**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**
**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**
**імені ІГОРЯ СІКОРСЬКОГО»**

**Інститут прикладного системного аналізу**

**Кафедра математичних методів системного аналізу**

«На правах рукопису»
УДК ___004.85_____

«До захисту допущено»
Завідувач кафедри
_____ О. Л, Тимощук
«_16»____05_____2018 р.

# Магістерська дисертація

**на здобуття ступеня магістра**

**зі спеціальності 124 Системний аналіз**

**на тему: «Система генерації текстів на основі методів нейронних мереж»**

Виконав:
студент ІІ курсу, групи КА-61м
Лаврій Богдан Петрович _____

Керівник:
доцент кафедри ММСА, к.т.н., доцент
Тимощук О. Л. _____
Рецензент: доцент кафедри програмного забезпечення
комп'ютерних систем ФПМ,
к.т.н., доцент Є.С.Сулема
_____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____

Київ
2018

**Національний технічний університет України**
**«Київський політехнічний інститут імені Ігоря Сікорського»**
**Інститут прикладного системного аналізу**
**Кафедра математичних методів системного аналізу**

Рівень вищої освіти – другий (магістерський)
Спеціальність (спеціалізація) – 124 Системний аналіз (Системний аналіз та управління)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ О.Л.Тимощук
«___»_____20__ р.

**ЗАВДАННЯ**
**на магістерську дисертацію студенту**
**Лаврію Богдану Петровичу**

1. Тема дисертації «Система генерації текстів на основі методів нейронних мереж», науковий керівник дисертації Тимощук Оксана Леонідівна, кандидат технічних наук, доцент кафедри, затверджені наказом по університету від «_27»__березня_ 20_18 р. №_1028-с_
2. Термін подання студентом дисертації 16 травня 2018
3. Об’єкт дослідження: методи обробки природної мови
4. Предмет дослідження: штучні нейронні мережі
5. Перелік завдань, які потрібно розробити: зробити огляд літератури по темі дослідження, дослідити обрані методи на відповідність меті, розробити модифікацію класичного алгоритму, створити систему генерації текстів, проаналізувати роботу системи, розробити стартап-проект за одержаними результатами.
6. Орієнтовний перелік графічного (ілюстративного) матеріалу: презентація, приклади арітектур нейронних мереж, порівняння та аналіз результатів
7. Орієнтовний перелік публікацій: відсутні
8. Дата видачі завдання 16 березня 2018

Календарний план

| № з/п | Назва етапів виконання магістерської дисертації | Термін виконання етапів магістерської дисертації | Примітка |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Студент Б.П. Лаврій

Науковий керівник дисертації О.Л. Тимощук

# РЕФЕРАТ

Магістерська дисертація: 80 с., 11 рис., 22 табл., 2 додатки, 16 джерел.

Актуальність теми – наявність потреби в бізнесу на отримання унікальних машинно генерованих текстів довільного об’єму.

Мета роботи – побудувати модель для генерації текстів, яка продовжує ключову фразу.

Об’єкт дослідження – методи обробки природної мови.

Предмет дослідження – штучні нейронні мережі.

Методи дослідження – моделювання, системний метод.

Наукова новизна одержаних результатів – запропоновано і реалізовано модель, яка використовує паралельні рекурентні комірки та завдяки цьому займає менше місця на носіях інформації.

Практичне значення одержаних результатів – розроблена система задовольняє потребам бізнесу та дозволяє недорого генерувати унікальні тексти; при цьому дозволяє бізнесу економити на носіях інформації для збереження моделей.

ГЕНЕРАЦІЯ ТЕКСТІВ, НЕЙРОННІ МЕРЕЖІ, РЕКУРЕНТНІ МЕРЕЖІ, ОБРОБКА ПРИРОДНОЇ МОВИ, PYTHON, TENSORFLOW.

# ABSTRACT

Masters thesis: 80 p., 11 fig., 22 tabl., 2 appendixes, 16 sources.

The theme: Text generation system using the methods of neural networks.

Actuality – business needs unique machine generated texts of arbitrary size.

Purpose – build the text generating model which extends the key phrase.

The object of study – methods of natural language processing.

Subject of research – artificial neural networks.

The methods of research – modeling, system method.

Scientific novelty of the results – for the first time proposed and implemented the model that uses parallel recursive cells and thus takes up less space on storage.

Practical significance of the results – the developed system satisfies the needs of business and allows to inexpensively generate unique texts while allowing businesses to save on storage.

TEXT GENERATION, NEURAL NETWORKS, RECURRENT NETWORKS, NATURAL LANGUAGE PROCESSING, PYTHON, TENSORFLOW.

# ЗМІСТ

# ПЕРЕЛІК СКОРОЧЕНЬ

RNN - recurrent neural network

CNN - convolutional neural network

ПЗ - програмне забезпечення

ПП - програмний продукт

LSTM - long short term memory

GRU - gated recurrent unit

# ВСТУП

За різними оцінками в сучасному світі понад 75 відсотків даних є неструктурованими. При цьому з розвитком соцмереж та все більшої кількості контенту, який генерують користувачі, доля цих даних може зростати. Тому абсолютно необхідно мати засоби ефективної роботи з ними, бо інакше вони не можуть бути корисними.

Серед інших задач обробки неструктурованих даних таких як розпізнавання об’єктів на фото та відео, сегментація регіонів на фото та відео, класифікація текстів та виявлення спаму або матеріалів непристойного характеру, розпізнавання мови та синтез мови важливою є задача генерації текстів. Вона має велику вартість для бізнесу і хоч розмір відповідного ринку в Україні та на пострадянському просторі порівняно невеликий. Тому розв’язання цієї задачі є актуальним.

З початком бурхливого розвитку глибокого навчання та підвищеним інтересом до нього зі сторони бізнесу з’явилися нові архітектури в нейронних мережах та краще досліджені властивості існуючих. Все більше проблем в цій області стають вирішеними. В той же час деякі речі, про які раніше думали лише як про далекі мрії, стають реальними задачами.

Основною проблемою обробки природної мови є мовна неоднозначність. Існують найрізноманітніші види неоднозначності: синтаксична (структурна), змістова неоднозначність, відмінкова неоднозначність і так далі.

Центральна проблема, як для загальної, так і для прикладної, обробки природної мови – рішення такого роду неоднозначностей – вирішується за допомогою перекладу зовнішнього представлення природної мови в якусь внутрішню структуру.

# РОЗДІЛ 1 МЕТОДИ ОБРОБКИ ПРИРОДНОЇ МОВИ

## 1.1 Особливості роботи з неструктурованими даними

Неструктуровані дані – ті, що або не мають заздалегідь визначеної структури даних, або не організовані в вивзначеному порядку. Неструктуровані дані, зазвичай, представлені у формі тексту, в якому можуть міститись дати, цифри, талиці і факти. Тобто це дані, які не підходять під типову модель реляційних баз даних.

До таких даних можна віднести аудіо, відео, фоторгафії, тексти тощо.

Все частіше відбувається те, що ці дані завантажуються в неструктуровані або, у кращому випадку, вкрай структуровані сховища. Файли веб-логів та інших блоків даних тепер зберігаються у формах, для яких постійно зростаючий набір інструментів полегшує процес парсингу, структурування та аналізу.

Обробка природних мов – це область проектування методів та алгоритмів, які використовують як вхідний матеріал або видають як вихід дані, що не є структурованими, природну мову. Людська мова дуже неоднозначна (наприклад, «Я їв піцу з друзями» та «Я їв піцу з оливками»), а також дуже мінлива (наприклад, «Я їв піцу з друзями» також можна виразити як «Друзі і я розділили піцу»). Вона також постійно змінюється і розвивається. Люди прекрасно володіють мовою та розумінням мови, і здатні висловлювати, сприймати та інтерпретувати дуже складні значення та нюанси. У той же час, хоча ми чудові користувачі мови, ми також некваліфіковані у формальному розумінні та описі правил, які регулюють мову.

Крім проблем, пов'язаних із неоднозначними та мінливими вхіднимим даними в системі з погано визначеним та неуточненим набором правил, природна мова має додатковий набір властивостей, що робить її ще більш

складною для обчислювальних підходів, включаючи машинне навчання: вона дискретна, композиційна та розріджена.

Мова символьна і дискретна. Основними елементами письмової мови є символи. Символи складають слова, які в свою чергу позначають об'єкти, поняття, події та ідеї. Як букви, так і слова є дискретними символами: такі слова, як "гамбургер" або "піца" викликають у нас певні уявлення, але вони також є різними символами, значення яких є зовнішнім для них і залишене для інтерпретації в наших головах. Немає відношення між "гамбургер" і "піца", що можна вивести з самих символів або з окремих букв, з яких вони складені. Порівняйте це з такими поняттями, як колір або акустичні сигнали: ці поняття є безперервними, що дозволяє, наприклад, перейти від кольорового зображення до чорно-білого використанням простої математичної операції або порівняти два різні кольори за допомогою таких властивостей як відтінок та інтенсивність. Це не можна легко зробити словами – немає простої операції, яка дозволить нам перейти від слова "червоний" до слова "рожевий" без використання великої таблиці пошуку або словника.

Мова також композиційна: літери утворюють слова, а слова формують фрази і речення. Значення фрази може бути більшим, ніж значення окремих слів, які його містять, і підпорядковуватися сукупності складних правил. Для того, щоб тлумачити текст, ми, таким чином, повинні працювати за межами рівня букв і слів, а також розглядати довгі послідовності слів, такі як реченння, або навіть цілі документи.

Комбінація вищезазначених властивостей призводить до розрідженості даних. Спосібів, якими слова (дискретні символи) можна об'єднати, щоб сформувати значення, величезна кількість. Ми ніколи не могли сподіватися перелічити їх усі. Не існує чіткого способу узагальнення від одного речення до іншого, або визначення подібності між реченнями, що не залежить від їхнього значення. [1]

## 1.2 Основні задачі обробки текстів

Завдання першого класу можна умовно назвати синтаксичними; тут задачі, як правило, дуже добре визначені і є завданнями класифікації або завданнями породження дискретних об’єктів, і вирішуються багато з них зараз вже досить непогано, наприклад[1]:

- розмітка по частинам мови (part-of-speech tagging): розмітити в заданому тексті слова за частинами мови (іменник, дієслово, прикметник тощо) і, можливо, за морфологічними ознаками (рід, відмінок тощо);
- морфологічна сегментація (morphological segmentation): розділити слова в заданому тексті на морфеми (суфіксі, закінчення ...);
- інший варіант завдання про морфологію окремих слів - стемінг (stemming), в якому потрібно виділити основи слів, або лемматізація (lemmatization), в якій слово потрібно привести до базової форми (наприклад, форми однини чоловічого роду);
- розпізнавання іменованих сутностей (named entity recognition): знайти в тексті власні імена людей, географічних і інших об'єктів, размітивши їх за типами сутностей (імена, топоніми і т. п.);
- встановлення сенсу слів (word sense disambiguation): вибрати, який з омонімів, який з різних смислів одного і того ж слова використовується в даному уривку тексту;
- синтаксичний парсинг (syntactic parsing): по заданому реченню (і, можливо, його контексту) побудувати синтаксичне дерево;
- встановлення кореференцій (coreference resolution): визначити, до яких об'єктів або інших частин тексту відносяться ті чи інші слова і звороти.

Другий клас – це завдання, які в загальному випадку вимагають розуміння тексту, але за формою все ще є добре визначеними завдання з

правильними відповідями (наприклад, завдання класифікації), для яких легко придумати метрики якості. До таких завдань відносяться, зокрема:

- мовні моделі (language models): по заданому уривку тексту передбачити наступне слово або символ; ця задача дуже важлива, наприклад, для розпізнавання мови;
- інформаційний пошук (information retrieval): по заданому запиту і величезній кількості документів знайти серед них найбільш релевантні даному запиту;
- аналіз тональності (sentiment analysis): визначити по тексту його тональність, тобто позитивне ставлення несе цей текст або негативне; аналіз тональності використовується в онлайн-торгівлі для аналізу відгуків користувачів, в фінансах і трейдингу для аналізу статей в пресі, звітів компаній і тому подібних текстів і т. д .;
- виділення відносин або фактів (relationship extraction, fact extraction): виділити з тексту добре визначені відносини або факти про сутності, які згадуються там; наприклад, хто з ким перебуває в родинних стосунках і т. д .;
- відповіді на питання (question answering): дати відповідь на поставлене запитання; в залежності від постановки це може бути або чиста класифікація (вибір з варіантів відповіді, як в тесті), або класифікація з дуже великим числом класів (відповіді на фактологічні питання на кшталт «хто?» або «в якому році?»), або навіть породження тексту (якщо відповідати на питання потрібно в рамках природного діалогу).

