период. Садржи детаљан опис свих објеката са којима сарађује, па лако може предложити ресторане у близини задате локације за наведени тип хране коју неко жели да једе у ценовном рангу који му одговара. Такође, у случају да жели да организује путовање, потребно је да унесе назив места које жели да посети и на пример, да изабере да жели да борави у хотелу, са доручком или било којим другим условима који су му потребни како би пријатно провео време. Кликком на дугме претражи добиће листу хотела који испуњавају захтеве са препорукама које су оставили други корисници. Зато је потребно да се на одређени начин корисницима апликације приближе искуства других корисника како би једноставније нашли прави избор за себе. Можда неке ствари које је једна особа замерила, другим људима не би представљале проблем ако ће то значити да им је потребан мањи буџет за путовање, па је пожељно да имају приступ разним искуствима како би им доношење одлуке било брже и једноставније.

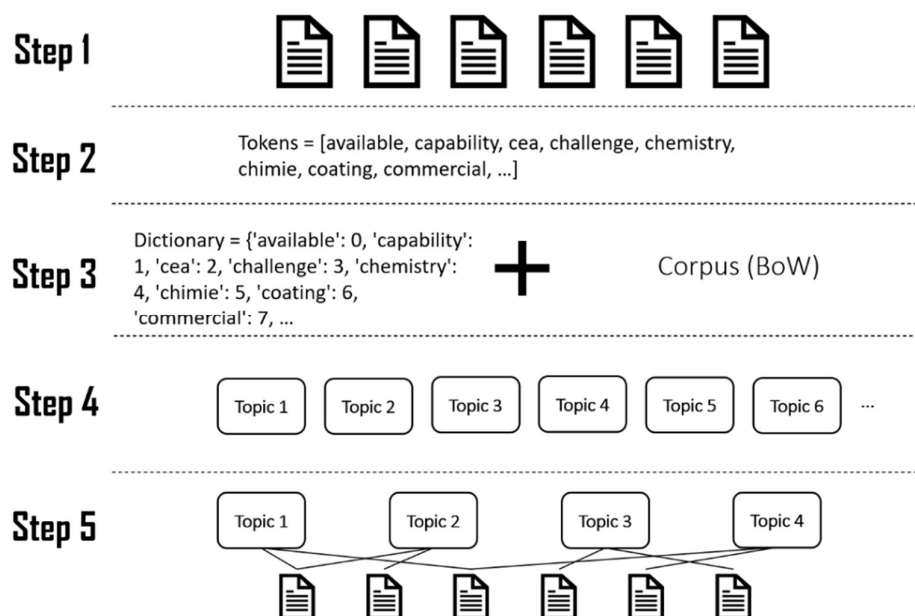
Дефинисање тема од којих се рецензије састоје може бити од користи и за руководство хотела. Ако се претпостави да менаџер жели да открије шта корисници мисле о услугама његовог хотела како би знао шта његов хотел истиче од других или шта је потребно поправити. Ако је у питању неки популарнији хотел, сусрешће се са више хиљада коментара које је потребно прочитати како би се дошло до закључка, што изискује превише времена. Уместо тога, коришћење података кроз направљен модел како би се добиле информације је доста ефикаснија варијанта.

Проблем над којим ће бити спроведено истраживање у овом раду се односи управо на хотелске услуге. Сређивањем података и применом латентне Дирихлеове алокације, а затим анализом добијених резултата, биће откривено које се то скривене теме налазе у великом броју рецензија остављених од стране корисника.

Назив скупа података над којим је вршено истраживање је „*Trip Advisor Hotel Review*“ и преузет је са сајта *Kaggle*<sup>1</sup>, који представља платформу за информисање, унапређивање вештина и стицање знања у области машинског учења и анализе података. Рад је рађен у *Jupyter Notebook* окружењу, у програмском језику *Python*.

### 3.1 ПОСТАВКА ЕКСПЕРИМЕНТА

Како би се направио кратак увод и јасније пролазило кроз целокупно истраживање, цео процес може да се сведе на пет кључних корака представљених на следећој слици.



Слика 10: Кораци изградње ЛДА модела (Ghanoum, 2021)

Кораци које обухвата процес су следећи (Ghanoum, 2021):

1. Прикупљање и учитавање података
2. Претпроцесирање података које се своди на чишћења података и разлагање свих рецензија у токене
3. Прављење речника и корпуса података
4. Проналажење оптималног броја тема и прављење модела
5. Визуелизација и анализа добијених резултата

*Први корак* се односи на учитавање одговарајућег скупа података, над којим ће бити спроведено истраживање уз помоћу *pandas* библиотеке.

*Други корак* који се односи на претпроцесирање података, може да се рашчлани на неколико мањих корака.

<sup>1</sup> <https://www.kaggle.com/datasets/andrewmvd/trip-advisor-hotel-reviews>

Библиотека која је коришћена за припрему података за даљу анализу је *spaCy*. Користи се и њен уграђен модел *en\_core\_web\_md*, који је обучен за рад на енглеском језику (Ghanoum, 2021).

Токенизација текста се односи на раздвајање текстуалног скупа података на појединачне елементе (Malik, Goldwasser, Johnston, 2020). С обзиром да је циљ да модел буде што бољи, ако се сви елементе текстуалног скупа података претворе у токене, добиће се доста неинформативних елемената. Како би се то избегло потребно да је да се очисти скуп података.

Чишћење података подразумева уклањање свих нежељених карактера (Malik, Goldwasser, Johnston, 2020). Теме које се добију на крају истраживања се састоје од низа речи и вероватноћа за сваку реч да припада тој теми. Уколико се не избаце непотребни карактери из скупа података, у анализу ће поред релевантних речи ући и на пример, знакови интерпункције, адреса електронске поште, итд., чиме се знатно смањује могућност да модел ради добро. Зато је потребно да се у анализу уврсте само речи. Такође, потребно је водити рачуна о томе да ли су речи писане великим или малим словима и на пример, пребацити сва слова у мала.

Још једна ствар на коју је битно обратити пажњу, када је све осим речи одстрањено из текста, јесте да ли су све речи које се ту налазе и потребне за креирање модела. Овде долази на ред елиминисање стоп речи (енг. *stopwords*) и одређених врста речи (енг. *part of speech- POS*). Стоп речи су речи које се избацују јер су безначајне и не носе никакве корисне информације, као на пример: *since, off, perhaps* и слично. Врсте речи се у машинском учењу означавају помоћу *POS* тагова, у виду посебних ознака које се додељују свакој речи како би се одредило којој категорији припадају. На следећој слици је дат пример како изгледају.

CC	Coordinating conjunction	NNS	Noun, plural	UH	Interjection
CD	Cardinal number	NNP	Proper noun, singular	VB	Verb, base form
DT	Determiner	NNPS	Proper noun, plural	VBD	Verb, past tense
EX	Existential there	PDT	Predeterminer	VBG	Verb, gerund or present
FW	Foreign word	POS	Possessive ending	participle	
IN	Preposition or subordinating conjunction	PRP	Personal pronoun	VBN	Verb, past participle
		PRP\$	Possessive pronoun	VBP	Verb, non-3rd person singular
JJ	Adjective	RB	Adverb	present	
JJR	Adjective, comparative	RBR	Adverb, comparative	VBZ	Verb, 3rd person singular
JJS	Adjective, superlative	RBS	Adverb, superlative	present	
LS	List item marker	RP	Particle	WDT	Wh-determiner
MD	Modal	SYM	Symbol	WP	Wh-pronoun
NN	Noun, singular or mass	TO	to	WP\$	Possessive wh-pronoun
				WRB	Wh-adverb

Слика 11: Различити POS тагови у *NLTK* библиотеци (Pythonspot, 2016)

Битни концепти у обради природног језика су и лематизација и задржавање (енг. *stemming*) и односе се на нормализацију текста. На пример, није пожељно да модел „*liked*” и „*likes*” посматра као две различите речи само зато што су написане у различитим временима. Из тог разлога је потребно да се сведу на заједнички корен речи, односно да се свака реч врати у основни облик. Задржавање је техника која се користи како би се уклонио суфикс речи. Лематизација је настала развојем задржавања и описује процес груписања различитих речи како би могле да се анализирају као једна ставка, и главна разлика у односу на задржавање је то што

лематизација уноси контекст у реч (Lang, 2022). Како лематизација даје боље резултате, она ће и бити коришћена у овом истраживању.

Токенизацију, лематизацију, уклањање стоп речи и POS тагова је урађено коришћењем *NLTK* библиотеке.

Следећим примером је приказано како изгледа неки текст у оригиналу, а затим како изгледа након извршене припрема података.

*'One of the tell-tale signs of cheating on your Spanish homework is that grammatically, it's a mess. Many languages don't allow for straight translation and have different orders for sentence structure, which translation services used to overlook. But, they've come a long way. With NLP, online translators can translate languages more accurately and present grammatically-correct results. This is infinitely helpful when trying to communicate with someone in another language. Not only that, but when translating from another language to your own, tools now recognize the language based on inputted text and translate it.'* (Tableau.com, n.d.)

Када се избаце сви знакови интерпункције, стоп речи и уврсте само одређене врсте речи у даљу анализу, изврши токенизација и лематизација, оригинални документ ће изгледати на следећи начин:

[*'tell', 'tale', 'sign', 'cheat', 'spanish', 'homework', 'grammatically', 'mess', 'language', 'allow', 'straight', 'translation', 'different', 'order', 'sentence', 'structure', 'translation', 'service', 'overlook', 'come', 'long', 'way', 'online', 'translator', 'translate', 'language', 'accurately', 'present', 'grammatically', 'correct', 'result', 'infinitely', 'helpful', 'try', 'communicate', 'language', 'translate', 'language', 'tool', 'recognize', 'language', 'base', 'inputted', 'text', 'translate'*]

Трећи корак се односи на прављење речника и корпуса, односно модела вреће речи.

Ако се погледају токени (речи) који су добијени види се да се поједини понављају више пута и може да се каже да су те речи битније у документу од осталих. Битно је напоменути да постоји могућност да се та реч појављује у том документу или у јако малом броју документа, самим тим неће имати велики утицај на целокупан корпус података.

Дакле, у овом кораку је потребно да се дефинишу два битна концепта. *Речник* пролази кроз читав скуп података и сваком токenu додељује јединствен број (ИД). Другим речима, пролазећи кроз скуп података свака реч која се до тада није појавила, добиће своје место у речнику као и идентификациони број који је обележава. Речи су организоване по алфабетном редоследу и исто тако им се додељују бројеви. Речник претходног примера приказан је у наставку.

{*'accurately': 0, 'allow': 1, 'base': 2, 'cheat': 3, 'come': 4, 'communicate': 5, 'correct': 6, 'different': 7, 'grammatically': 8, 'helpful': 9, 'homework': 10, 'infinitely': 11, 'inputted': 12, 'language': 13, 'long': 14, 'mess': 15, 'online': 16, 'order': 17, 'overlook': 18, 'present': 19, 'recognize': 20, 'result': 21, 'sentence': 22, 'service': 23, 'sign': 24, 'spanish': 25, 'straight': 26, 'structure': 27, 'tale': 28, 'tell': 29, 'text': 30, 'tool': 31, 'translate': 32, 'translation': 33, 'translator': 34, 'try': 35, 'way': 36*}



Концепт који се у прављењу модела у оквиру овог рада показао корисним је филтрирање речника. *Филтрирање речника* је пожељно урадити у ситуацијама када постоји велики број опсервација у скупу података, а самим тим и велики број елемената речника. С обзиром да је циљ да оптималан број тема који буде одређен моделом обухвати већи број опсервација како би модел био меродаван, пожељно је избећи могућност да се добије тема која ће се појавити у јако малом броју опсервација. Да би се то избегло добра пракса је да се одмах након што се речник направи уради филтрирање речника. Тиме, могу да се уклоне речи које се појављују у, на пример, мање од 1% опсервација (подршка речи у скупу података је мања од једног процента) или да се у модел уврсте само речи које се налазе у, на пример, 50% опсервација. Битно је да се уклоне речи са виским фреквенцијама јер као што није од интереса да се добију теме које покривају свега неколико докумената, потребно је уклонити и могућност да теме садрже речи које се појављују у сваком документу, јер тиме долази до великог преклапања између тема.