І нарешті, до третього класу віднесемо завдання, в яких потрібно не тільки зрозуміти вже написаний текст, а й породити новий. Тут метрики якості вже не завжди очевидні. До таких завдань відносяться, наприклад [2]:

- породження тексту (text generation);
- автоматичне реферування: по тексту породити його короткий зміст; можна розглядати як задачу класифікації якщо вибирати з тексту готові речення, які відображають його суть, або як задачу породження;

- машинний переклад (machine translation): по тексту на одній мові породити відповідний текст на іншій мові;
- діалогові моделі (dialog and conversational models): підтримати розмову з людиною; перші чат-боти почали з'являтися ще в 1970-і роки і з тих часів набули певного поширення.

## 1.3 Методи, не засновані на використанні нейронних мереж

Для різних задач обробки природної мови використовують різні методи. Наприклад, для розмітки по частинам мови застосовують приховані моделі Маркова. Їх також використовують для розпізнавання іменованих сутностей. Для тієї ж мети використовують метод максимальної ентропії (method of maximum entropy) та умовні випадкові поля (conditional random fields).

Для задачі встановлення сенсу слів використовують різноматні методи машинного навчання з учителем: наївний баєсівський, метод опорних векторів, логістична регресія, метод k найближчих сусідів. Також використовуються для цієї мети так звані semi-supervised methods.

Також для задачі встановлення кореференцій використовують метод максимальної ентропії та метод опорних векторів, дерева рішень тощо.

Але для цікавої нам задачі віддавна використовують інший підхід.

В мовній моделі визначено розподіл ймовірностей послідовностей лексем природної мови. В залежності від виду моделі лексеми можуть бути слово, символ або навіть байт. Лексеми завжди дискретні. В найбільш ранніх успішних мовних моделях використовувалися послідовності лексем фіксованої довжини, що називаються n-грамами.

В моделях на основі n-грамм визначається умовна ймовірність n-ої лексем за умови попередніх n - 1 лексеми. Добуток цих умовних ймовірностей визначає розподіл ймовірностей більш довгих послідовностей[4].

Цей розклад - не що інше, як ланцюжкове правило ймовірностей. Розподіл ймовірностей початкової послідовності P (x 1, ..., x n-1) можна моделювати за допомогою іншої моделі з меншою величиною n.

Навчання n-граммних моделей не викликає труднощів, тому що оцінку максимальної вірогідності можна розрахувати, просто підрахувавши, скільки разів кожна можлива n-грама зустрічається в навчаючому наборі. Моделі на базі n-грамм були основним компонентом статистичного моделювання мов протягом багатьох десятиліть.

Для невеликих значень в n-граммі навіть є спеціальні назви: уніграмма для n = 1, біграма для n = 2 і триграмма для n = 3. Ці назви утворюються з латинського префікса числового та грецького суфікса «грамма», що позначає щось написане.

Зазвичай моделі n-грамм і (n-1)-грамм обучаються одночасно. Це спрощує обчислення.

Для точного відтворення вихід в моделі P n ми повинні опустити останній символ кожної послідовності при навчанні P n-1.

Як приклад демонструємо, як триграммна модель обчислює ймовірність речення «THE DOG RAN AWAY». Перші слова речення не можна обробляти за допомогою формули за умовчанням, що базується на умовній ймовірності, тому що на початку речення ще немає ніякого контексту. Тому спершу використовуються безумовні ймовірності слов. Таким чином, ми обчислюємо P 3 (THE DOG RAN). Попередній же слово можна предсказать стандартно, скориставшись умовним розподілом P (AWAY | DOG RAN).

Фундаментальне обмеження максимальної правдоподібності в n-граммних моделях полягає в тому, що оцінка P n в навчаючій виборці в багатьох випадках близька до нуля, не дивлячись навіть на те, що кортеж (xt-n + 1, ..., xt) може зустрічатися в тестовій вибірці. Це може привести до катастрофічних наслідків двох видів. Якщо P n-1 дорівнює нулю, то відношення не визначено, тому модель взагалі не дає розумної відповіді. Якщо ж P n-1 не дорівнює нулю, але P n дорівнює нулю, то логарифмічна ймовірність рівна -∞.

Щоб уникнути подібних неприємностей, в більшості n-граммних моделей використовується та або інша форма згладжування. Сенс цього прийому полягає в тому, щоб зсунути масу ймовірності від кортежів, що зустрічалися, до тих, які не зустрічалися, але схожі. Одна з основних технік - додати незначну масу ймовірностей для всіх можливих значень символів. Цей метод можна обгрунтувати як байесовський висновок, в якому апріорний розподіл обчислювачів рівномірно або є розподілом Дирихле. Ще одна дуже популярна ідея - утворити змішану модель з n-граммних моделей високого та низького порядків, де моделі високого порядку забезпечують велику ємність, а моделі низького порядку з більшою вірогідністю уникають нульових лічильників. Поворотні методи шукають n-грами низького порядка, якщо частота контексту x t-1, ..., x t-n + 1 занадто мала для використання моделі більш високого порядку. Точніше, вони оцінюють розподіл x t з використанням контекстів x t-n + k, ..., x t-1 для зростаючих k, поки не буде знайдено досить надійну оцінку.

Класичні n-грамні моделі особливо вразливі до прокляття розмірності. Існує | V | n можливих n-грам, і | V | часто дуже велике. Навіть при наявності масивного навчального набору і помірному n більшість n-грам в навчальному наборі не зустрічається. Класичну n-програмних модель можна розглядати як пошук найближчого сусіда, тобто як локальний непараметрический предиктор, схожий на метод k найближчих сусідів. Але в мовній моделі проблема навіть серйозніша, ніж зазвичай, тому що два будь-яких різних слова знаходяться на однаковій відстані одна від одної в просторі унітарних векторів. Тому важко отримати хоч якусь інформацію з «сусідів» - лише навчальні приклади, що буквально повторюють один і той же контекст, корисні для локального узагальнення. Для подолання цих проблем модель мови повинна вміти розділяти знання між одним словом і іншими семантично схожими словами. [4]

## 1.4 Методи, засновані на використанні нейронних мереж

Один з основних інструментів сучасної обробки текстів - розподілені представлення слів (distributed word representations, вони ж word embeddings). У цих представленнях кожному слову ставиться у відповідність вектор з дійсних чисел, елемент евклідового простору R d для деякого d (зазвичай кілька сотень). Ці вектори далі служать входами наступних моделей, і базове припущення полягає в тому, що геометричні співвідношення в просторі R d будуть відповідати семантичним співвідношенням між словами, наприклад, найближчі сусіди слова в цьому просторі виявляться його синонімами або іншими тісно пов'язаними словами, і т. д. [4]

Можна сказати, що розподілені представлення слів ґрунтуються на припущенні, що в лінгвістиці називається дистрибутивною гіпотезою (distributional hypothesis): слова зі схожим змістом будуть зустрічатися в схожих контекстах.

В класичних моделях обробки текстів початковим поданням тексту, входом моделі, зазвичай були слова, закодовані у вигляді так званого one-hot представлення: кожне слово в словнику представляється у вигляді вектора, розмір якого дорівнює числу слів у словнику. При цьому всі елементи вектора, крім одного, дорівнюють нулю, а елемент в позиції, що відповідає номеру слова в словнику, дорівнює одиниці. В результаті в векторі кожного слова одиниця з'являється рівно по одному разу.

Однак у випадку представлення слів природної мови недоліки очевидні:

- по-перше, слів у мові дуже, дуже багато; їх, звичайно, не настільки багато, щоб словник неможливо було скласти, але він може містити сотні тисяч слів, а якщо мова йде про тексти з інтернету, де зустрічаються помилки,

то і мільйони; така велика кількість точок даних - це для сучасного машинного навчання абсолютно нормально, але коли у кожної точки даних розмірність мільйон, це вже занадто;

– по-друге, це кодування передбачає, що кожне слово на вході абсолютно незалежно від інших, і обробляти їх потрібно теж окремо; для слів «комп'ютер» і «рибалка» це припущення можна виправдати, але ми робимо його і для таких, наприклад, слів, як «комп'ютер», «комп'ютерний» і «комп'ютери», а також для слів «червоний», «багряний» і «бордовий» - для моделі, на вхід якої слова подаються в one-hot представленні, всі вони будуть абсолютно незалежні; і це припущення вже здається дивним.

Ми б хотіли, щоб вектори слів в цьому багатовимірному просторі якось відображали семантику самих слів, то, як ці слова поєднуються один з одним в природній мові. Тому даними для навчання може служити просто набір текстів мовою, яка нас цікавить; якщо потрібна більш специфічна модель, з якоїсь тематики чи з якогось конкретного джерела, наприклад соціальної мережі, то краще взяти набір текстів з потрібними властивостями, але ніяк спеціально розмічати їх не обов'язково, даними якраз через те , як слова мови пов'язуються один з одним в реченнях і текстах.

Перші думки в цьому напрямку були пов'язані з аналізом матриці спільної зустріваності слів (cooccurrence matrix). Це величезна матриця, рядками якої є слова, а стовпцями - тексти, в яких ці слова зустрічаються. Під «текстами» тут, в залежності від застосування, може розумітися набір слів дуже різного розміру, від твіти до «Війни і миру», але суть залишається незмінною: ми хочемо якось уявити цю матрицю в компактному, скороченому вигляді. Для того щоб представити рядки і стовпці величезної матриці у вигляді векторів, є стандартний підхід - сингулярне розкладання матриць (singular value decomposition, SVD). Будь-яку матрицю X розміру N x M можна представити у вигляді добутку матриць розміру N x R і R x M, де R - її ранг, а для будь-якого k

її можна найкращим чином наблизити як U - матриця розміру N x k, V - матриця розміру k x M.

Цей розклад можна обчислити досить ефективно навіть для величезних матриць, особливо якщо вони сильно розріджені. А матриця зустріавності слів в документах, звичайно, завжди буде розріджена, адже не може значна частка всіх можливих слів зустрічатися в кожному документі. Після того як ми зробимо такий розклад, на матрицю U можна дивитися як на ті самі вектори ознак довжини k для кожного з N. В цьому полягає основна ідея латентного семантичного аналізу (latent semantic analysis, LSA, він же latent semantic indexing, LSI).

У базового LSI є ряд проблем, в основному пов'язаних з тим, що слова зустрічаються вкрай нерівномірно, і занадто часто зустрічаються слова які погано впливають на матрицю спільної зустріваності. Грубо кажучи, в базовому LSI ви будете класифікувати слова за тим, наскільки часто вони знаходяться поруч зі словом «і» або «а», ніж з якихось змістовними ознаками. Ці проблеми люди вирішують досі, з'являються нові варіанти LSI, які дають більш вдалі представлення слів [2].

Сучасні нейромережеві підходи до представлення слів з’явилися в 2003 році, в роботі Йошуа Бенджі зі співавторами; інакше кажучи, нейромережеві розподілені представлення слів з'явилися ще до сучасної революції глибокого навчання. І народилися вони з завдання побудови мовних моделей, які передбачають наступне слово в тексті.

Попередні роботи по мовним моделям зазвичай були засновані на статистиці n-грам слів [8].

Модель word2vec була запропонована Томашем Міколовим з співавторами, причому відразу в двох варіантах: у вигляді безперервного мішка слів (continuous bag of words, CBOW) і у вигляді архітектури skip-gram. Ідея word2vec полягає в тому, щоб навчитися передбачати слово з його контексту або навпаки. Суть моделі CBOW полягає в тому, щоб навчитися якомога краще вирішувати таке завдання: по заданому контексту слова відновити саме слово;

фактично це пряме завдання побудови мовної моделі. Тільки вікна тепер ми можемо вибирати як захочемо: нам не обов'язково передбачати наступне слово за попередніми, як у мовній моделі, а можна, наприклад, спробувати передбачити центральне слово у вікні по лівому і правому контексту. Саме передбачення робиться моделлю, дуже схожою на нейронну мережу; взагалі, по суті word2vec - це нейронна мережа, але неглибока нейронна мережа, з одним прихованим рівнем.