Ова опција, као и прављење самог речника, је омогућена у оквиру *gensim* библиотеке и постоје три параметра чију је вредност важно одредити (Ghanoum, 2021):

- *no\_below*: Токени који се појављују у мање докумената од задатог броја се избацују из речника;
- *no\_above*: Токени који се појављују у документима у проценту већем од наведеног се избацују из речника;
- *keep\_n*: Лимитирање речника на одређен број елемената које ће да садржи; у случају да се додели вредност *None*, приказаће све токене (подразумевана вредност је 100.000).

На овај начин ће се осигурати да се у темама, као доминантне речи, појављују речи које су релевантне за читаву колекцију података и може да се направи корпус. Прављење корпуса је такође урађено уз помоћ *gensim* библиотеке.

*Корпус* се прави на основу модела вреће речи, који је описан у одељку 2.2.1. Након што је сваком токenu додељен јединствени ИД, корпус садржи сваки тај ИД и његову фреквенцију појављивања у датом документу.

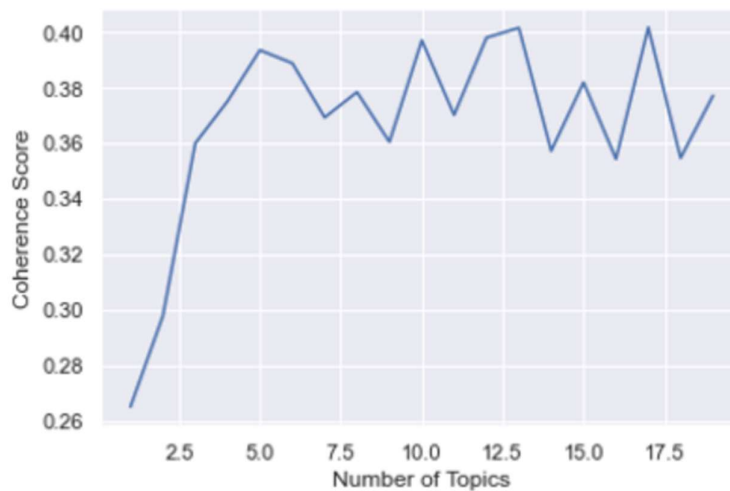
[(0, 1), (1, 1), (2, 1), (3, 1), (4, 1), (5, 1), (6, 1), (7, 1), (8, 2), (9, 1), (10, 1), (11, 1), (12, 1), (13, 5), (14, 1), (15, 1), (16, 1), (17, 1), (18, 1), (19, 1), (20, 1), (21, 1), (22, 1), (23, 1), (24, 1), (25, 1), (26, 1), (27, 1), (28, 1), (29, 1), (30, 1), (31, 1), (32, 3), (33, 2), (34, 1), (35, 1), (36, 1)]

Када је направљен корпус, који представља улазни елемент за прављење модела, долази се до *четвртог корака*, који се односи на одређивање оптималног броја тема и прављење модела.

Једна од мана овог модела је то што број тема мора унапред да се одреди. Како би се избегло нагађање без икаквих основа у оквиру овог рада проналажење оптималног броја тема је урађено на два начина.

1. Мера кохерентности 'с\_v'
2. Кохерентност у комбинацији са Џакардовим индексом

За скуп изјава или чињеница каже се да је *кохерентан* ако међусобно подржавају једни друге. *Мере кохерентности* мере степен семантичке сличности речи које имају велику вероватноћу да се нађу у некој теми. Користи се за процену квалитета тема и квалитет се одређује уз помоћ статистика и вероватноћа извучених из корпуса, са посебним фокусом на контекст речи. Постоји доста алгоритама за израчунавање кохерентности у оквиру *gensim* библиотеке, али онај који је коришћен је ‘*c\_v*’. Оцена кохерентности за њега се креће од нула (потпуна некохерентност) до један (потпуна кохерентност). Све вредности преко 0,5 се могу сматрати прихватљивим. Мера кохерентности као улазне параметре прима направљени модел са темама и корпус, док излаз представља укупна кохерентност тема (Pedro, 2022). Мере кохерентности се рачунају за задати опсег броја тема и затим из графичког приказа може да се закључи који број тема је оптималан за модел. Један визуелни приказ коришћеног алгорита дат је на слици испод.



**Слика 12:** Пример одређивања оптималног броја тема помоћу мере кохерентности приказано графички (Ghanoum, 2021)

Током рада коришћена мера кохерентности није давала поуздане резултате око тога који број тема је оптималан па је испробана још једна опција за добијање оптималног броја тема и поред ‘*c\_v*’ мере кохерентности, израчунат је и Џакардов индекс сличности.

*Џакардова сличност* се рачуна између главних речи две теме. Када би се десило да две теме имају идентичне главне речи, Џакардова сличност би била један. Док је најнижа вредност коју може да има, у случају када су речи потпуно различите нула. Мана овог модела је што у случају да се користе исте речи у обе теме али да постоје мање варијације у вероватноћама да припадају темама, неће бити препознато (Mantyla, Claes, Farooq, 2018). Може се дефинисати као величина пресека два скупа узорака А и В подељена са величином уније иста два скупа и рачуна се на следећи начин:

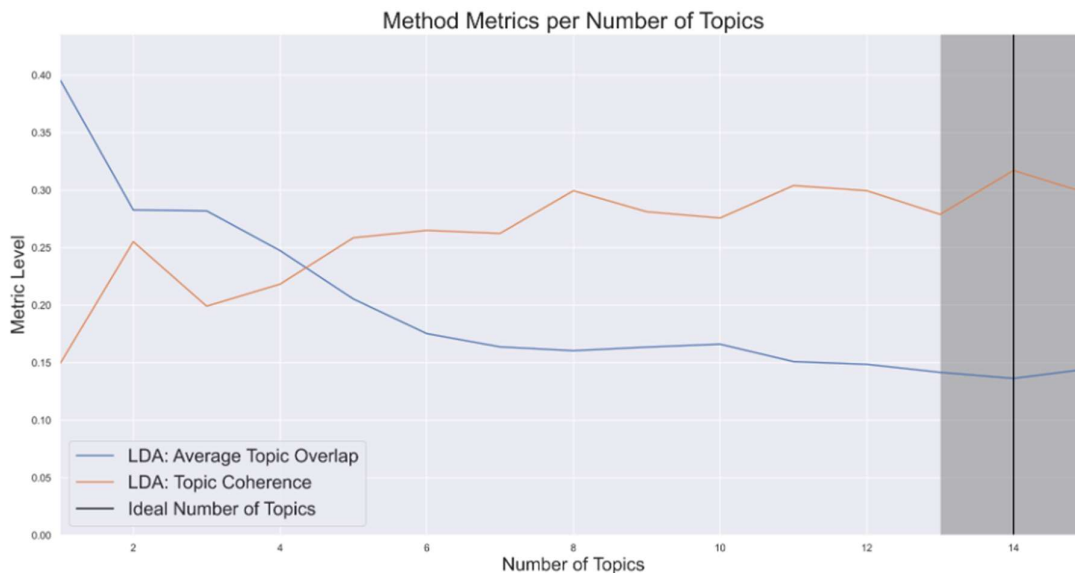
$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (6)$$

Дакле како Џакардова сличност служи за статистичку процену различитости и како је циљ да теме немају велики степен преклапања, у интересу је да има ниску вредност. На следећој слици биће приказана имплементација у *Jupyter Notebook* окружењу.

```
def jaccard_similarity(topic_1, topic_2):  
  
    intersection = set(topic_1).intersection(set(topic_2))  
    union = set(topic_1).union(set(topic_2))  
  
    return float(len(intersection))/float(len(union))
```

**Слика 13:** Дефинисање функције за израчунавање Џакардове сличности

Како је у пожељно да кохерентност има што вишу вредност, а Џакардова сличност што нижу, оптималан број тема се налази тако што се обе мере сличности представе графички, након чега се одреди где је највећа разлика између њих.



**Слика 14:** Пример одређивања оптималног броја тема помоћу мере кохерентности и Џакардове сличности, приказано графички

Слика 14 приказује пример како би изгледао описани графикон. Сива линија приказује да је највећа разлика између ове две мере сличности када је оптималан број тема 14.

Главна разлика између мера кохеренције и Џакардове сличности је у томе што мере кохеренције гледају сличност између речи у теми и ако су те речи међусобно кохерентне сматра тему добром, док се код Џакардове сличности гледа однос две теме. Зато је ово добра пракса за одређивање оптималног броја тема јер даје увид у квалитет тема како појединачно, тако и међусобно.

Како би се потврдило да је одређени број тема стварно оптималан, још један концепт који може да се користи је перплексност (енг. *Perplexity*). Перплексност је статистичка метрика за мерење перформанси језичких модела. Представља количину збуњености и што је цифра већа, то су нови подаци изненађујући, тако да је пожељно

да вредност буде ниска (Stodel, 2020). Може да се израчуна уз помоћ следеће формуле:

$$PP(W) = P(w_1 w_2 \dots w_N)^{-\frac{1}{N}} = \sqrt[N]{\frac{1}{P(w_1 w_2 \dots w_N)}} \quad (7)$$

где је  $W$  реченица направљена од низа речи  $(w_1 w_2 \dots w_N)$  величине  $N$ .

Када је одређен оптималан број тема, остаје да се направи модел. Модел се прави уз помоћу *LdaMulticore* из *gensim* библиотеке. *LdaMulticore*<sup>2</sup> користи сва језгра *CPU*-а за паралелизацију и убрзање обуке модела. Паралелизација користи вишепроцесорску обраду која омогућава да систем покрене више процеса одједном, тј. да разбије процес израде модела у мање нити које могу да раде независно. Замена за овај алгоритам је *LdaModel*, такође из *gensim* библиотеке, који је еквивалентан али користи једно језгро, што продужава време прављења модела.

*LdaMulticore* прима следеће параметре:

- *corpus* – приликом израде модела алгоритам анализира корпус како би се пронашла расподела речи у свакој теми и расподела тема у сваком документу; у случају да није дат модел остаје не обучен;
- *num\_topic* – број тема који је потребно издвојити из корпуса;
- *id2word* – речник направљен у претходном кораку; пресликава ИД-јеве речи у речи; користи се за одређивање величине вокабулара, за отклањање грешака и штампање тема;
- *workers (optional)* – број језгра која ће учествовати у изради модела; препоручује се да се за израду модела користи једно језгро мање од броја језгра који поседује компјутер;
- *chunksize (optional)* – број докумената који се посматрају одједном; подразумевана вредност је 2000;
- *passes (optional)* – број пролазака кроз цео корпус током обуке модела; подразумевана вредност је 1;
- *iterations (optional)* – максималан број итерација кроз корпус током израде модела; тј. колико често ће одређени део корпуса пролази током обуке; подразумевана вредност је 50;
- *random\_state (optional)* – корисно у случају поновног прављења истог модела.