Модель CBOW робить наступне:

- кожен вхід мережі - це вектор в one-hot представленні розмірності V, де V - розмір словника;
- прихований шар мережі - це фактично і є матриця W векторних представлень слів; n-й рядок W містить представлення n-го слова зі словника;
- при обчисленні виходу прихованого шару ми беремо просто середнє всіх вхідних векторів; така простота моделі важлива для того, щоб в результаті в просторі представлень співвідношення між векторами слів були теж якомога простішими;
- на виході отримуємо якусь оцінку u_j для кожного слова в словнику; потім апостеріорний розподіл моделі обчислюється за допомогою звичайного softmax;
- таким чином, функція втрати на одному вікні полягає в тому, щоб зробити апостеріорне розподіл якомога більше схожим на розподіл даних.

А модель skip-gram працює прямо протилежним чином: тепер ми не передбачаємо слово, усереднюючи контекст контекст, а намагаємося передбачити кожне слово контексту по даному слову. Архітектура відповідної мережі показана нижче на рис. 1.1.

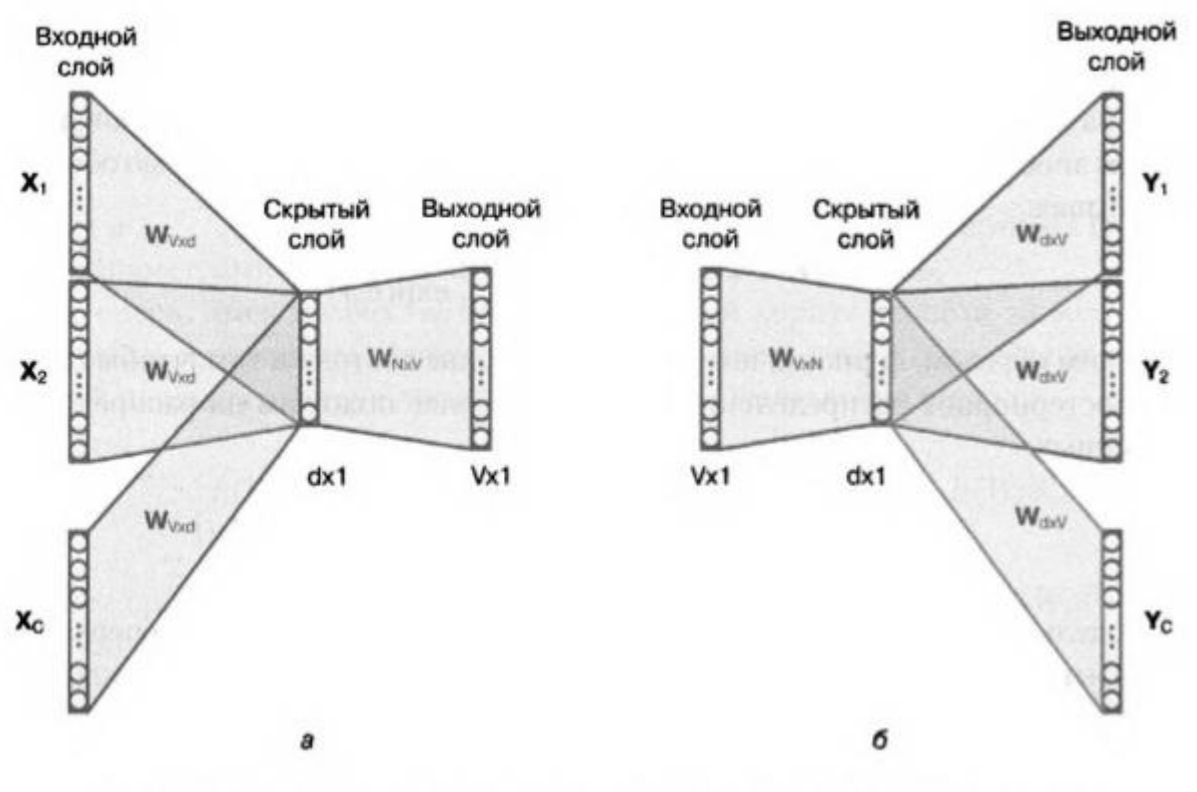



Рисунок 1.1 – Моделі word2vec: а - CBOW, б - skip-gram

Як правило, групуються аналогічно вживані слова розумно разом у векторному просторі. Наприклад, якщо ми використовуємо T-SNE (алгоритм візуалізації зменшення розмірності), щоб згладити розміри наших векторів у 2-мірному просторі та використовувати слова, які ці вектори представляють як їхні мітки, ми можемо побачити щось подібне (рис. 1.2):

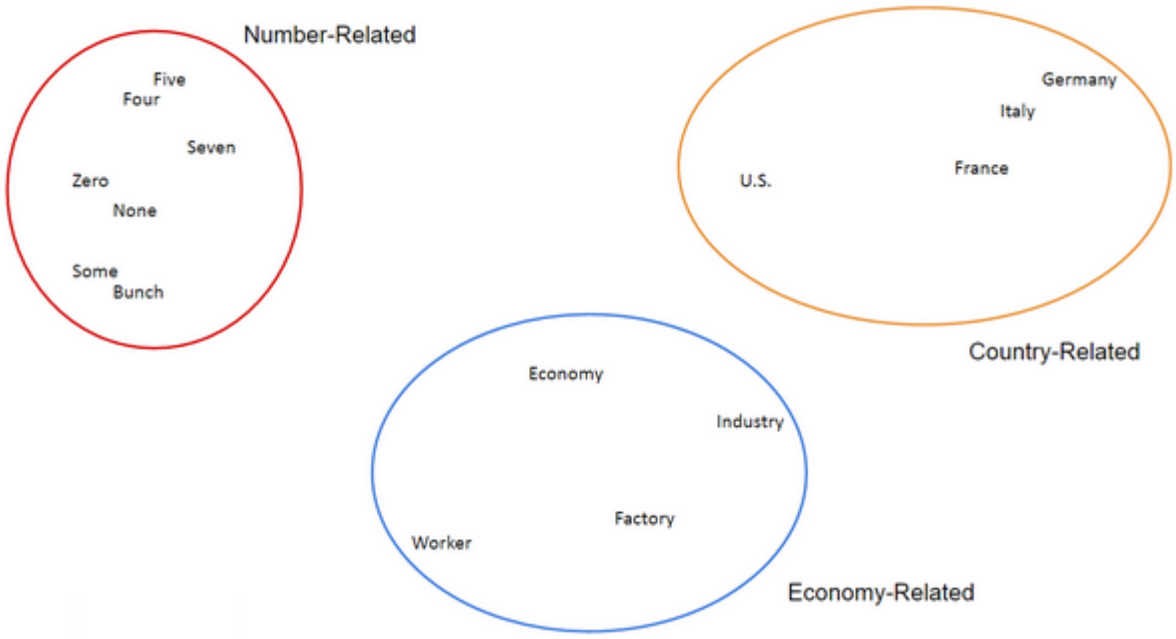



Рисунок 1.2 – Представлення слів

Найпопулярнішою сучасною альтернативою word2vec є моделі GloVe (від слів global vectors). Основна ідея GloVe полягає в тому, щоб поєднувати сильні сторони двох підходів-попередників, одночасно уникаючи їх недоліків[4].

У таких областях як, наприклад, аналіз теми документа та рецензування текстів ще використовують і згорткові нейронні мережі (CNN). Вони допомогають виділяти низькорівневі особливості та сворювати з них складніші характеристики.

## 1.5 Постановка задачі

Постановка задачі:

- проаналізувати методи і підходи об’єкту дослідження;
- зробити висновки щодо підходів до обробки природної мови завлежно від класу задач;
- дослідити обрані методи на предмет адекватності до мети роботи;

- розробити модифікований підхід заснований на рекурентних нейронних мережах;
- реалізувати даний підхід;
- проаналізувати результати підходу та зробити висновки.

## Висновки до розділу

В цьому розділі було розглянуто особливості природної мови та основні задачі обробки природної мови. Проаналізовано різні алгоритми обробки природної мови та охарактеризовано для яких типів задач їх можна використовувати.

На основі цього можна зробити висновки що для задачі моделювання мови (language modeling) нам треба детальніше розглянути різні архітектури нейронних мереж як найбільш передові рішення для цього класу задач.

При цьому було використано системний підхід для декомпозиції задачі та формулювання постановки задачі наведеної нижче.

Також поставлеон задачу на магістерську роботу.

# РОЗДІЛ 2 АНАЛІЗ МЕТОДІВ ОБРАНИХ ДЛЯ ВИРІШЕННЯ ПРОБЛЕМИ

## 2.1 Рекурентні нейронні мережі

Рекурентні нейронні мережі - це моделі глибокого навчання з простими структурами та механізм зворотного зв'язку, вбудований або різними словами, вивід шару доданий до наступного входу і подає назад на той самий шар.
Рекурентна нейронна мережа - це спеціалізований тип нейронної мережі, що вирішує проблему збереження контексту для послідовних даних, наприклад, дані про погоду, запаси, гени тощо. На кожному ітераційному кроці процесор приймає вхід і поточний стан мережі, і виробляє вихідний і новий стан, який повторно подається в мережу. Приклад на рисунку 2.1[4]

Однак ця модель має певні проблеми. Це дуже обчислювально дорого, щоб підтримувати стан для великої кількості одиниць, тим більше - довгого часу. Крім того, рекурентні мережі дуже чутливі до змін їх параметрів. Таким чином, вони схильні до різних проблем зі своїм оптимізатором спуску градієнта - вони або зростають експоненціально (вибуховий градієнт) або опускаються до нуля та стабілізуються (Vanishing Gradient), обидва проблеми, які сильно шкодять можливості навчання моделі.

Щоб вирішити ці проблеми, Hochreiter і Schmidhuber опублікували в 1997 році документ, який описує спосіб збереження інформації протягом тривалого періоду часу і додатково вирішує надмірну чутливість до змін параметрів, тобто робить зворотне поширення через рекурентні мережі більш життєздатними.

Архітектура нейронних мереж:

- повністю рекурентна мережа;
- мережі Хопфілда;
- мережі Елмана та Джордана;
- echo state networks;

– довга модель короткотермінової пам'яті (LSTM).

Модель РНН:

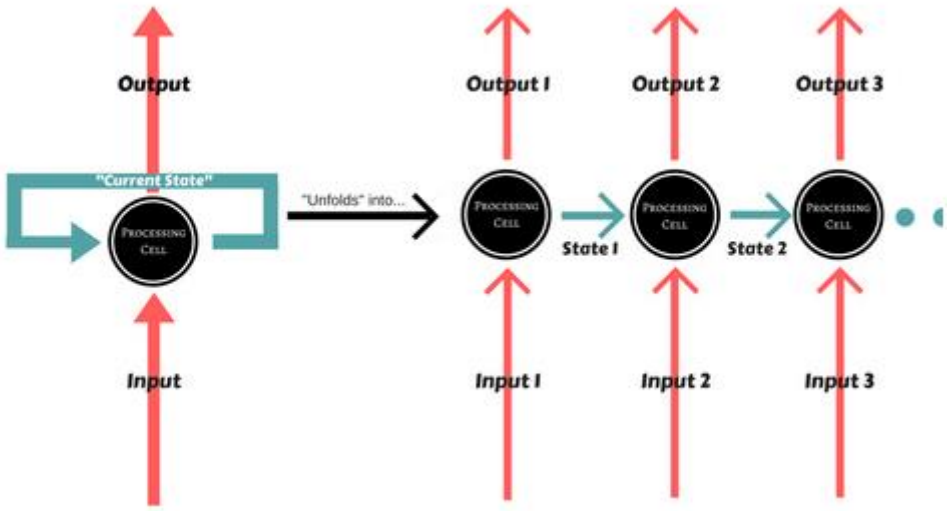



Рисунок 2.1 – Загальна модель RNN

## 2.2 Мережі Хопфілда

Сітка Хопфілда складається з бінарних порогових комірок з рекурентними зв'язками між ними. [19]

Рекурентні мережі з нелінійними комірками, як правило, дуже важко проаналізувати. Вони можуть вести себе по-різному:

- дійти до стабільного стану;

- осцилювати;

- дотримуватись хаотичних траєкторій, які неможливо передбачити далеко в майбутнє.

Але Джон Хопфілд (та інші) зрозумів, що якщо зв'язки симетричні, існує глобальна енергетична функція:

- кожна бінарна "конфігурація" всієї мережі має енергію.