Горе-описани параметри су они који су коришћени за израду модела, поред њих постоје и *alpha*, *eta*, *decay*, *offset*, *eval\_every*, *minimum\_phi\_value*, *per\_word\_topics*, *dtype*, *gamma\_threshold* и *minimum\_probability*.

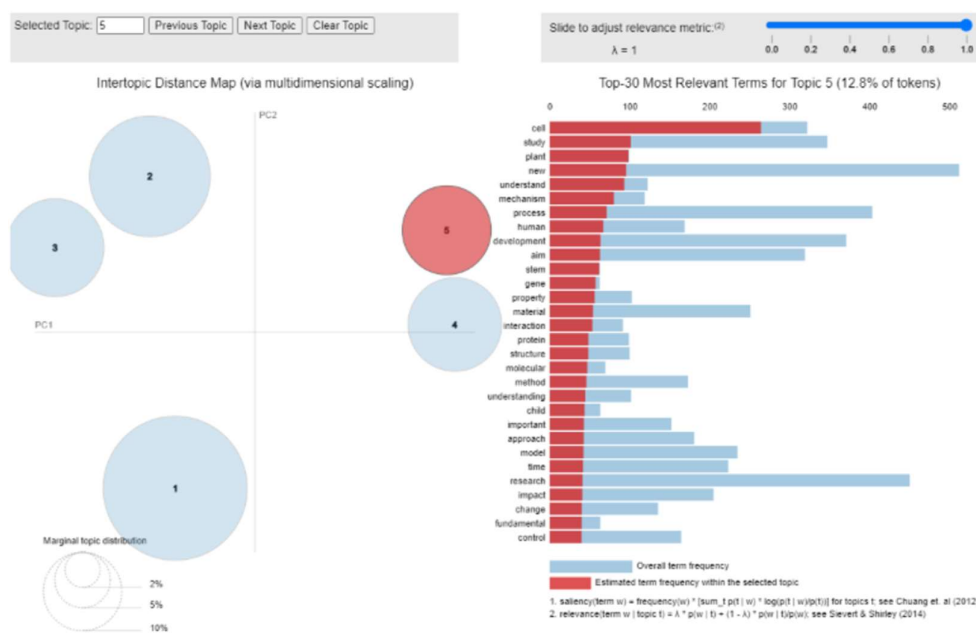
Излаз из ЛДА модела је листа тема, где је свака тема представљена кроз речи са вероватноћом за сваку реч да припада тој теми, сортирано у опадајућем редоследу. Пример како изгледа једна тема ЛДА модела је дат у наставку.

<sup>2</sup> <https://radimrehurek.com/gensim/models/ldamulticore.html>

(0, '0.015\*'datum' + 0.013\*'service' + 0.010\*'research' + 0.010\*'support' + 0.009\*'european' + 0.009\*'network' + 0.009\*'social' + 0.009\*'people' + 0.008\*'provide' + 0.008\*'platform''))

Пети корак у овом истраживању се састоји из визуелизације и анализе добијених резултата. Овде је могуће да се оде пар корака уназад ако је уочено да је нешто потребно кориговати. Такође, у овом кораку се именују теме на основу њихових главних речи, посматра се расподела тема у целокупном корпусу и извлаче се закључци који ће потврдити корисност прављења оваквих модела.

За визуелизацију тема се користи *pyLDavis*<sup>3</sup>, такође из *gensim* библиотеке. *pyLDavis* је дизајниран да помогне корисницима да тумаче теме у моделу тема који је прилагођен корпусу текстуалних података. Користан је за добијање прегледа модела, пажљиво посматрање тема и преглед речи које су повезане са темама. У свом интерфејсу садржи „*relevance metric*“ која омогућава кориснику да прилагоди приказ речи у теми ради бољег разумевања. (24)



Слика 15: Пример изгледа *pyLDavis* визуелног модела (Ghanoum, 2021)

Визуелни приказ ЛДА модела може да се посматра из два дела.

Лева страна, „*Intertopic Distance Map*“, приказује различите теме и растојање између њих. Сличне теме су позициониране једна ближе другој, а различите теме су удаљене. Величина круга одговара учесталости теме коју представља у корпусу. Избором одређеног круга добија се детаљнији приказ теме коју представља на десној страни.

Десна страна приказује 30 најбитнијих речи изабране теме на графикону. У случају да ниједна тема није изабрана, на графикону ће бити представљено 30

<sup>3</sup> <https://pyldavis.readthedocs.io/en/latest/readme.html>

најзаступљенијих речи у читавом корпусу. Овде постоје два битна термина, истакнутост (енг. *saliency*) и релевантност (енг. *relevance*). Истакнутост је мера колико је термин чест у корпусу. Релевантност речи приказује колико је реч „битна“ у означеној теми. Може се подесити параметром  $\lambda$ , где мања вредност овог параметра даје већу тежину појмовима, док већи одговара вероватноћи појављивања речи по теми.

Овим кораком је завршено представљање изградње ЛДА модела и у наредном поглављу ће се проћи кроз изградњу модела на конкретном примеру. Цео процес ће бити детаљно описан, са свим изменама и проблемима до којих је дошло у току рада. У њему ће бити приказана прва четири корака истраживања док ће пети корак бити обрађен у поглављу „Дискусија резулата“.

## 4. РЕЗУЛТАТИ ИСТРАЖИВАЊА

У овом поглављу ће бити приказани резултати истраживања чији је поступак објашњен у претходном поглављу, 3.2 *Поставка експеримента*. Примери из тог поглавља нису повезани са скупом података који ће бити анализиран у овом поглављу и служили су да се прикаже како може да изгледа резултат практичне примене обрађених концепата. Ради лакшег повезивања садржаја ова два поглавља, резултати истраживања ће, као што је и поставка, бити представљени кроз истих пет корака:

1. Прикупљање и учитавање података
2. Претпроцесирање података које се своди на чишћења података и разлагање свих рецензија у токене
3. Прављење речника и корпуса података
4. Проналажење оптималног броја тема и прављење модела
5. Визуелизација и анализа добијених резултата

*Први корак*, односно сам почетак истраживања, почиње учитавањем скупа података „Trip Advisor Hotel Review“, који је већ поменут у поглављу 3.1. Садржи 20.491 опсервација, везаних за утиске које су корисници оставили о хотелу у којем су боравили, које ће бити анализирани. Скуп података долази са две колоне, „Review“, која садржи текстуалне податке и „Rating“ која садржи оцене од 1 до 5 остављене уз рецензију. Колона „Rating“ ће бити изостављена из анализе јер је фокус рада на моделовању тема и ЛДА методи. За нека каснија истраживања и унапређење модела, ова колона може да се покаже врло корисно и да допринесе прављењу бољег модела. У наставку ће бити приказано неколико опсервација почетног скупа података, ради бољег увида у то како отприлике изгледају подаци над којима ће бити вршена анализа.

	Review
0	nice hotel expensive parking got good deal sta...
1	ok nothing special charge diamond member hilt...
2	nice rooms not 4* experience hotel monaco seat...
3	unique, great stay, wonderful time hotel monac...
4	great stay great stay, went seahawk game aweso...
5	love monaco staff husband stayed hotel crazy w...
6	cozy stay rainy city, husband spent 7 nights m...
7	excellent staff, housekeeping quality hotel ch...

**Слика 16:** Изглед почетних опсервација скупа података

У *другом кораку* је урађена припрема података за прављење корпуса. Припрема је рађена следећим редом:

- провера да ли у скупу података постоје недостајуће вредности – не постоји ниједна опсервација која садржи недостајућу вредност;

- трансформисање свих речи у речи написане малим словима;
- избацивање свих елемената из скупа података који нису речи;
- избацивање стоп речи.

Затим је урађена токенизација, уклањање стоп речи које се налазе у оквиру *NLTK* библиотеке, лематизација токена и на крају избацивање свих речи које не припадају изабраним *POS* тагова, односно свих речи које нису именице, глаголи и придеви. Није лоше напоменути и да је испробано прављење модела са још неким *POS* таговима, али се ова комбинација показала као најкориснија јер помаже да се ближе опише тема. Стоп речи које су избачене су оне које се налазе у *NLTK* библиотеци.

Пример опсервације пре и након извршене припреме је у наставку:

**Табела 3:** Пример изгледа опсервације пре и након припреме податак

Опсервација пре припреме:	Опсервација после припреме:
<i>'nice hotel expensive parking got good deal stay hotel anniversary, arrived late evening took advice previous reviews did valet parking, check quick easy, little disappointed non-existent view room room clean nice size, bed comfortable woke stiff neck high pillows, not soundproof like heard music room night morning loud bangs doors opening closing hear people talking hallway, maybe just noisy neighbors, aveda bath products nice, did not goldfish stay nice touch taken advantage staying longer, location great walking distance shopping, overall nice experience having pay 40 parking night,</i>	<i>['hotel', 'expensive', 'parking', 'deal', 'hotel', 'anniversary', 'advice', 'previous', 'valet', 'parking', 'check', 'quick', 'easy', 'little', 'disappointed', 'existent', 'view', 'room', 'room', 'clean', 'size', 'bed', 'comfortable', 'woke', 'stiff', 'neck', 'high', 'pillow', 'soundproof', 'heard', 'music', 'room', 'morning', 'loud', 'bang', 'door', 'opening', 'closing', 'hear', 'hallway', 'noisy', 'neighbor', 'aveda', 'bath', 'product', 'goldfish', 'touch', 'advantage', 'longer', 'great', 'distance', 'shopping', 'overall', 'experience', 'pay', 'parking']</i>

У скуп података је сада додата нова колона „*Review\_clean*“, која сваку рецензију колоне „*Review*“ приказује у виду листе токена. Скуп података сада изгледа на следећи начин:

	Review	Review_clean
0	nice hotel expensive parking got good deal sta...	[nice, hotel, expensive, parking, good, deal, ...
1	ok nothing special charge diamond member hilt...	[ok, nothing, special, charge, diamond, member...
2	nice rooms not 4* experience hotel monaco seat...	[nice, room, experience, hotel, monaco, seattl...
3	unique, great stay, wonderful time hotel monac...	[unique, great, stay, wonderful, time, hotel, ...
4	great stay great stay, went seahawk game aweso...	[great, stay, great, stay, seahawk, game, awes...
5	love monaco staff husband stayed hotel crazy w...	[love, monaco, staff, husband, stayed, hotel, ...
6	cozy stay rainy city, husband spent 7 nights m...	[cozy, stay, rainy, city, husband, spent, nigh...
7	excellent staff, housekeeping quality hotel ch...	[excellent, staff, quality, hotel, staff, make...

**Слика 17:** Изглед скупа података након додавања нове колоне

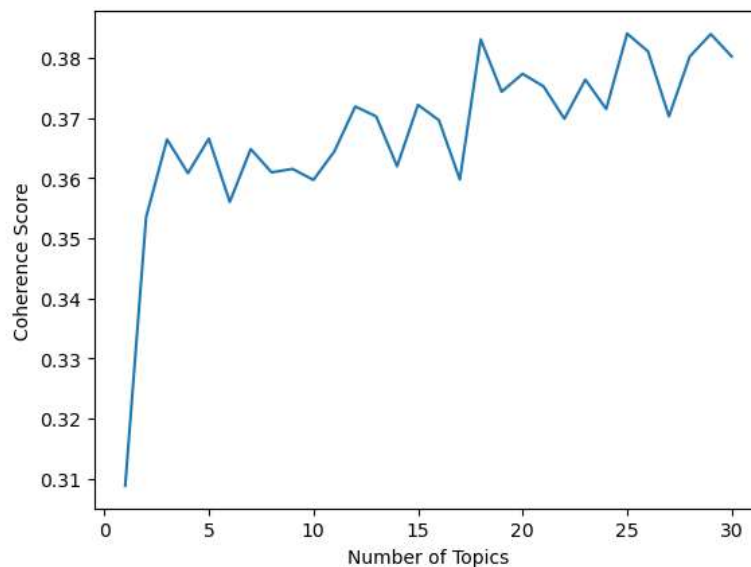


Сада се прелази се на *трећи корак* и прво се прави речник. Речник без икаквог филтрирања има 33.511 елемената. Како је јасно да је овај речник превелик да би се користио у прављењу корпуса и да су велике шансе да користећи њега у темама доминирају речи које се појављују у готово свим опсервацијама, чиме ће се теже разликовати теме, док ће речи које се једва појављују у речнику непотребно продужити трајање израде модела, потребно је да се изврши филтрирање речника.

Испробавањем више комбинација, *no\_below* параметар је постављен на 150 јер се то показало као добра граница с обзиром да је након тога остало сасвим довољно речи у речнику које добро описују теме. Параметар *no\_above* је постављен на 80% и након извршеног филтрирања број речи осталих у речнику је 1.301.

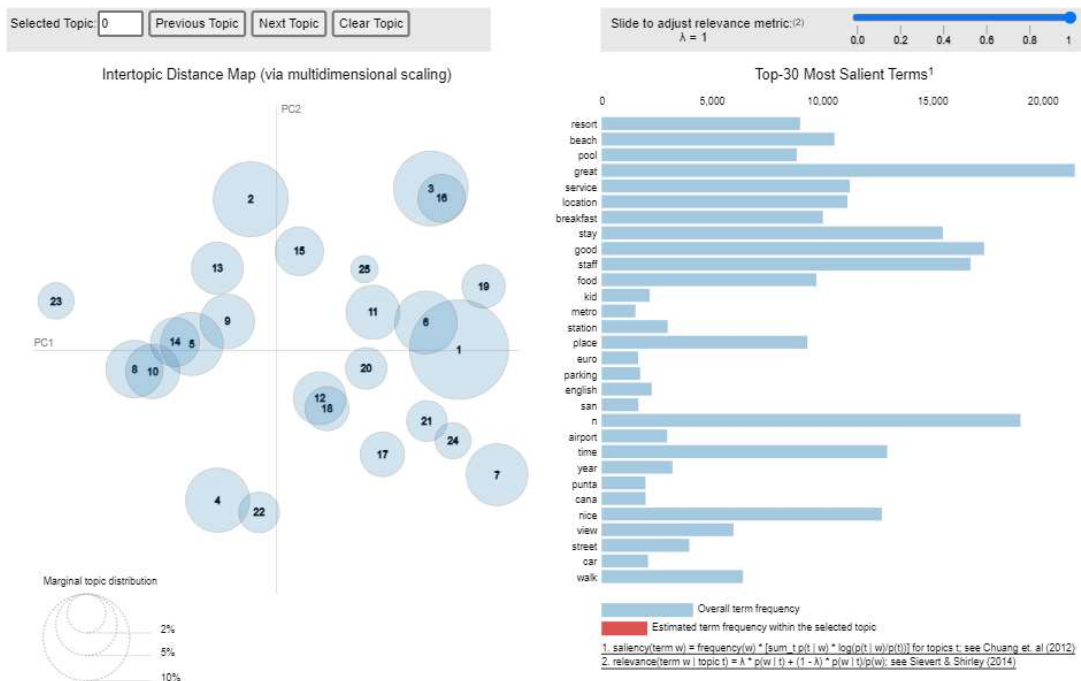
Следећи корак је да се уз помоћ речника креира корпус, који представља улазни параметар за алгоритам. Ово доводи истраживање до *четвртог корака*, прављења модела.

Да би се направио модел потребно је да се нађе оптимална вредност за хипер-параметар број тема, које представљају излаз из модела. Проналажење оптималног броја тема ће бити урађено коришћењем мере кохерентности '*c\_v*'. Кохерентност се рачуна за 30 модела, са истим подешавањима свих параметара осим броја тема, који ће бити у опсегу од 1 до 30.



**Слика 18:** Графички приказ одређивања оптималног броја тема помоћу мере кохерентности

Као резултат је добијено да је 25 оптималан број тема и кохерентност овог модела износи 0,3841. Посматрањем само кохерентности може да се закључи да је овај модел лош, али како би се то потврдило биће приказана и визуелизација модела представљена помоћу *pyLDavis*.



Слика 19: Визуелни приказ модела

Прва ствар по којој овај модел може да се прогласи неодговарајућим је избор речи које улазе у модел. Како је почетни речник садржао неке елементе који не припадају ниједном POS тагу, јер заправо и не постоје и настале су у процесу токенизације потребно је да се поново обави сређивање речника и испочетка приступи прављењу модела.

Из тог разлога се истраживање враћа на *трећи корак* где се покушава да се речник учини погоднијим за анализу. Спуштањем *no\_above* границе на 50%, губи се само неколико речи па ће та граница да се користи у наставку, док је *no\_below* границу постављена на 200. Затим су избачене неке додатне речи које нису од посебног значаја у одређивању тема. Речи које су избачене су следеће:

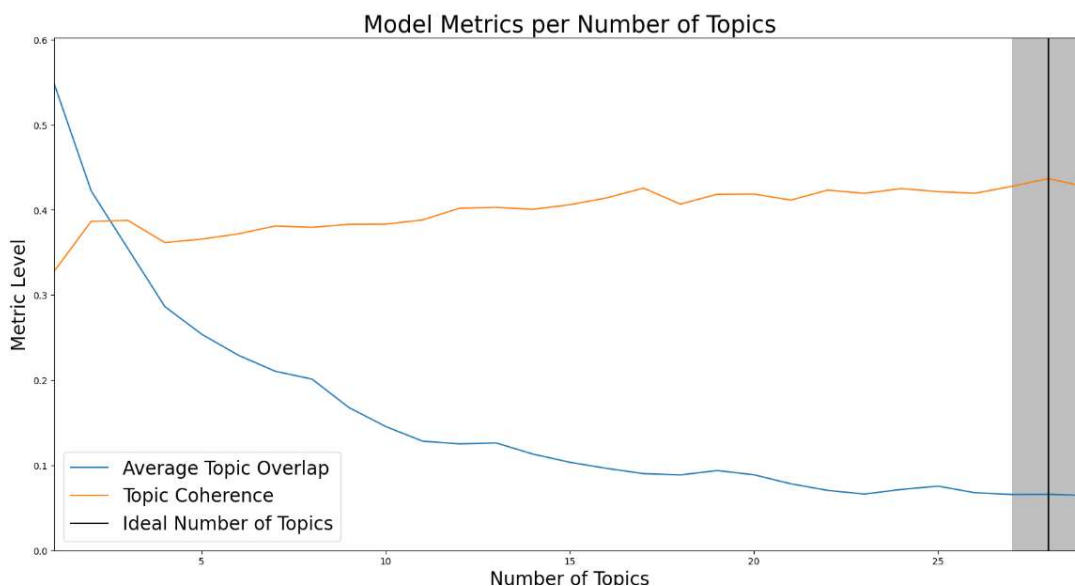
- *'review', 'reviewer'* – ове речи се неће користити у анализи јер је јасно да ако се анализа спроводи над скупом података о рецензијама да се заправо и ради о рецензијама и да није потребно да се то додатно напомиње у темама које ће бити генерисане;
- *'minute', 'min', 'pm', 'hour', 'night', 'day'* – како у анализу нису увршћени бројеви, а временске одреднице најчешће говоре о одређеном броју сати, минута, итд. без бројева уз њих, не могу да буду од користи приликом генерисања тема;
- *'location', 'place'* – рецензије говоре о месту које су корисници апликације посетили, спомињањем речи локација и место, не добијају се корисне информације за боље одређивање тема;
- *'star'* – реч говори о квалитету хотела израженом бројевима од један до пет; како бројеве нису увршћени у анализу, ова реч не носи корисне информације;
- *'stay', 'give', 'pleased', 'say', 'show', 'none', 'part', 'make', 'maker', 'take', 'get', 'tell', 'ask', 'include', 'know', 'use', 'self', 'think', 'thank', 'stayed', 'need', 'try', 'thing',*

'nothing', 'add', 'non', 'left', 'send', 'time' – још неке речи, већином глаголи који се јако често употребљавају али не садрже корисне информације за проблем који ми решавамо.

Још једна битна измена која је направљена је уклањање свих елемената који садрже само један или два карактера. Они су настали у процесу токенизације када су речи са апострофом раздвојене на два дела.

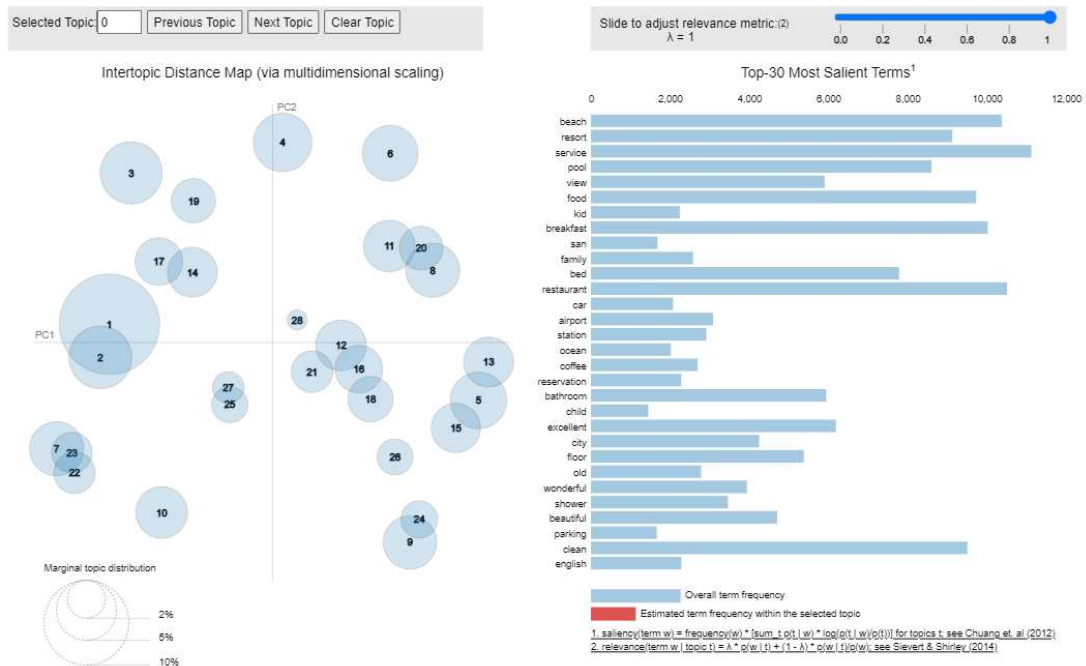
Овако добијен речник сада има 996 елемената и са њим поново се прави корпус, а затим и модел.

У другом покушају је мери кохерентности 'c\_v' придружена и Џакардова сличност и са истим подешавањима параметара (*passes* = 20, *iterations* = 100 и *random\_state* = 100) као у првом покушају направљени су модели за теме у опсегу од један до 30. Графичким приказом, приказаним на слици испод, представљен је однос ове две мере за све направљене моделе и модел који је имао највећу разлику између њих је издвојен као оптималан.