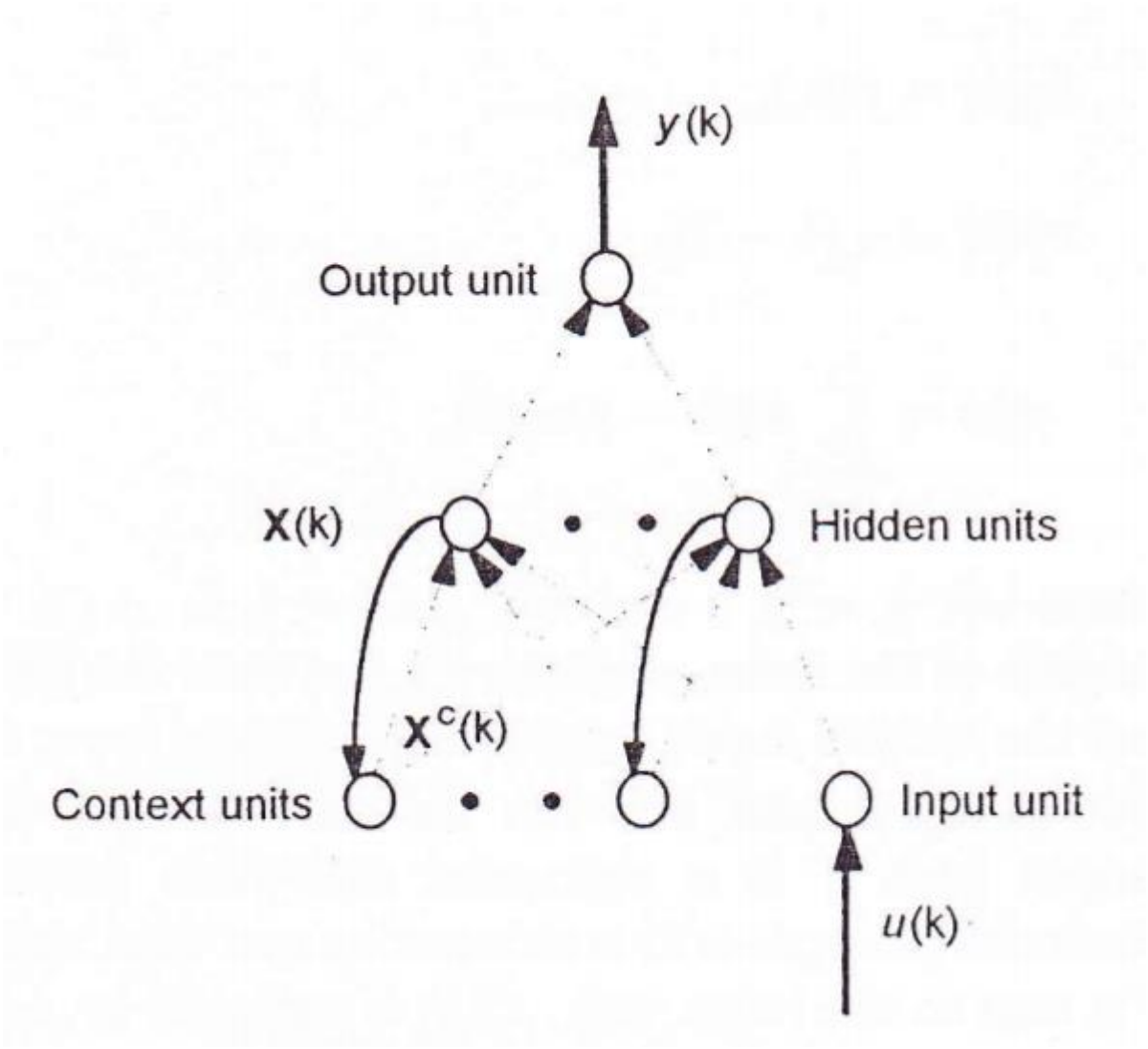
- правило бінарного порогового рішення призводить до встановлення мережі на мінімум цієї енергетичної функції.

Це дозволяє робити мережі з памяттю. Але розмір цієї памяті залежить від кількості нейронів в мережі та такі архітектури слабо підходять для нашої задачі.

## 2.3 Мережі Елмана

Елман (1990) представив просту рекурентну нейронну мережу (у цьому випадку мережа має лише один вхідний блок і однин вихідний). У базовій мережі Елмана, крім вхідного блоку, прихованих шарів та блоку виводу, є також контекстні блоки. Вхідні та вихідні пристрої взаємодіють із зовнішнім середовищем, тоді як приховані та контекстні блоки – ні. Блок введення - це лише буферна одиниця, яка пропускає сигнали, не змінюючи їх. Вихідний блок - це лінійна одиниця, яка накопичує сигнали, що подаються на нього. Приховані одиниці можуть мати лінійні чи неін'єрні функції активації. Контекстні блоки використовуються лише для запам'ятовування попередніх активацій прихованих блоків і можуть розглядатися як lunction як одностадійні тимчасові затримки. Потішні з'єднання (пунктирні лінії) можна модифікувати; повторювані з'єднання (суцільні лінії) фіксуються. Оскільки повторювані з'єднання фіксуються, мережа Елмана іноді викликає частково рекурентну мережу. [18]

Приклад на рисунку 2.2 нижче:




Рисунко 2.2 – Мережа Елмана

Мережі Елмана зараз майже не використовують для моделювання мов та генерації текстів так, як інші методи є кращими за них.

## 2.4 Мережі з LSTM та GRU комірками

Як видно з показаного раніше, довга короткотермінова пам'ять складається з лінійної одиниці, оточеної трьома логістичними воротами[5].

Назва цих воріт варіюється від місця до місця, але найпоширенішими назвами для них є:

– вхід "Вхід" або "Write", який обробляє запис даних в інформаційну комірку,
– "Вихідний" або "Read" ворота, який обробляє передачу даних назад у рекурентну мережу, і
– ворота "Keep" або "Forget", яка обробляє збереження та зміну даних, що зберігаються в інформаційній комірці.

Схема роботи на рисунку 2.2:

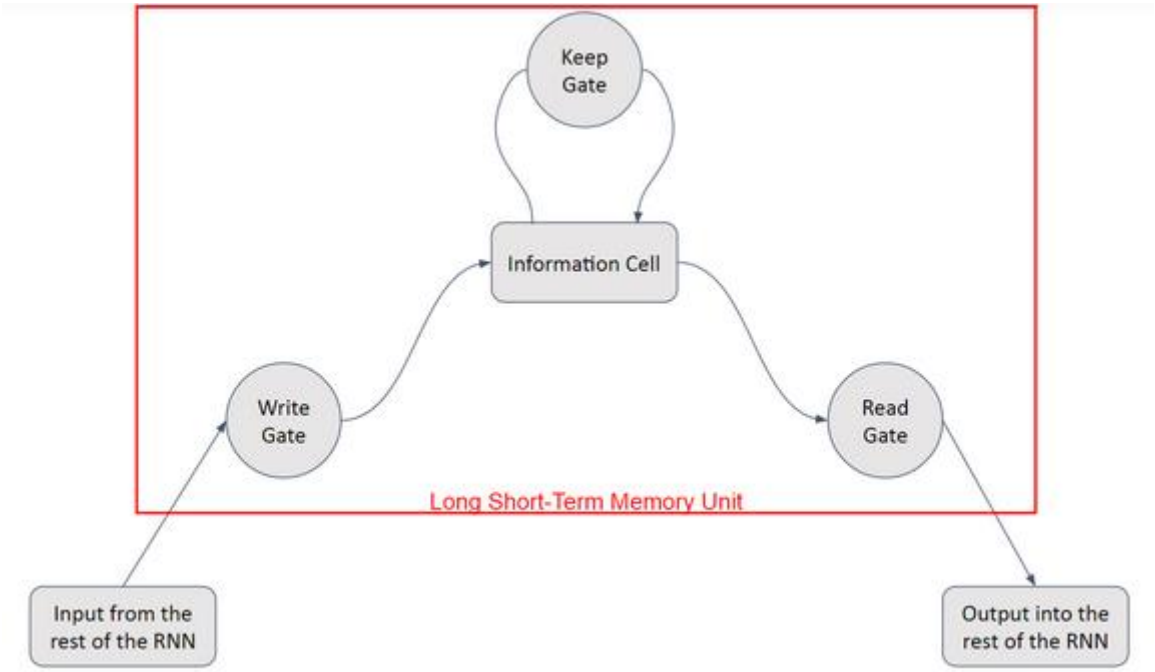



Рисунок 2.2 – Схема LSTM комірки

Гейтінгові рекурентні блоки (GRUs) - це механізм стробування в рекурентних нейронних мережах, введений в 2014 році Кюнхьюном Чо. Їх продуктивність у моделюванні поліфонічної музики та моделювання мовного сигналу виявилася подібною до довготривалої короткочасної пам'яті. [20]

Вони мають менше параметрів, ніж LSTM, оскільки їм не вистачає вихідних воріт. Схема роботи на рисунку 2.3:

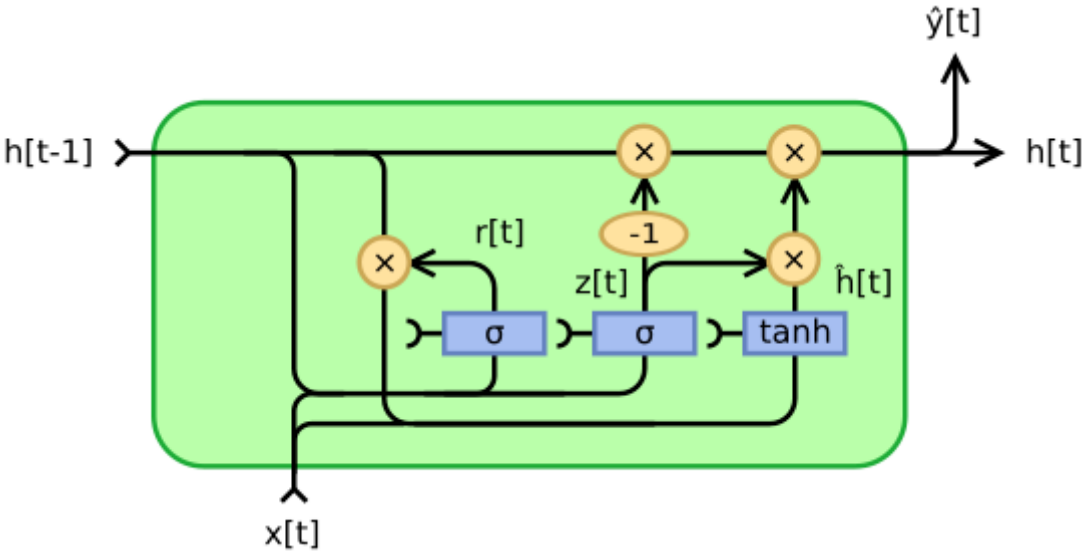



Рисунок 2.3 - GRU

## 2.5 Проблема перенавчання і узагальнююча здатність

У випадку перенавчання (overfitting) статистична модель реагує на випадкову похибку або шум, замість взаємозв'язку, що лежить в основі даних. Перенавчання виникає тоді, коли модель є надлишково ускладненою. Перенавчена модель погана для прогнозування, бо вона занадто сильно реагує на другорядні та неважливі насправді чинники.

Для запобігання перенавчанню розроблено велику кількість різних методів:

- Підвищення кількості даних;
- Структури розподілу ваги (CNN, RNN);
- Поступове зменшення вагів;
- Рання зупинка;
- Усереднення моделей;
- Дропаут;
- Нормалізація батчів;
- Структури регулювання ваги;

• Байєсове пристосування нейронних мереж;

• Генераційна підготовка;

• Розрідженсть в прихованих шарах.

## 2.6 Метрики якості моделей

Є кілька показників для оцінки моделювання мов. Орієнтовані на застосування оцінюють їх у контексті виконання завдання вищого рівня, наприклад, шляхом вимірювання поліпшення якості перекладу при переведенні компонентів мовного моделювання в систему перекладу з моделі A в модель B.

Кращою є оцінка заплутаності або невизначеності (perplexity) на небачених раніше реченнях. Заплутаність - це інформаційне теоретичне вимірювання того, наскільки добре вібраційна модель прогнозує вибірку. Низькі значення свідчать про кращу відповідність. Враховуючи текстовий корпус з слів w1, ..., wn (n може бути в мільйонах) і функцію мовної моделі LM, яка присвоює імовірність слову на основі його історії, заплутаність LM по відношенню до корпусу: 2 1 n Pn i = 1 log2 LM (wi | w1: i1) Хороші мовні моделі (тобто, що відображають реальне використання мови) призначають високі імовірностей для подій у корпусі, що призведе до нижчих значень заплутаності.