**Слика 20:** Графички приказ одређивања оптималног броја тема помоћу мере кохерентности и Џакардијеве сличности

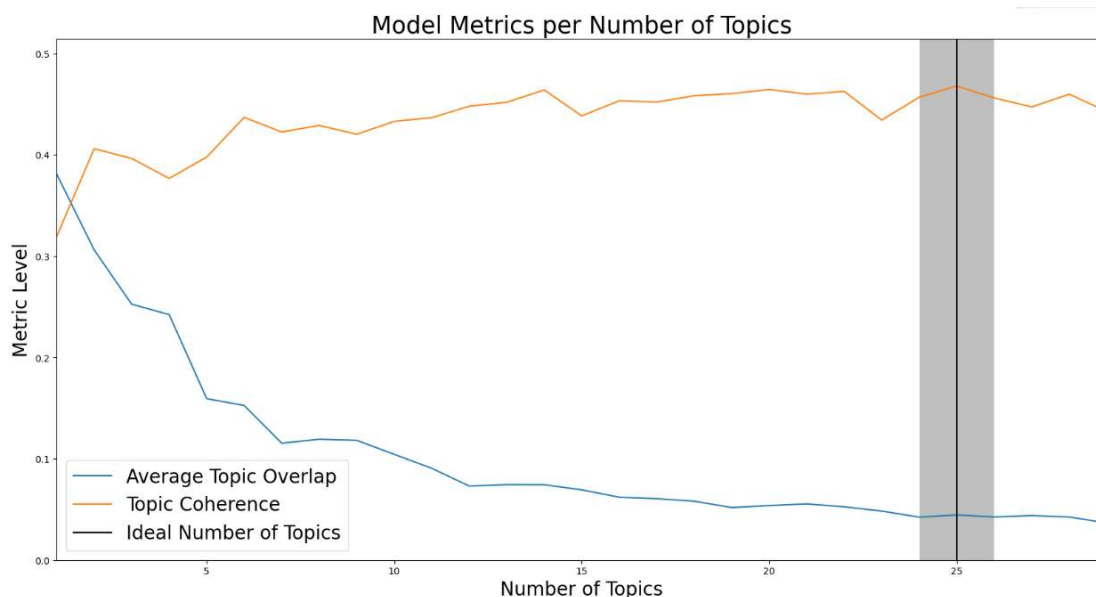
Оптималан број тема је 28 и његова визуелизација је приказана у наставку:



Слика 21: Визуелни приказ модела

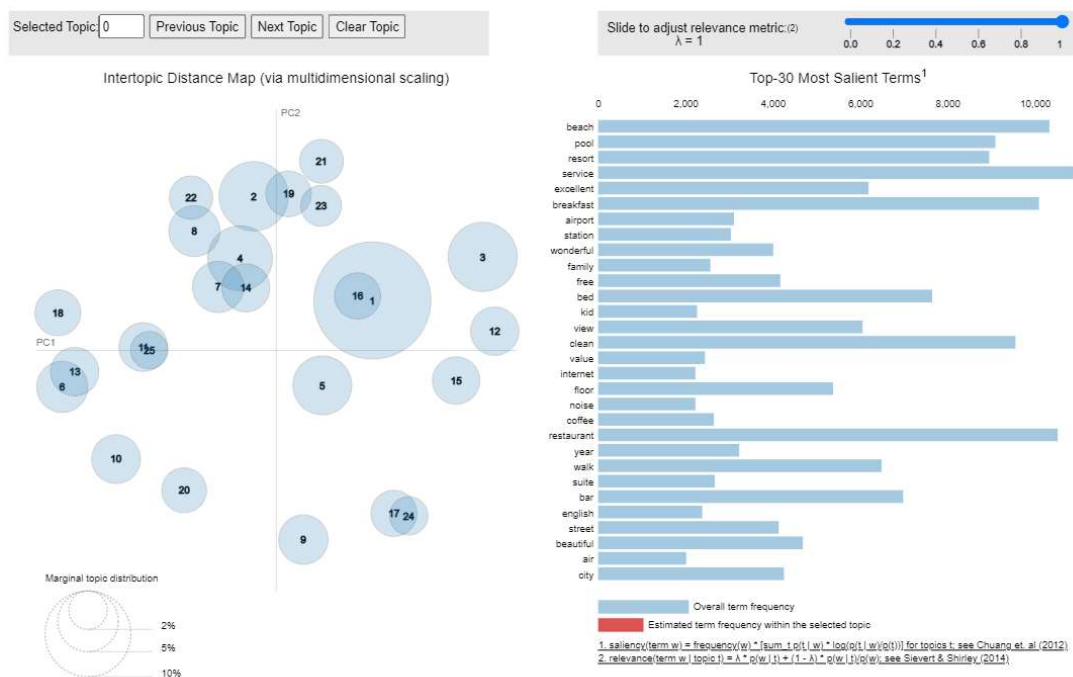
Како је 28 приказано као оптималан број тема, ради провере да ли ће се добити бољи резултати, измењени су параметри *passes* и *iterations* и додат је параметар *chunksize*. Параметар *random\_state* је остао непромењен. Од свих покушаја, најбољи резултати су добијени када су вредности биле подешене на следећи начин: *passes* = 20, *iterations* = 100 и *chunksize* = 700. Унапређење је посматрано уз помоћу кохерентности. Модел са слике 20 има кохерентност 0,432, док модел са овим параметрима има кохерентност 0,462.

Како су се на овом моделу нова подешавања параметара показала као боља, тестирано је да ће то важити и на моделима са неким другим бројем тема, па је на исти начин, али са новим поставкама параметара поново приступљено одређивању оптималног броја тема. Да би се то постигло поново су коришћене кохерентност и Џакардова сличност.



**Слика 22:** Графички приказ одређивања оптималног броја тема помоћу мере кохерентности и Џакардијеве сличности након поновног подешавања параметара

Са графичког приказа изнад види се да је сада оптималан број тема 25 и визуелизација таквог модела је приказана на слици испод.



**Слика 23:** Визуелни приказ модела након поновног одређивања оптималног броја тема

Сада када је пронађен оптималан број тема на два начина, остаје да се донесе одлука који ће се модел користити, о чему ће се детаљније говорити у следећем поглављу.

## 5. ДИСКУСИЈА РЕЗУЛТАТА

Након што су у претходна два поглавља објашњени сви концепти коришћени за израду модела, начин на који је припремљен скуп података „*Trip Advisor Hotel Review*“ за израду модела, па затим и пронађен оптималан број тема и направљен модел, сада долази на ред анализа добијених резултата. Потребно је да се прикаже каква је расподела тема међу документима, која тема је најзаступљенија у скупу података, као и да о чему говори свака од генерисаних тема. Такође, може да буде приказано и које су се речи издвојиле као доминантне по темама.

Како су добијена два модела, потребно је да се донесе одлука који модел ће се користити у наставку. Ради лакшег доношења одлуке, у наставку су табеларно приказане разлике у добијеним мерама сличности ова два модела.

	'c_v'	перплексност	Џакардова сличност
Модел са 28 тема	0,4321	-6,3573	0,0445
Модел са 25 тема	0,4569	-6,3456	0,0486

Из табеле се види да нема великих варијација у резултатима између модела са 25 и 28 тема, али опет модел са 25 тема има боље резултате за кохерентност и Џакадову сличност, док модел са 28 тема има бољу перплексност. Како су варијације у мерама сличности минималне, одлука може да се донесе сагледавањем визуелног приказа.

Иако ниједан од ова два модела нема „идеалну“ расподелу и има доста преклапања, ако се погледају визуелизације модела са другим бројем тема, на слици испод, може да се уочи се да су модели са слика 20 и 22 далеко бољи.



Слика 24: Примери лоших модела коришћењем визуелизације

Сада, ако се упореде визуелизације модела са 28 тема и модела са 25 тема, приказаних на сликама 20 и 22, може да се виде да је мање преклапања и бољи распоред добијен ако се користи модел са 28 тема па се он проглашава оптималним моделом и користи у наставку рада.

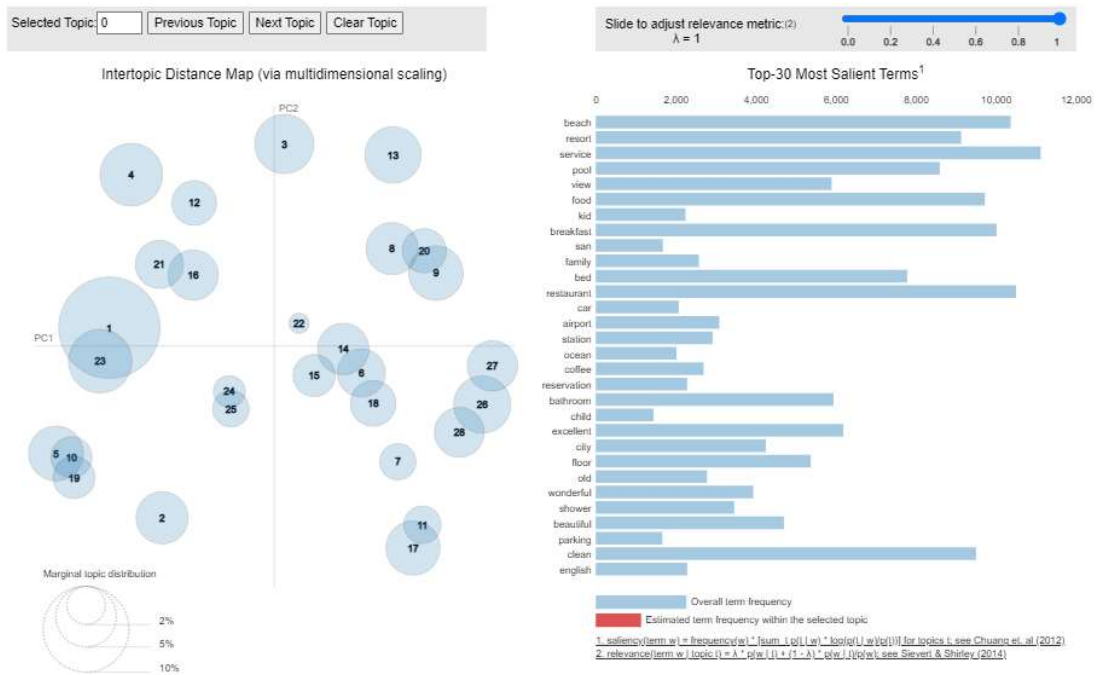
У наставку ће бити приказане све теме са десет доминантних речи које им припадају и вероватноћом појављивања сваке од тих речи у датој теми.

**Табела 4:** Табеларни приказ првих десет најдоминантнијих речи за сваку тему

ТЕМА	ЛИСТА РЕЧИ
<i>Topic 1</i>	0.038*"beach" + 0.026*"resort" + 0.023*"pool" + 0.022*"food" + 0.018*"bar" + 0.017*"restaurant" + 0.017*"drink" + 0.016*"water" + 0.015*"buffet" + 0.010*"lot"
<i>Topic 2</i>	0.035*"wonderful" + 0.024*"home" + 0.023*"trip" + 0.022*"perfect" + 0.022*"honeymoon" + 0.021*"feel" + 0.021*"experience" + 0.017*"service" + 0.016*"way" + 0.016*"husband"
<i>Topic 3</i>	0.024*"small" + 0.020*"bed" + 0.020*"bathroom" + 0.019*"door" + 0.017*"floor" + 0.015*"noise" + 0.013*"shower" + 0.013*"sleep" + 0.012*"old" + 0.012*"little"
<i>Topic 4</i>	0.038*"told" + 0.035*"desk" + 0.024*"check" + 0.020*"card" + 0.017*"problem" + 0.017*"phone" + 0.015*"manager" + 0.012*"key" + 0.012*"service" + 0.012*"door"
<i>Topic 5</i>	0.039*"food" + 0.038*"resort" + 0.029*"beach" + 0.024*"punta" + 0.024*"cana" + 0.023*"holiday" + 0.021*"week" + 0.018*"trip" + 0.017*"vacation" + 0.011*"year"
<i>Topic 6</i>	0.059*"view" + 0.043*"suite" + 0.034*"floor" + 0.018*"harbour" + 0.017*"service" + 0.016*"hong" + 0.016*"kong" + 0.016*"lounge" + 0.015*"singapore" + 0.015*"sydney"
<i>Topic 7</i>	0.097*"san" + 0.049*"juan" + 0.036*"hilton" + 0.033*"old" + 0.032*"francisco" + 0.024*"square" + 0.023*"union" + 0.022*"puerto" + 0.019*"restaurant" + 0.018*"rico"
<i>Topic 8</i>	0.041*"parking" + 0.020*"free" + 0.020*"street" + 0.019*"noise" + 0.018*"floor" + 0.017*"smoking" + 0.017*"seattle" + 0.016*"price" + 0.016*"downtown" + 0.016*"lot"
<i>Topic 9</i>	0.028*"bed" + 0.026*"area" + 0.025*"large" + 0.021*"free" + 0.020*"bathroom" + 0.016*"small" + 0.015*"suite" + 0.015*"internet" + 0.015*"restaurant" + 0.012*"breakfast"
<i>Topic 10</i>	0.098*"kid" + 0.078*"family" + 0.058*"child" + 0.044*"year" + 0.040*"club" + 0.038*"pool" + 0.037*"old" + 0.032*"daughter" + 0.029*"son" + 0.020*"adult"
<i>Topic 11</i>	0.076*"car" + 0.053*"inn" + 0.029*"street" + 0.024*"quarter" + 0.024*"clean" + 0.024*"french" + 0.023*"new" + 0.021*"cable" + 0.020*"holiday" + 0.018*"block"
<i>Topic 12</i>	0.045*"airport" + 0.028*"morning" + 0.026*"coffee" + 0.025*"breakfast" + 0.022*"shuttle" + 0.021*"wine" + 0.019*"check" + 0.015*"flight" + 0.014*"bottle" + 0.013*"free"
<i>Topic 13</i>	0.051*"bed" + 0.043*"bathroom" + 0.041*"shower" + 0.027*"air" + 0.026*"floor" + 0.025*"window" + 0.015*"water" + 0.014*"conditioning" + 0.013*"bath" + 0.013*"wall"

<i>Topic 14</i>	0.035*"pool" + 0.032*"area" + 0.031*"value" + 0.027*"price" + 0.024*"money" + 0.022*"bali" + 0.021*"villa" + 0.020*"restaurant" + 0.019*"bar" + 0.017*"spa"
<i>Topic 15</i>	0.033*"bed" + 0.030*"weekend" + 0.026*"birthday" + 0.024*"fantastic" + 0.023*"bit" + 0.017*"site" + 0.015*"small" + 0.015*"little" + 0.015*"bar" + 0.015*"clean"
<i>Topic 16</i>	'0.104*"service" + 0.030*"business" + 0.021*"guest" + 0.020*"property" + 0.020*"check" + 0.019*"royal" + 0.015*"concierge" + 0.014*"westin" + 0.013*"customer" + 0.013*"desk"
<i>Topic 17</i>	0.059*"excellent" + 0.036*"helpful" + 0.033*"breakfast" + 0.032*"clean" + 0.027*"florence" + 0.025*"wonderful" + 0.024*"recommend" + 0.020*"comfortable" + 0.019*"city" + 0.017*"new"
<i>Topic 18</i>	0.043*"taxi" + 0.040*"paris" + 0.036*"english" + 0.030*"airport" + 0.018*"luggage" + 0.017*"metro" + 0.017*"clean" + 0.016*"walk" + 0.014*"speak" + 0.014*"trip"
<i>Topic 19</i>	0.038*"beautiful" + 0.036*"majestic" + 0.025*"fun" + 0.024*"resort" + 0.022*"pool" + 0.022*"wedding" + 0.022*"wonderful" + 0.020*"food" + 0.019*"awesome" + 0.018*"friend"
<i>Topic 20</i>	0.083*"breakfast" + 0.045*"coffee" + 0.041*"fruit" + 0.033*"fresh" + 0.027*"egg" + 0.026*"juice" + 0.022*"cheese" + 0.021*"bread" + 0.020*"tea" + 0.019*"cereal"
<i>Topic 21</i>	0.081*"service" + 0.069*"restaurant" + 0.061*"food" + 0.028*"bad" + 0.020*"poor" + 0.017*"dinner" + 0.017*"experience" + 0.016*"breakfast" + 0.013*"bar" + 0.013*"overall"
<i>Topic 22</i>	0.161*"palace" + 0.133*"riu" + 0.090*"tokyo" + 0.043*"japanese" + 0.036*"soap" + 0.032*"shampoo" + 0.014*"clean" + 0.014*"machine" + 0.010*"conditioner" + 0.010*"june"
<i>Topic 23</i>	0.063*"resort" + 0.029*"beach" + 0.019*"pool" + 0.016*"trip" + 0.015*"food" + 0.014*"tour" + 0.012*"water" + 0.012*"beautiful" + 0.012*"island" + 0.011*"ride"
<i>Topic 24</i>	0.049*"reservation" + 0.042*"rate" + 0.035*"terrace" + 0.026*"view" + 0.020*"helpful" + 0.020*"balcony" + 0.019*"expedia" + 0.017*"manager" + 0.016*"book" + 0.016*"breakfast"
<i>Topic 25</i>	0.109*"beach" + 0.063*"view" + 0.047*"pool" + 0.047*"ocean" + 0.033*"waikiki" + 0.023*"construction" + 0.019*"balcony" + 0.016*"palm" + 0.015*"area" + 0.014*"tower"
<i>Topic 26</i>	0.049*"station" + 0.034*"walk" + 0.028*"train" + 0.023*"amsterdam" + 0.020*"city" + 0.020*"bus" + 0.020*"breakfast" + 0.019*"clean" + 0.018*"london" + 0.013*"central"
<i>Topic 27</i>	0.038*"modern" + 0.031*"clean" + 0.027*"comfortable" + 0.026*"small" + 0.025*"quiet" + 0.018*"street" + 0.017*"bed" + 0.016*"bathroom" + 0.016*"price" + 0.015*"walk"
<i>Topic 28</i>	0.040*"barcelona" + 0.031*"metro" + 0.030*"city" + 0.028*"euro" + 0.022*"walk" + 0.021*"bar" + 0.019*"breakfast" + 0.019*"plaza" + 0.016*"madrid" + 0.015*"clean"



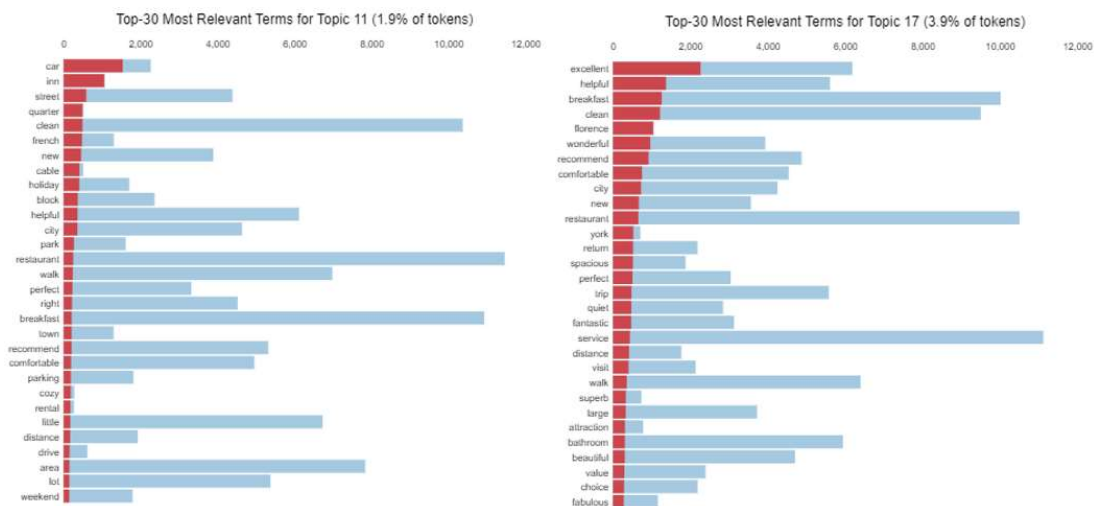


Слика 25: Визуелизација коначног оптималног модела

У наставку уз помоћ слике и табеле могу да се именују све теме. Пре него што се приступи именовању тема битно је напоменути да *pyLDavis* подразумевано сортира теме према проценту учешћа у скупу података. Како би се ово избегло и поклопио редослед са редоследом који је добијен штампањем првих десет речи сваке теме помоћу *gensim* библиотеке, потребно је да се уведе параметар *sort\_topics=False*.

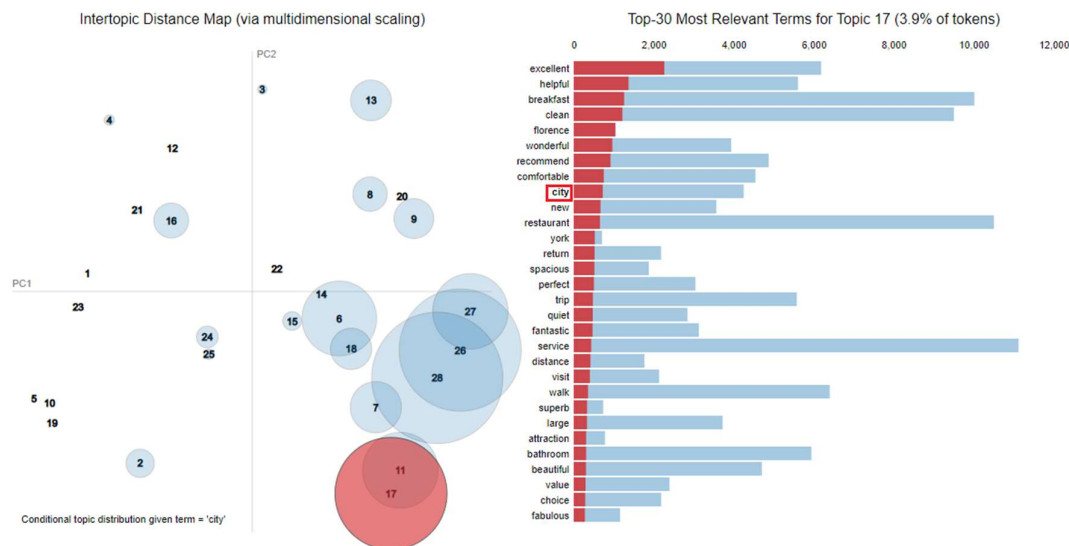
Прво могу да се разматрају теме које се међусобно преклапају, да се види шта им је заједничко и који детаљи их разликују једну од друге.

Прве теме које ће бити анализирани су теме 11 и 17 и ако се обрати пажња на речи из горње табеле, може да се уочи да нема преклапања. Из тог разлога не би било лоше да се у анализу уврсти и листа 30 најрелевантнијих речи за обе теме, које се налазе на десној страни *pyLDavis* визуелизације. Такође, ово ће проширити списак речи који ће помоћи у прецизнијем именовању теме.