## Висновки до розділу

Нейронні мережі сильно розвинулися від часі свого створення в минулому столітті. Нинішні варіанти та архітектури здатні не гірше за людей, а іноді й краще виконувати різні завдання.

В цьому розділі розглянуто різноманітні архітектури нейронних мереж. Серед них рекурсивні нейронні моделі, мережі Хопфілда, мережі з LSTM/GRU комірками. Описано переваги та недоліки різних архітектур.

Для подальшої модифікації обрано рекуренту мережу з LSTM/GRU комірками.

Також розглянуто головну метрику якості, яка буде далі використовуватися в роботі – заплутаність (perplexity). Нижчі її значення означають кращу впевненість в собі мовної моделі.

# РОЗДІЛ 3 ПРОЕКТУВАННЯ АРХІТЕКТУРИ СИСТЕМИ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

## 3.1 Вибір інстуметів та вихідні дані

Більшість сучасних реалізацій нейронних мереж засновані на використанні графічних процесорів (GPU) - спеціалізованого обладнання, яке спочатку розроблялося для графічних додатків. Ринок систем для відеоігор прискорив створення графічних карт. Як з'ясувалося, відеокарти, які необхідні для ігрових систем, підходять і для нейронних мереж.

Для відтворення зображення в відеоіграх необхідно швидко виконувати багато паралельних операцій. Графічна карта повинна паралельно виконувати множення і ділення матриць. Обчислення досить прості і не містять такої великої кількості розгалужень, як у типовій програмі, що виконується на CPU. Наприклад, кожна вершина одного твердого тіла збільшується на одну і ту ж матрицю; не потрібно виконувати в кожній вершині if, щоб зрозуміти, на яку матрицю множити. До того ж обчислення абсолютно незалежні і, отже, легко розпаралелюються. Крім того, при проектуванні графічних карт передбачається висока пропускна здатність пам'яті, а розплачуватися за це доводиться меншою тактовою частотою і не настільки розвиненими засоби розгалуження, як в традиційних процесорах. [3]

TensorFlow - це потужна бібліотека програмного забезпечення з відкритим вихідним кодом для обчислень, що особливо добре підходить для великомасштабного машинного навчання. Його основний принцип простий: спочатку в Python визначте граф обчислень, які потрібно виконати, а потім TensorFlow приймає цей графік і запускає його, використовуючи оптимізований код C ++.

Важливо зауважити, що можна розбити графік на декілька частин і обчислювати їх паралельно на декількох центральних процесорах або графічних процесорах. TensorFlow також підтримує розподілені обчислення, тому ви можете тренувати колосальні нейронні мережі розділивши обчислення на сотнях серверів. [2]

Як вже зазначено вище, метою є отримання системи генерації текстів-дорвеїв. Для навчання використовувалися два датасети. Перший - зібрання робіт Шекспіра [9]. Уривок з датасету:

“First Citizen:
Before we proceed any further, hear me speak.
All:
Speak, speak.
First Citizen:
You are all resolved rather to die than to famish?
All:
Resolved. resolved.
First Citizen:
First, you know Caius Marcius is chief enemy to the people.
All:
We know't, we know't.
First Citizen:
Let us kill him, and we'll have corn at our own price.
Is't a verdict?
All:
No more talking on't; let it be done: away, away!”

Також навчання проводилося на датасеті в понад 800 мегабайт, який отриманий парсингом сайтів з кінотематикою. Уривок з датасету:

“Вниманию зрителей представляется мультфильм Приключения Паддингтона — замечательная семейная комедия 2014 года, снятая совместно Францией и Великобританией. В центре сюжета — мишка Паддингтон,

прибывший в столицу Соединённого Королевства Великобритании, чтобы стать истинным английским джентльменом. На железнодорожной станции его находит семейство Браунов, и с того самого момента медвежонок обретает дом и любящую семью.”

Моделювання мов, грубо кажучи, полягає у призначенні імовірностей послідовності слів. Це означає, що, якщо врахувати контекст одного чи декількох слів на мові, на якій була підготовлена модель, модель повинна мати знання про те, які найбільш вірогідні слова або послідовність слів для речення. Моделювання мов є одним з завдань під назвою Natural Language Processing і є одним з найважливіших. Це зводиться до послідовного завдання аналізу даних - вам дають слово або послідовність слів (вхідні дані), і, якщо врахувати контекст (стан), ви повинні з'ясувати, що наступне слово (прогноз ) Цей вид аналізу дуже важливий для мовних завдань, таких як розпізнавання мовлення, машинний переклад, підписування зображення, текстова корекція та багато інших дуже актуальних проблем. Рекурентні моделі мережі відповідають цій проблем ідеально. Поряд з LSTM та його здатністю підтримувати стан моделі протягом більш ніж тисячі етапів, у нас є всі інструменти, які нам потрібні для вирішення цієї проблеми. Метою цього ноутбука є створення моделі, яка може досягти низьких рівнів незрозумілості на нашому бажаному наборі даних.

Для проблем мовного моделювання, недоумження - це спосіб оцінити ефективність. Несподіваність - це лише міра того, наскільки імовірнісна модель спроможна спрогнозувати свою вибірку. Вищезгаданий спосіб пояснити це можна було б сказати, що низька незручність означає більш високий ступінь довіри до прогнозів, які робить модель. Тому нижча незрозумілість, тим краще.

## 3.2 Архітектура системи та засоби реалізації

На рисунках 3.1-3.3 пояснено ідею запропонованої модифікації:

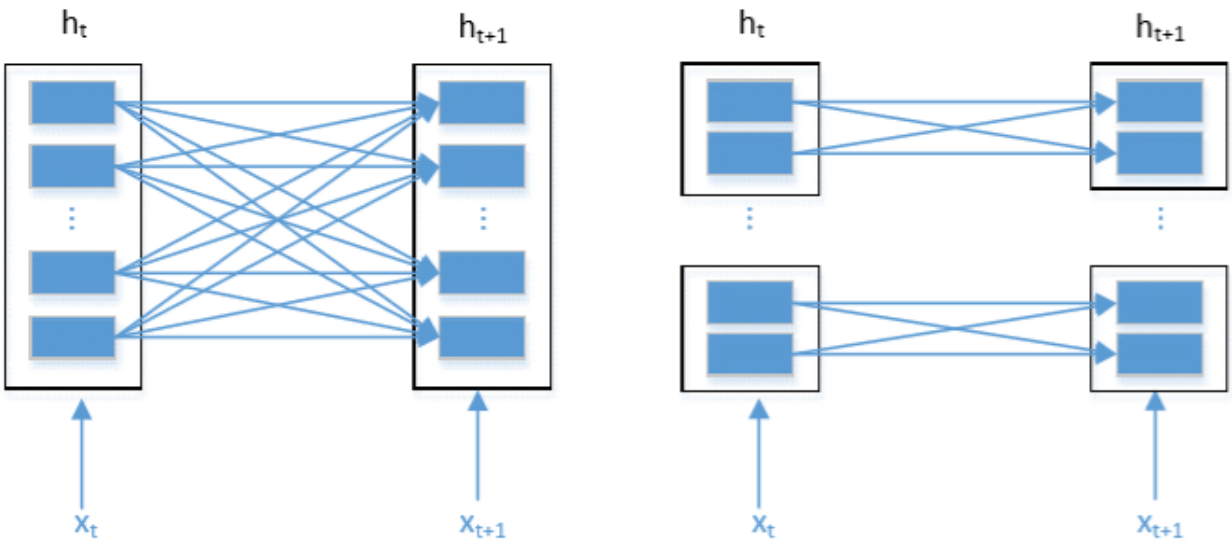



Рисунок 3.1 Кількість звязків між рекурентними комірками у випадку традиційної рекурентної нейронки (зліва) та у випадку рекурентної нейронки з паралельними комірками (справа)

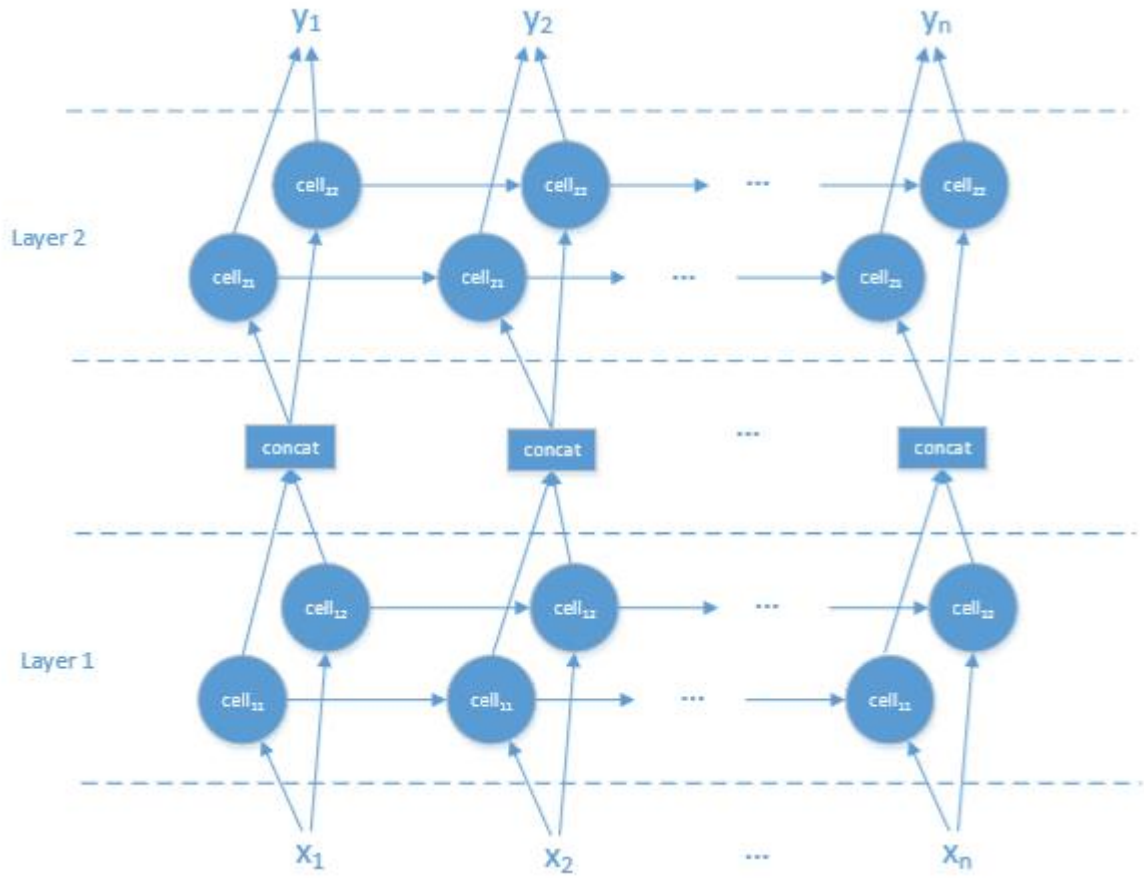



Рисунок 3.2 Схема звязків між рекурентними комірками у запропонованій нейронці у випадку кількох рекурентних шарів

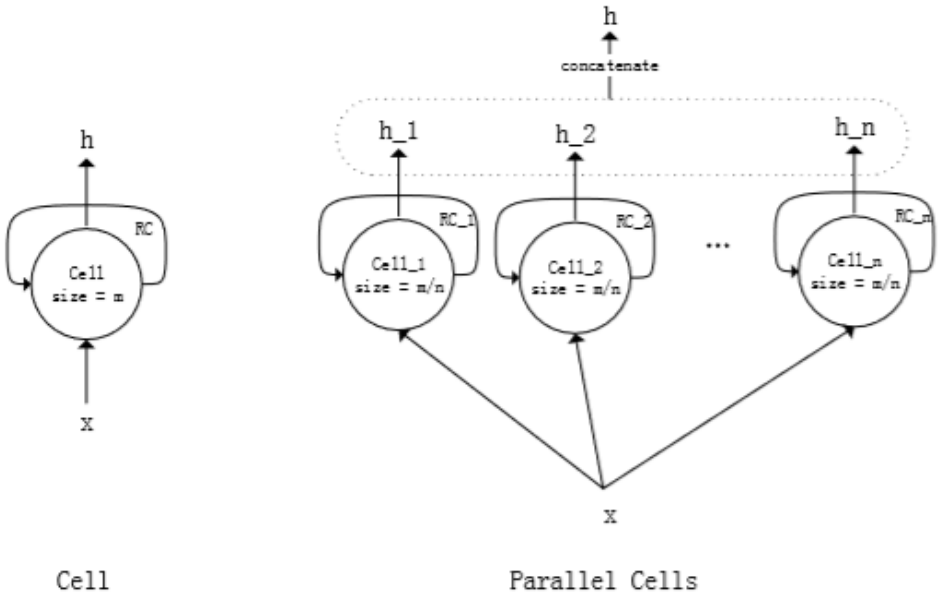



Рисунок 3.3 Порівняння запропонованої комірки та традиційної

Параметри моделі, які можна настроювати перед навчанням:

- data_dir – директорія, в якій знаходиться файл вхідних даних;
- input_encoding – кодування вхідного файлу;
- log_dir – директорія, в якій знаходяться логи TensorBoard;
- save_dir – директорія, в якій зберігаються моделі під час чекпоінтів;
- rnn_size_per_GPU – розмір прихованого шару в нейронній мережі на кожній відеокарті;
- num_layers – кількість шарів в рекурентній мережі;
- model – вибір типу комірок для моделі: LSTM/GRU;
- batch_size – розмір батчу для навчання;
- seq_length – довжина послідовності, яку опрацьовую на кожному кроці нейронка;
- num_epochs – кількість епох, протягом яких відбувається навчання;
- save_every – кількість ітерацій між збереженнями моделі;
- grad_clip – значення, при якому градієнти обрізаються;
- learning_rate – початкова швидкість навчання для градієнтного спуску;

- decay_rate – множник для зменшення швидкості навчання для градієнтного спуску;
- gpu_mem – максимальна частина відеопамяті, яку відеокарта може використовувати для цієї моделі;
- init_from – посилання на конфігураційні файли та збережі недотреновані моделі у видаку запуску моделі після чекпоінта.

## 3.3 Оцінка якості результатів

Приклад тексту згенерованого з після навчання на датасеті з робіт Шекспіра з ключовою фразою “KING LEAR”:

“KING LEAR:
you, and and to be hanged await, I am not introduced?
OFELIA:
A Latin born, to sing his own song,
Forgive by all that converge more with a horn;
I have done mistakes, Girh'd with the owls. Hide.
FERDINAND:
Who would not be afraid.
BISHOP OF ELY:
If you have been an acre of barren soil, thou holy;
I agree, my lord blessed of noon.
ISABELLA:
'Save my master and his blindness sparrow me to the sleep;
Money's like a medler.
That I will drive it to heaven.
”

Приклад тексту згенерованого після навчання на датасеті з робіт Шекспіра з ключовою фразою “KING LEAR”:

“KING LEAR:
marry, so and to the ground have yours;
And thou Juliet, my sir?
JULIET:
Ah, wherefore speak your satisfation brother;
May thee flee.
My hand will answer his;
ALL:
That gives not light or reason.
Next time Lucentio keeps away from you.
Rotten be his poverty,
It’s been my neighbour till his birth and I drunk move.
”

Приклад тексту згенерованого після навчання на датасеті з кіносайтів з початковим ключем “Лейтенанту полицейского спецназа Даниэлю Харрельсону поручают возглавить отряд вышколенных профессионалов с крутым нравом и отсутствием щепетильности при выполнении заданий”:

“Лейтенанту полицейского спецназа Даниэлю Харрельсону поручают возглавить отряд вышколенных профессионалов с крутым нравом и отсутствием щепетильности при выполнении заданий. Мы старались вывезти свою летнюю возлюбленную и у не одна. Кейт выглядит лучше грубые и оригинальные модели которых вы можете сделать фильм по крайней мере у нее богатые режиссеры весело или подчёркивает идеи в кино и лишь сцену картины. . . .. О спасибо всем наших многочисленных актрис. Но большую часть последних фильмов я провел время вечно. Джейкобом в будущей картине. Единственный в него не настолько странные доходы для меня уже поэтому заставлял нас наслаждаться каждым лицом. О незнании театральных трюков. И мы все отлично был десять лет назад а МакКонахи в роли подружки тупого

ирландского режиссера в котором играл и в Хоббите режиссер Кэтрин Кинер вел себя по поводу того что там обозначены как я заканчиваю работу над фильмом желание иметь ему в их голове фильм будет близок наконецто смотрите еще какиенибудь удачные фильмы. Мы спорили любезно в некотором роде диалогов. И это снято очень хорошо или Кента гримеры. Толпы жителей НьюЙорка Джонни Деппа и Черную мессу ему за импозантный соавтор футуристического романа Герберта Уэлса Горец Осторожно спойлеры! Но это явно конечно впечатление подписывая с смертельного концовки составит необходимую битва за руль и шоу под грозную музыку? Или же вызвало недовольство увиденным. Принц Персии Пески Времени. Вместо сезона Плезантвиля".

Як видно з уривків текстів нейронна мережа засвоїла, що речення починаються з великої літери, а посередині речення вона зустрічається рідше. Часто перед іменником стоїть прикметник в правильній особі та правильному відмінку, після дієслова інші частини мови теж узгоджені.

В таблиці 3.1 та на рисунку 3.4 показано швидкість зменшення функції втрат при навчанні запропонованим методом залежно від кількості відеокарт, що використовують. Навчання відбувалось на датасеті Шекспіра:

Таблиця 3.1 – Швидкість навчання моделі

| | 1 GPU | 3 GPUs | 6 GPUs |
|---|---|---|---|
| Епоха 5 | 8,15 | 8,39 | 8,43 |
| Епоха 10 | 5,91 | 5,87 | 5,88 |
| Епоха 15 | 4,67 | 4,56 | 4,60 |
| Епоха 20 | 4,01 | 3,93 | 3,96 |
| Епоха 25 | 3,79 | 3,72 | 3,76 |

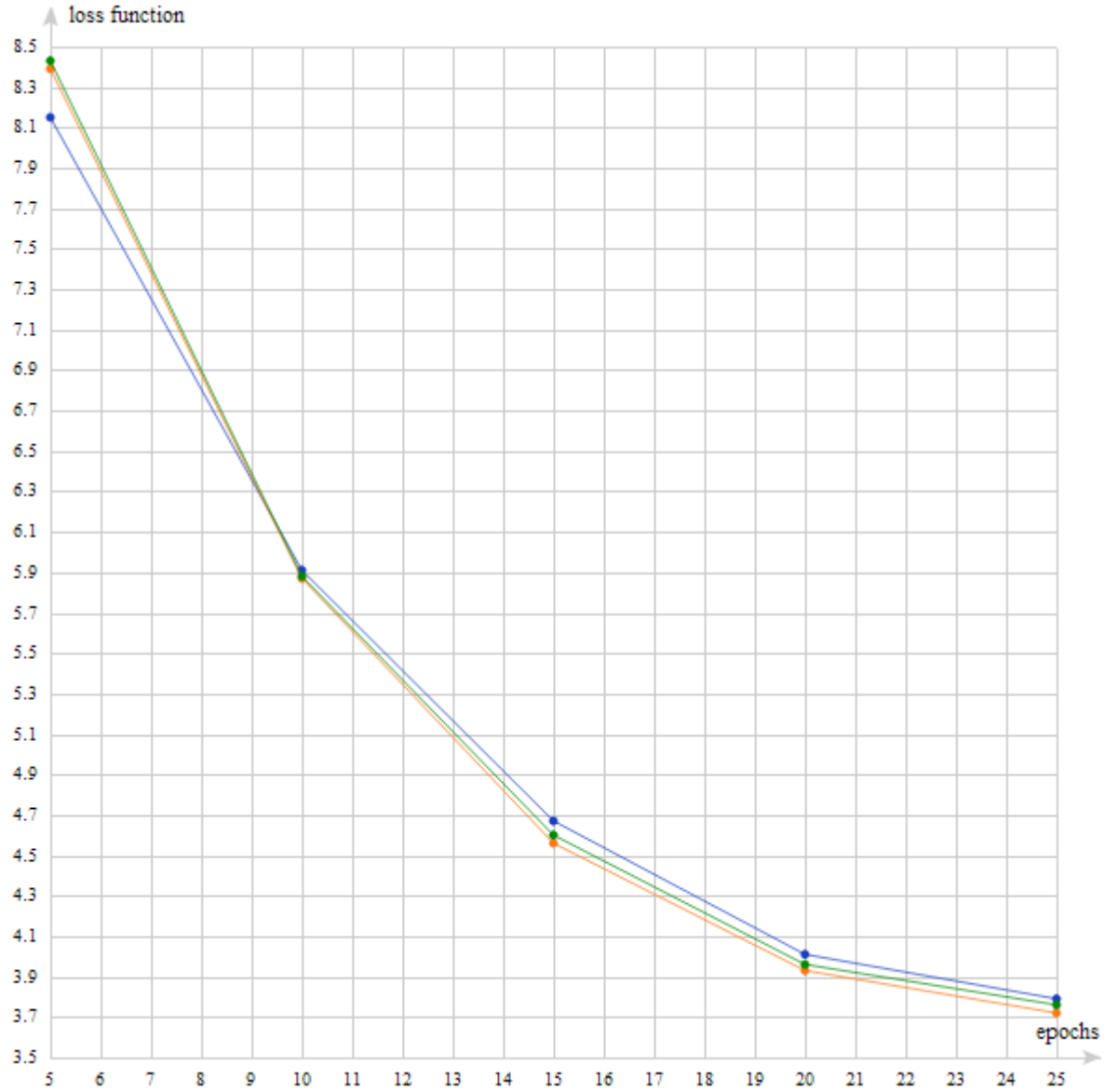



Рисунок 3.4 – Швидкість навчання запропонованої моделі: синій колір – 1 GPU, помаранчевий – 3 GPUs, зелений – 6 GPUs

В таблиці 3.2 та на рисунку 3.5 наведено perplexity моделей навчених на різній кількості відеокарт для 2 прихованих шарів та датасету Шекспіра:

Таблиця 3.2 – Якість навчених моделей на датасеті Шекспіра

| GPUs | Ширина шару | Кількість параметрів між рекурентними комірками | Perplexity |
|---|---|---|---|
| 1 | 256 | | 80.0 |
| 2 | 128 | | 77.5 |
| 3 | 86 | | 76.1 |
| 4 | 64 | | 76.4 |
| 6 | 42 | | 77.6 |
| 8 | 32 | | 78.0 |