Слика 26: Листа доминантних речи за теме број 11 и 17

Како обе теме садрже реч *'city'* али се разликују по местима о којима говоре (у теми број 11 се помињу Фиренца и Њујорк, док се у теми 17 помиње хотел *'Holiday Inn'* и Француска, може да се закључи да се говори о Паризу јер се у њему налази поменути хотел), јасно је да говоре о градским хотелима. Иако стављање речи *'city'* у наслов теме сада изгледа као добра опција, не би било лоше то избећи јер се на слици види да је ова реч доста заступљена и у осталим темама.



Слика 27: Расподела речи *'city'* по темама

Даке, иако је јасно по чему су сличне било би пожељно да се мало дубље дефинишу, па чак и назову по градовима. На пример, тема број 17 је једина тема која у себи садржи реч *'Florence'*, док се у теми број 11 већина речи односи на погодност саобраћаја и простора за шетње. Овом анализом је прикупљено довољно информација да се именују ове две теме:

- Тема број 11: „Suitable environment“

- Тема број 17: „Florence & Comfortable“

Сада ако се обрати пажња на теме број 1 и 23, са слике је могуће видети да обухватају највећи део појављивања речи *'resort'* и *'beach'*. Тема број 1 ставља акценат на оброцима (*'lunch'*, *'dinner'*, *'restaurant'*, *'drink'*, *'bar'*, ... ), док тема број 23 говори о скупљим хотелима (*'excellence'*, *'golf'*, ... ). Именовање је извршено на следећи начин:

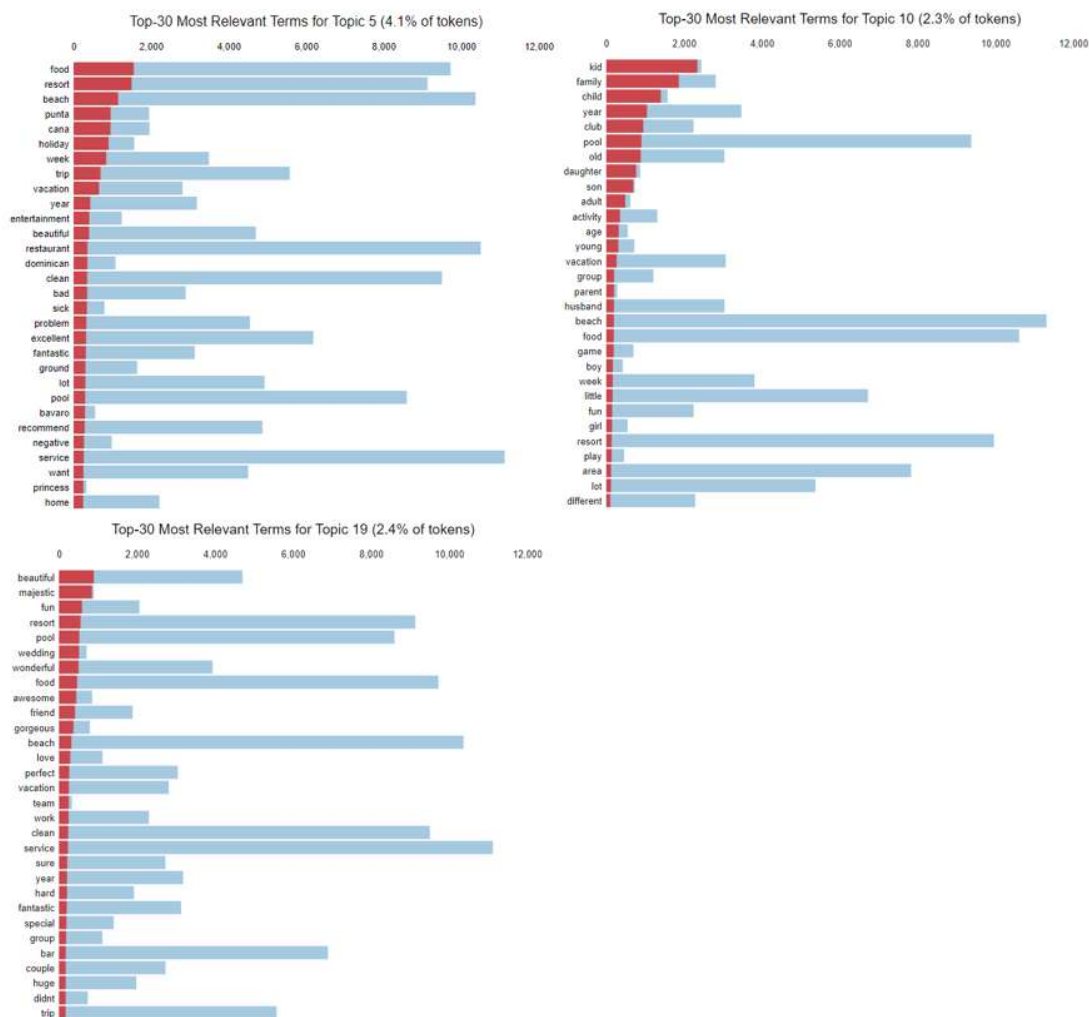
- Тема број 1: „Resort & Beach(Pool) & Meal“
- Тема број 23: „Resort & Beach(Pool) & Excellence“

Како постоје још две групе од по две теме (16 и 21, 24 и 25), имена ће им бити додељена спровођењем анализе на сличан начин.

Може да се напомене да теме, иако имају преклапања, не мора да значи да говоре о сличном искуству. На пример, теме 16 и 21 говоре о особљу и услугама које су им пружили, али се у теми 21 о томе говори у негативном контексту, што није случај са темом број 16.

Ако се погледају теме број 5, 10 и 19, на слици, види се да и оне међусобно деле речи као што су *'Resort'*, *'Beach'*, *'Pool'* али оне нису доминантне речи ове три теме, самим тим рецензије код којих је акценат на овим речима ће вероватније припасти темама број 1 и 23. Из овог разлога из сваке од наведене три теме ће бити извучено оно што је карактеристично за њих појединачно и на тај начин ће бити именоване:

- Тема број 5: „Punta Cana & Bad hygienic“
- Тема број 10: „Family vacation“
- Тема број 19: „Wedding & Couple location“



**Слика 28:** Листа најдоминантнијих речи за теме 5, 10 и 19

По истом принципу, посматрањем доминантних речи и контекста који речи чине као целина, додељени су називи и осталим темама. У наставку се налази табеларни приказ свих именованих тема.

**Табела 5: Именоване теме**

ТЕМА	ИМЕНОВАНЕ ТЕМЕ
<i>Topic 1</i>	„Resort & Beach(Pool) & Meal”
<i>Topic 2</i>	„Wonderful & Honeymoon“
<i>Topic 3</i>	„Dirty rooms“
<i>Topic 4</i>	„Rude staff“
<i>Topic 5</i>	„Punta Cana & Bad hygienic“
<i>Topic 6</i>	„Harbour & Hong Kong & Sydney & Singapore“
<i>Topic 7</i>	„San Juan & San Francisco & Puerto Rico“
<i>Topic 8</i>	„Parking & Smoke“
<i>Topic 9</i>	„Apartment“
<i>Topic 10</i>	„Family vacation“
<i>Topic 11</i>	„Suitable environment“
<i>Topic 12</i>	„Relaxing weekend“
<i>Topic 13</i>	„Bathroom & Room“
<i>Topic 14</i>	„Villa & Bali“
<i>Topic 15</i>	„Birthday celebration“
<i>Topic 16</i>	„Business trip“
<i>Topic 17</i>	„Florence & Comfortable“
<i>Topic 18</i>	„Public transport & Paris“
<i>Topic 19</i>	„Wedding & Couple location“
<i>Topic 20</i>	„Breakfast & Buffet table“
<i>Topic 21</i>	„Bad service“
<i>Topic 22</i>	„Palace & Tokyo“
<i>Topic 23</i>	„Resort & Beach(Pool) & Excellence”
<i>Topic 24</i>	„Beautyful balcony view & Romance & Expedia“
<i>Topic 25</i>	„Beautyful balcony view & Waikiki“
<i>Topic 26</i>	„Urban location & Amsterdam & London & Berlin“
<i>Topic 27</i>	„Quiet area & Boston“
<i>Topic 28</i>	„Turst area & Spain“

У табели у наставку ће за сваку тему бити излистан број рецензија код којих та тема има највећи удео.

**Табела 6:** Додељивање сваке рецензије најдоминантнијој теми

ТЕМА	БРОЈ РЕЦЕНЗИЈА
<i>Topic 1</i>	3264
<i>Topic 2</i>	3558
<i>Topic 3</i>	3150
<i>Topic 4</i>	1943
<i>Topic 5</i>	801
<i>Topic 6</i>	1503
<i>Topic 7</i>	689
<i>Topic 8</i>	1068
<i>Topic 9</i>	792
<i>Topic 10</i>	437
<i>Topic 11</i>	378
<i>Topic 12</i>	329
<i>Topic 13</i>	589
<i>Topic 14</i>	393
<i>Topic 15</i>	351
<i>Topic 16</i>	204
<i>Topic 17</i>	495
<i>Topic 18</i>	126
<i>Topic 19</i>	76
<i>Topic 20</i>	59
<i>Topic 21</i>	66
<i>Topic 22</i>	10
<i>Topic 23</i>	8
<i>Topic 24</i>	36
<i>Topic 25</i>	20
<i>Topic 26</i>	78
<i>Topic 27</i>	46
<i>Topic 28</i>	22

У табели изнад свака рецензија је додељена теми којој највероватније припада, али као што је већ истакнуто, то не значи да ту рецензију чини само та једна тема. Сваки документ може да се представи као мешавина више тема са вероватноћама да припада свакој од њих. Да би ово било јасније из скупа података је извучено неколико примера и представљена је њихова расподела по темама.