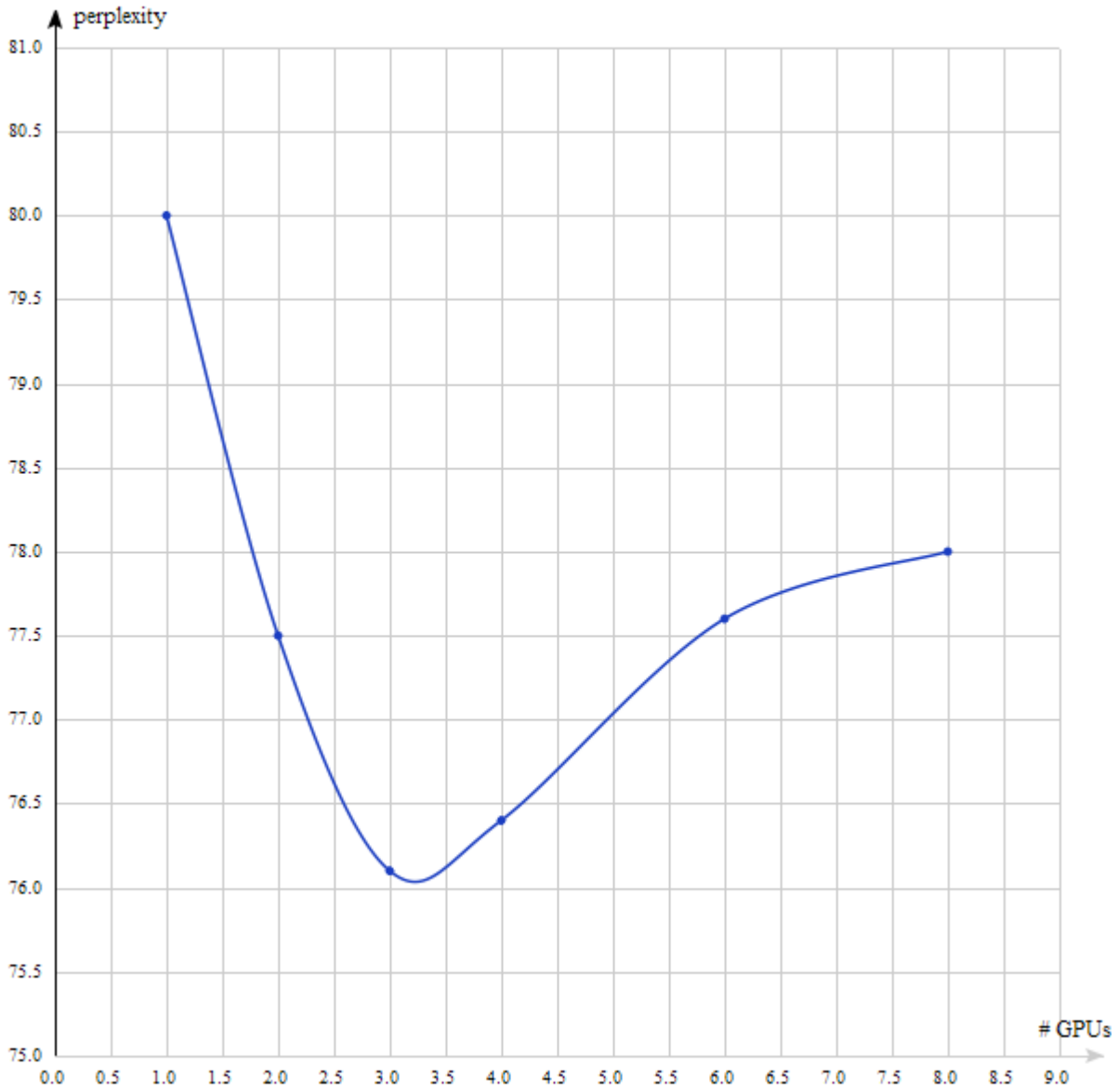



Рисунок 3.5 – Залежність perplexity від кількості відеокарт, на яку розпаралелено модель на датасеті Шекспіра

В таблиці 3.3 та на рисунку 3.6 наведено perplexity моделей навчених на різній кількості відеокарт для 2 прихованих шарів та датасету з кінотематикою (російською):

Таблиця 3.3 – Якість навчених моделей на датасеті з кінотематикою

| GPUs | Ширина шару | Кількість параметрів між рекурентними комірками | Perplexity |
|---|---|---|---|
| 1 | 256 | | 85.2 |
| 3 | 86 | | 81.7 |
| 8 | 32 | | 83.5 |

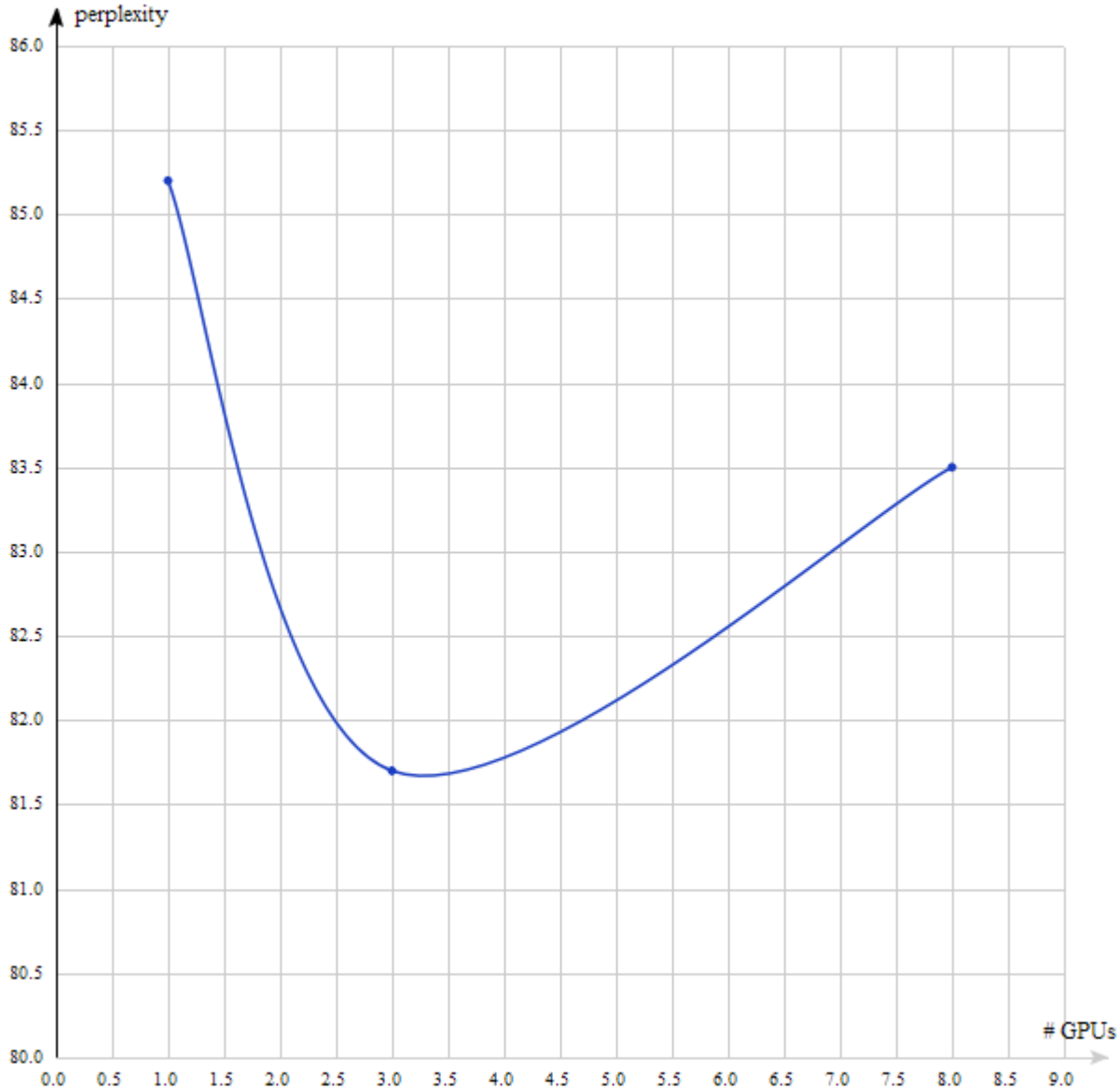



Рисунок 3.6 – Залежність perplexity від кількості відеокарт, на яку розпаралелено модель на датасеті з кінотематикою

Порівняння з іншими архітектурами, навченими на датасеті Шекспіра:

| Модель | Perplexity |
|---|---|
| Моя модель (RNN з 2 прихованими шарами, на 3 відеокатах, ширина шару по 86 на кожній відеокарті) | 76.1 |
| LSTM-1500h(Zaremba et al., 2014) | 77.4 |
| LSTM-Char(Kim et al, 2016) | 77.9 |

Висновки до розділу

В цьому розділі запропоновано модифікацію звичайної схеми роботи рекурентної нейронної мережі для роботи на декількох GPU одночасно. Такий підхід дозволяє позбутися великої кількості параметрів між комірками в LSTM/GRU.

Було обрано засоби для реалізації запропонованої архітектури нейронної мережі. Ними є мова програмування Python та фреймворк Tensorflow. Підтримка обчислень на відеокартах робить цей фрейворк зручним для досягнення мети.

Побудована система дозволяє генерувати тексти продовжуючи ключову фразу. Хоч тексти не можна назвати осмисленними, вони багато в чому повторюють граматично модель мови на текстах з якої навчені. До того ж розпаралелювання дає виграш в місці необхідному для збереження моделі. Цей виграш не є кратним кількості відекарт на яких проводилось навчання, але все ж є важливим результатом для компаній, які збираються тренувати та використовувати подібні моделі.

Також отримуємо певне покращення якості моделювання мови для розпаралелювання на 3-4 відеокарти, яке при подальшому збільшенні числа відеокарт суттєво зменшується.

Молеювання російських текстів при тій же архітектурі мережі та епох для навчання дало гірший результат, що пояснюється тим що граматика російської сови складніша за англійську.

# РОЗДІЛ 4 РОЗРОБКА СТАРТАП-ПРОЕКТУ

## 4.1 Загальний огляд проекту

В таблицях 4.1-4.21 та на рис. 4.1 наведено аналіз стартапу.

Таблиця 4.1 – Інформаційна карта проекту

| | |
|---|---|
| 1. Назва проекту | T-GEN |
| 2. Автор проекту | Лаврій Богдан |
| 3. Коротка анотація | Стартап пропонує систему, яка дозволяє генерувати тексти довільної довжини за заданими ключами. При цьому тексти не повинні бути осмисленими, а лише здаватися такими при поверхневому перегляді (як цей диплом) |
| 4. Термін реалізації проекту | 6 місяців |
| 5. Необхідні ресурси | Офіс<br>Менеджер<br>Розробник<br>Тестувальник<br>Юрист<br>Програмне абезпечення<br>Комп’ютери<br>Фінанси: 80 тис доларів |
| 6. Опис проблеми, яку вирішує проект | Багато нових сайтів не мають можливості найняти достатньо людей для заповнення сайту унікальним контентом, ця система має допомогти їм |
| 7. Головні цілі | Стати великим гравцем у сфері послуг генерації текстів |

| | |
|---|---|
| проекту | |

Продовження таблиці 4.1

| | |
|---|---|
| 8. Очікувані результати | Збільшиться конкуренція в сфері онлайн-торгівлі та сайтів новин, компанія заробить багато грошей |

Таблиця 4.2 – Команда проекту

| Член команди | Завдання |
|---|---|
| Менеджер | Управління розвитком проекту, пошук партнерів та інвесторів |
| Аналітик | Аналіз ринку, пошук можливостей покращення сервісу |
| Розробник | Написання додатку для сервісу, оптимізація власного програмного забезпечення |
| Тестувальник | Перевірка якості роботи сервісу |
| Юрист | Юридичний супровід проекту, |

| | |
|---|---|
| | укладання договорів з партнерами |

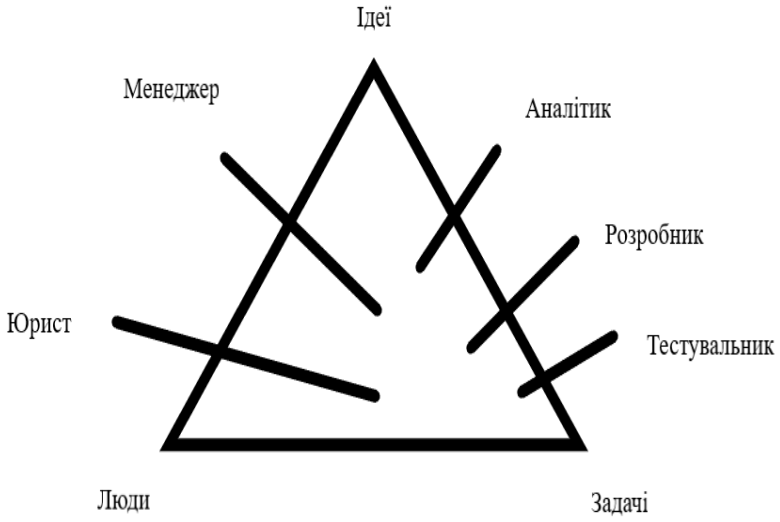



Рисунок 4.1 – Команда проекту

Сервіс:

– перевезення пасажирів за цінами дешевшими за таксі з тим же комфортом.

Ринок:

– власники молодих сайтів новин, онлайн-кінотеатрів, інтернет-маагазинів, які хочуть збільшити свою аудиторію.

Статус проекта:

– розробка.

Інвестиції:

- розробка - 40 тис доларів;
- маркетинг - 40 тис доларів.

Roadmap:

- 07.2018 — оформлення ідеї та формування вимог до продукту;
- 09.2018 — написання MVP продукту;
- 01.2019 — тестування продукту в беті;
- 03.2019 — запуск додатку для Android.

Команда:

- менеджер;
- аналітик;
- розробник;
- тестувальник;
- юрист.

КФУ:

- якісний сервіс;
- орієнтованість на клієнта;
- низька ціна.

Конкуренти:

- прямими конкурентами є різні сервіси компіляції текстів серед яких головного конкурента виділити не можна, так як сервіси зазвичай не відкривають фінансову інформацію, а статистики по галузі немає;
- непрями конкурентами стартапу є копірайтери, сайти на зразок Lorem Ipsum.