**Табела 7: Пример расподеле тема по документима**

Рецензија	Вероватноћа припадности темама
Пример 1: 'nice hotel expensive parking got good deal stay hotel anniversary, arrived late evening took advice previous reviews did valet parking, check quick easy, little disappointed non-existent view room room clean nice size, bed comfortable woke stiff neck high pillows, not soundproof like heard music room night morning loud bangs doors opening closing hear people talking hallway, maybe just noisy neighbors, aveda bath products nice, did not goldfish stay nice touch taken advantage staying longer, location great walking distance shopping, overall nice experience having pay 40 parking night, '	8 - 0.675: „Parking & Smoke“ 13 - 0.225: „Bathroom & Room“ 24 - 0.078: „Beautiful balcony view & Romance & Expedia“
Пример 2: 'shame hotel wasnt good restaurant, arrived clift late afternoon struggle luggage 3 bags, reception staff unhelpful uninterested, eventually managed sorted shown room 9th floor, room suite tried make separate living room putting curtain inbetween bedroom living room, bathroom tiny dirty, stayed mum unfortunatley night didnt feel suffering bad foot, decided phone reception ask doctor come hotel told ther wasnt local receptionist closest told phone, eventually decided hospital just safe, came hospital evening doormen talking girls outside let, following night ate hotel restaurant aisa cuba fantastic, think hotel intrest restaurant bar, end day sleeping ignored wouldnt stay, '	4 - 0.318: „Rude staff“ 3 - 0.204: „Dirty rooms“ 9 - 0.176: „Apartment“ 6 - 0.165: „Harbour & Hong Kong & Sydney & Singapore“ 21 - 0.115: „Bad service“
Пример 3: 'basking barcelona booked short trip barcelona easyjet.great package price included stay petit palace barcelona, hotel just minutes paseo gracia main streets city.the hotel ultra modern minimalistic stylish, rooms good sized room set used board room, staff friendly helpful, definitely stay again.try noti restaurant just minutes walk away, great food lovely atmosphere.by way temperatures weekend january 20 degrees c, warm lunch beach, '	28 - 0.609: „Turst area & Spain“ 1 - 0.213: „Resort & Beach (Pool) & Meal“ 12 - 0.1002: „Relaxing weekend“ 22 - 0.046: „Palace & Tokyo“
Пример 4: 'looks no brains stayed clift twice times experience mixed, hotel looks great clean cool lines high ceilings big windows comfortable beds, standard room pretty small, stayed corner room suit nights room glamour filled light good views, service poor spotty indifferent, staff desk overburdened phone calls handling guests harried rushed not warm, dollars night bedroom suit expect kind service, ritz-carlton covered chintz great service absense good service appreciate staff warm exhibit modicum charm beautiful hotel.i say best way hotel like beautiful girl party nothing say, '	13 - 0.324: „Bathroom & Room“ 16 - 0.221: „Business trip“ 4 - 0.1897: „Rude staff“ 21 - 0.166: „Bad service“ 27 - 0.081: „Quiet area & Boston“

Као што се у теми број 17 појавио град Фиренца као реч која је једна од најдоминантнијих речи ове теме и реч која се само у овој теми и појављује, исти случај се „провлачио“ кроз још неколико тема са називима разних туристичких локацијама широм света. Приликом именовања тема, значај ових речи за те теме није могао да буде занемарен иако постоји шанса да се ова тема појави у склопу рецензије хотела који се не налази у овом граду. Са друге стране, имена ових локација могу да буду од користи у случају да неко жели да борави баш на некој од тих локација јер избор претраге на основу те теме може знатно да убрза претрагу хотела. Како би се направио „компромис“, ове теме поред имена градова садрже још неке елементе који ће додатно описивати сваку од тих тема.

Након што је направљен модел и након што су именоване теме и показано неколико примера требало би се осврнути на примену модела. Примена ЛДА модела и корист коју он доноси у овом експерименту може да се сагледа из два угла. Први говори о користи из перспективе корисника апликације који користи модел да лакше донесе одлуку о хотелу у којем ће одсести. Други се односи на примену од стране менаџмента хотела како би лакше пронашли мане и врлине свог објекта.

ЛДА модел помаже да се уштеди доста времена јер генерише теме за сваку рецензију и тиме пружа могућност да се разуме општа тема одређене рецензије иако њен садржај није прочитан. Потребно је замислити ситуација у којој корисник у току истраживања хотела у којем жели да одседне наиђе на рецензију из табеле означену као „Пример 1“. Уместо да прочита цео текст како би знао о чему се ради довољно је да погледа називе тема које је дефинишу и да добије општу слику о томе шта је написано у рецензији. Наравно, ако се ради о једној или неколико рецензија, никакав проблем не представља читање сваке појединачно, али ако је у питању већи број њих, корисник *Tripadvisor* апликације или било које платформе где се остављају рецензије ће вероватно одустати након свега пар минута читања. Зато коришћење овог модела нуди корисницима апликације да за јако мало утрошеног времена виде главне карактеристике сваког хотела на основу искуства других корисника. Такође, у случају да корисник има унапред дефинисану идеју шта тражи од хотела, може да добије предлоге избором теме која га занима.

Други начин за употребу модела се односи на корист коју сам хотел има од апликације. Прегледом тема које се појављују у рецензијама, рангирањем тих тема по учесталости појављивања, на једноставнији и бржи начин менаџмент хотела може да добије преглед о утисцима који је хотел оставио на госте који су у њему боравили. На тај начин, могу детаљније да прочитају рецензије које садрже лоше теме како би видели шта то тачно изазива незадовољство код корисника. Ако се погледа „Пример 2“ из табеле, види се да ту рецензију карактерише непријатно особље и лоша услуга и да је то оно чему би требало посветити посебну пажњу како би се избегло незадовољство гостију хотела.

Уз помоћ истог примера, могуће је сагледати и предност тога што је свака рецензија представљена као мешавина различитих тема. Када би опис целе рецензије била тема број 9 – „*Apartment*“ то можда не би било довољно да се донесе закључак о искуству особе која је оставила рецензију, али када јој се додају и теме „*Rude staff*“, „*Dirty*



*rooms*“, „*Bad service*“, јасно је да се у овом случају исказује незадовољство и изгледом апартмана и да би било пожељно порадити на његовом ентеријеру.

Са друге стране, у случају да су теме које се помињу позитивне, то може да наведе на размишљање да ако је тај гост приметио и ценио то довољно да о томе остави рецензију, вероватно ће и остали гости пронаћи задовољство у тим стварима и не би било лоше да се те похвале и елементи тих рецензија користе у виду маркетиншке кампање како би хотел привукао још гостију.

## 6. ЗАКЉУЧАК

Главни циљ овог рада јесте упознавање са моделовањем тема са акцентом на латентну Дирихлеову алокацију и применом тема које су добијене као излаз модела. Полазећи од најпростијих облика моделовања тема, њихово унапређивање и додатно разрађивање је временом довело до ЛДА модела.

Иако је практична примена модела, најбољи начин за приказивање и разумевање овог модела пре тога је било неопходно да се истраживањем оригиналног документа о ЛДА моделу и осталих доступних литература открију основни концепти и начин функционисања овог модела.

Након теоријских основа приказани су кораци које је потребно спровести и који резултирају ЛДА моделом. За почетак је потребно упознавање са скупом података и разумевање проблема. Скуп података који је коришћен у раду је „*Trip Advisor Hotel Review*“ и обухвата 20.491 опсервација над којима је извршена анализа у току спроведеног истраживања. Следећи корак је била припрема података која је укључивала сређивање података за процес токенизације који ја затим послужио за добијање речника. Затим је речник филтриран и ослобођен још неких додатних речи које се нису показале као корисне за истраживање. Речник је коришћен за прављење корпуса података који је улаз у ЛДА модел.

Испробавањем различитих подешавања параметара и различитих начина за проналажење оптималног броја тема направљен је модел који садржи 28 различитих тема и свака од њих је представљена као листа речи са вероватноћама да припадају одређеној теми. Када се добила листа тема, како би модел имао сврху у даљој примени свака тема је именована. Именовање тема је извршено на основу анализе речи које се налазе у датој теми и проналажења заједничког контекста за те речи.

На крају, као најбитнији део и суштина спровођења овог истраживања, дати су начини примене ових модела са стране корисника апликације и менаџмента хотела. ЛДА модел се показао јако корисним у ситуацијама када је потребна анализа неструктурираних података.

Како не постоји савршен модел, већ увек постоји додатни простор да се сваки модел унапреди и да се испроба неки нови концепт или унапреди постојећи, у наставку су излистани неки од начина за унапређење дефинисаног ЛДА модела:

- Прављење новог модела који ће користити и колону „*Rating*“ јер рецензија анализирана са оценом коју су дали корисници хотела може да резултира прецизнијим темама;
- Како речник садржи доста речи, од користи може да буде још детаљнија анализа речника да би се проверило да нека реч која не носи корисне информације није „залутала“;
- Коришћење *Rank-biased overlap (RBO)* мере сличности која је настала као унапређење Џакардове сличности;
- Испробавање различитих комбинација *POS* тагова;

- Свакој теми може да се придружи вербални опис који ће ближе одређивати тему и тиме ће постојати опција дубљег разумевања суштине теме;
- У овом раду су коришћени само униграми, самим тим увођење биграма, триграма или неког другог н-грама може да резултује бољим моделом.

## ЛИТЕРАТУРА

- Albanese N. (2022). Topic Modeling with LSA, pLSA, LDA, NFM, BERTopic, Top2Vec: a Comparison [<https://towardsdatascience.com/topic-modeling-with-lsa-plsa-lda-nmf-bertopic-top2vec-a-comparison-5e6ce4b1e4a5#78f7>, датум приступа: 19.10.2022.]
- Blei, D., Ng, A., & Jordan M. (2003). *Latent Dirichlet Allocation*, the Journal of machine Learning research 3, str. 993–1022.
- Deerwester, S., Dumais, S., Landauer, Furnas, T., & Harshman, R. (1990). *Indexing by latent semantic analysis*, br. 6, str. 391–407.
- Delibašić, B., Suknović, M., & Jovanović, M. (2009). *Algoritmi mašinskog učenja za otkrivanje zakonitosti u podacima*. Fakultet organizacionih nauka, Beograd.
- GeeksforGeeks.org (2022) [<https://www.geeksforgeeks.org/what-is-information-retrieval/>, датум приступа: 15.10.2022.]
- Ghanoum T. (2021) Topic Modelling in Python with spaCy and Gensim [<https://towardsdatascience.com/topic-modelling-in-python-with-spacy-and-gensim-dc8f7748bdbf>, датум приступа: 11.8.2022.]
- Lang N. (2022) Stemming vs. Lemmatization in NLP [<https://towardsdatascience.com/stemming-vs-lemmatization-in-nlp-dea008600a0>, датум приступа: 25.10.2022.]
- Nair D. (2016) Text Mining 101: Topic Modeling [<https://www.kdnuggets.com/2016/07/text-mining-101-topic-modeling.html>, датум приступа: 13.10.2022.]
- Malik, U., Goldwasser, M., & Johnston, B. (2020). *Python mašinsko učenje*. Kompjuter Biblioteka, Beograd.
- Mantyla, M., Claes, M., & Farooq, U. (2018). *Measuring LDA Topic Stability from Clusters of Replicated Runs*
- Marasović, A. (2015). *Latentna semantička analiza, varijante i primjene*.
- Pascual, F. (2019) Topic Modeling: An Introduction [<https://monkeylearn.com/blog/introduction-to-topic-modeling/>, датум приступа: 12.10.2022.]
- Pasupat, P. (2021) LSA/PLSA/LDA [<https://ppasupat.github.io/a9online/wtf-is/lsa-plsa-lda.html>, датум приступа: 18.10.2022.]
- Pedro, J. (2022) Understanding Topic Coherence Measures [<https://towardsdatascience.com/understanding-topic-coherence-measures-4aa41339634c>, датум приступа: 27.10.2022.]
- Pythonpot.com (2016) [<https://pythonspot.com/nltk-speech-tagging/>, датум приступа: 15.11.2022.]

Savev, S. (2015) LSI/LSA/SVD – a Bit of History [<https://stefansavev.com/blog/lsi-slash-lsa-slash-svd-a-bit-of-history/>, датум приступа: 14.10.2022.]

Serrano, L. (2020) Latent Dirichlet Allocation [[https://www.youtube.com/watch?v=T05t-SqKArY&ab\\_channel=Serrano.Academy](https://www.youtube.com/watch?v=T05t-SqKArY&ab_channel=Serrano.Academy), датум приступа: 8.11.2022.]

Stodel M. (2020) Using callbacks and logging during training with gensim [<https://www.meganstodel.com/posts/callbacks/>, датум приступа: 27.10.2022.]

Tableau.com (n.d.) [<https://www.tableau.com/learn/articles/natural-language-processing-examples>, датум приступа: 25.10.2022.]

Voita L. (2022) Word Embeddings [[https://lena-voita.github.io/nlp\\_course/word\\_embeddings.html#one\\_hot\\_vectors](https://lena-voita.github.io/nlp_course/word_embeddings.html#one_hot_vectors), датум приступа: 14.10.2022.]