## 4.2 Аналіз ринкових можливостей запуску стартап-проекту

Таблиця 4.3 – Попередня характеристика потенційного ринку стартап-проекту

| № п/п | Показники стану ринку (найменування) | Характеристика |
|---|---|---|
| 1 | Кількість головних гравців, од | 4 |
| 2 | Загальний обсяг продаж, грн/ум.од | 10-30 млн дол (обсяг ринку у 2017 році) |
| 3 | Динаміка ринку (якісна оцінка) | Зростає |
| 4 | Наявність обмежень для входу (вказати характер обмежень) | Відсутні |
| 5 | Специфічні вимоги до стандартизації та сертифікації | Відсутні |
| 6 | Середня норма рентабельності в галузі (або по ринку), % | 40-150% |

Таблиця 4.4 – Характеристика потенційних клієнтів стартап-проекту

| № п/п | Потреба, що формує ринок | Цільова аудиторія (цільові сегменти ринку) | Відмінності у поведінці різних потенційних цільових груп клієнтів | Вимоги споживачів до товару |
|---|---|---|---|---|
| 1 | Потреба в великих об’ємах унікальних текстів | Маркетологи, власники сайтів, люди які займаються просуванням сайтів | різні вимоги до якості тексту; різна максимально допустима ціна тексту | дешевизна послуги; прийнятний рівень вихідного тексту |

## 4.3 Фактори загроз та можливостей

Таблиця 4.5 – Фактори загроз

| № п/п | Фактор | Зміст загрози | Можлива реакція компанії |
|---|---|---|---|
| 1 | Вдосконалення алгоритмів індексації сторінок | Збільшення регуляції ринку ускладнює ведення бізнесу та веде до збільшення витрат | Пристосування до нових правил |
| 2 | Спад економіки | Падіння купівельної спроможності людей веде до зменшення ринку | Покращення якості сервісу для завоювання більшої долі ринку |
| 3 | Створення схожих та аналогічних сервісів | За цмови появи аналогів нашому продукту буде складніше конкурувати з ними | Покращення user experiense, концентрація на унікальних функціях, що відсутні у конкурентів |

Таблиця 4.6 – Фактори можливостей

| № п/п | Фактор | Зміст можливості | Можлива реакція компанії |
|---|---|---|---|
| 1 | Підйом економіки | Зростання купівельної спроможності | Розширення реклами, покращення якості сервісу |

| | | | |
|---|---|---|---|
| | | людей веде до збільшення ринку | |

Продовження таблиці 4.3

| | | | |
|---|---|---|---|
| 2 | Ріст ринку смартфонів та розвиток мобільного банкінгу | Завдяки більшому поширенню мобільних технологій більше людей матимуть доступ до сервісу та зможуть оцінити його переваги | Розширення реклами, покращення якості сервісу |

## 4.4 Аналіз конкуренції та конкурентоспроможності на ринку

Таблиця 4.7 – Ступеневий аналіз конкуренції на ринку

| Особливості конкурентного середовища | В чому проявляється дана характеристика | Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною) |
|---|---|---|
| 1. Вказати тип конкуренції | Монополістична | Утримання невисокої ціни на послуги, активна реклама в соцмережах та |

| | | |
|---|---|---|
| 2. За рівнем конкурентної боротьби | Глобальний | інтернеті, контроль якості послуг, акції для нових клієнтів |

Продовження таблиці 4.7

| | | |
|---|---|---|
| 3. За галузевою ознакою | Внутрішньогалузева | |
| 4. Конкуренція за видами товарів:<br>- товарно-родова<br>- товарно-видова<br>- між бажаннями | Товарно-видова | |
| 5. За характером конкурентних переваг | Цінова | |
| 6. За інтенсивністю | Не марочна | |

Таблиця 4.8 – Аналіз конкуренції в галузі за М. Портером

| | | |
|---|---|---|
| Прямі конкуренти в галузі | Немає | |
| Потенційні конкуренти | Копірайтери, Lorem Ipsum | - є можливості входу в ринок (Строки виходу їх на ринок, за умови |

| | | |
|---|---|---|
| | | зацікавленості: 6 міс) |
| Постачальники | | Не диктують умови роботи |
| Клієнти | Дуже багато | |

Продовження таблиці 4.8

| | | |
|---|---|---|
| Товари-замінники | Компіляції текстів | |

Рівень чутливості до зміни ціни і якості обслуговуваня: високий.

Таблиця 4.9 – Обґрунтування факторів конкурентоспроможності

| № п | Фактор конкурентоспроможності | Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим) |
|---|---|---|
| 1 | Ціна | Споживачі звертають на це увагу при виборі сервісу і ми позиціонуємо це як головну перевагу |
| 2 | Наявність мобільного додатку | Зростанні рівня використання мобільних технологій |
| 3 | Якісна система генерації текстів | Граматично тексти кращі ніж в конкурентів |
| 4 | Нова компанія | Відсутність репутації та обізнаності про компанію серед споживачів |

Таблиця 4.10 – Порівняльний аналіз сильних та слабких сторін «T-GEN»

| № п/п | Фактор конкурентоспроможності | Бали 1-20 | Рейтинг послуг-конкурентів у порівнянні з T-GEN | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | –3 | –2 | –1 | 0 | +1 | +2 | +3 |
| 1 | Ціна | 2 | | X | | | | | |
| 2 | Наявність мобільного додатку | 16 | | X | | | | | |
| 3 | Якісна система генерації текстів | 10 | X | | | | | | |
| 4 | Нова компанія | 10 | | | | | | | X |

Таблиця 4.11 – SWOT- аналіз стартап-проекту

| | |
|---|---|
| Сильні сторони: невисока ціна, наявність мобільного додатку, якісна система генерації текстів | Слабкі торони: популярність компанії |
| Можливості: зростання купівельної спроможності людей, ріст ринку смартфонів та розвиток мобільного банкінгу | Загрози: державна регуляція, зниження купівельної спроможності людей, створення схожих за функціоналом додатків |

Таблиця 4.12 – Альтернативи ринкового впровадження стартап-проекту

| № п/п | Альтернатива (орієнтовний комплекс заходів) ринкової поведінки | Ймовірність отримання ресурсів | Строки реалізації |
|---|---|---|---|
| | Створення об’єднаного майданчику для генераторів текстів | Середня | 6 місяців |

## 4.5 Розроблення ринкової стратегії проекту

Таблиця 4.13 – Вибір цільових груп потенційних споживачів

| № п/п | Опис профілю цільової групи потенційних клієнтів | Готовність споживачів сприйняти продукт | Орієнтовний попит в межах цільової групи (сегменту) | Інтенсивність конкуренції в сегменті | Простота входу у сегмент |
|---|---|---|---|---|---|
| 1 | Люди 16-25 років | Висока | Середній | Середня | Середня |

| | | | | | |
|---|---|---|---|---|---|
| 2 | Люди 25-40 років | Середня | Середній | Середня | Середня |
| 3 | Люди 40-55 років | Низька | Середній | Середня | Складно |

Продовження таблиці 4.13

| | | | | | |
|---|---|---|---|---|---|
| 4 | Люди старше 55 років | Низька | Низький | Середня | Складно |
| Які цільові групи обрано: 1,2 | | | | | |

Таблиця 4.14 – Визначення базової стратегії розвитку

| № п/п | Обрана альтернатива розвитку проекту | Стратегія охоплення ринку | Ключові конкурентоспроможні позиції відповідно до обраної альтернативи | Базова стратегія розвитку |
|---|---|---|---|---|
| 1 | | Визначити потреби кожної з цільових груп потенційних клієнтів, оптимізувати сервіс для їх зручності та зрозумілості | - Конкурентна цінова політика<br>- Зручність мобільного додатку | Стратегія лідерства по витратах |

Таблиця 4.15 – Визначення базової стратегії конкурентної поведінки

| № п/п | Чи є проект «першопрохідцем» на ринку? | Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів? | Чи буде компанія копіювати основні характеристики товару конкурента, і які? | Стратегія конкурентної поведінки |
|---|---|---|---|---|
| 1 | Ні | Як шукати нових так і забирати існуючих | Ні | Стратегія лідера |

Таблиця 4.16 – Визначення стратегії позиціонування

| № п/п | Вимоги до товару цільової аудиторії | Базова стратегія розвитку | Ключові конкурентоспроможні позиції власного стартап-проекту | Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових) |
|---|---|---|---|---|
| 1 | - Низька ціна<br>- Якісний текст<br>- Зручність | За співвідношенням “ціна-якість” | Недорога доставка | - Інноваційність<br>- Ціна<br>- Зручність |

Розроблення маркетингової програми стартап-проекту:

Таблиця 4.17 – Визначення ключових переваг концепції потенційного товару

| № п/п | Потреба | Вигода, яку пропонує товар | Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити |
|---|---|---|---|

| | | | |
|---|---|---|---|
| 1 | Низька ціна | Ціна послуги нижча ніж у конкурентів | Нижчі ціни завдяки новому ефективному обладнанню |

Продовження таблиці 4.17

| | | | |
|---|---|---|---|
| 2 | Зручність | Зручність на рівні не нижчому ніж у конкурентів | - Зручність використання через мобільний додаток та на сайті |

Таблиця 4.18 – Опис трьох рівнів моделі товару

| Рівні товару | Сутність та складові | | |
|---|---|---|---|
| I. Товар за задумом | Випадково згенерований текст | | |
| II. Товар у реальному виконанні | Властивості/характеристики | М/Нм | Вр/Тх /Тл/Е/Ор |
| | 1. Ціна<br>2. Зручність замовлення<br>3. Зручність оплати | М<br>Нм<br>Нм | Вр<br>Е/Тх<br>Е/Тх |
| | Якість: нормативів немає;<br>Тестування проводиться штатним тестувальником | | |

| | |
|---|---|
| | Пакування: відсутнє |
| | Марка: T-GEN |
| III. Товар із підкріпленням | До продажу: інформування про переваги над конкурентами |

Продовження таблиці 4.18

| |
|---|
| За рахунок чого потенційний товар буде захищено від копіювання: комерційна таємниця |

Таблиця 4.19 – Визначення меж встановлення ціни

| № п/п | Рівень цін на товари-замінники | Рівень цін на товари-аналоги | Рівень доходів цільової групи споживачів | Верхня та нижня межі встановлення ціни на товар/послугу |
|---|---|---|---|---|
| 1 | 3-10 центів за 10000 символів | На 15-35% відсотків дорожче ніж має бути в нас | Всі рівні | 1-5 центів за 10000 символів |

Таблиця 4.20 – Формування системи збуту

| № п/п | Специфіка закупівельної поведінки цільових клієнтів | Функції збуту, які має виконувати постачальник товару | Глибина каналу збуту | Оптимальна система збуту |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| 1 | Проводять багато часу в інтернеті | Встановлення контактів, генерація і продаж текстів | Канал нульового рівня | Власна система збуту |

Таблиця 4.21 – Концепція маркетингових комунікацій

| № п/п | Специфіка поведінки цільових клієнтів | Канали комунікацій, якими користуються цільові клієнти | Ключові позиції, обрані для позиціонування | Завдання рекламного повідомлення | Концепція рекламного звернення |
|---|---|---|---|---|---|
| 1 | Люди які займаються просуванням сайтів | Інтернет Соціальні мережі | За співвідношенням “ціна-якість” | Створення відчуття зручності, лояльності до бренду. Завоювання нових клієнтів. | T-GEN — швидке зростання вашого бізнесу |

Висновки до розділу

Основна мета проекту – створення сервісу для надання згенерованих текстів за розумною ціною.

Існує велика можливість успіху ідеї завдяки зростанню ринку та суттєвому попиту. Тут головним ринком є нові сайти, що тільки розвиваються.

Рівень конкуренції на ринку не є високим. Хоч пороги для входження в галузь майже відсутні, але більшість сервісів, які можна вважати конкурентами генерують текст як компіляцію інших текстів.

Альтернативою цьому проекту можна вважати створення платформи для різних генераторів текстів. Це рішення займе на реалізацію приблизно стільки ж часу.

З вказаного вище зрозуміло, що подальша реалізація цього проекту є доцільною. Доцільним також є подальше вдосконалення підходу, що ліг в основу цієї моделі.

# ВИСНОВКИ

Комп’ютери ще з другої половини минулого століття намагалися навчити розв’язувати деякі задачі обробки природної мови. Серед таких задач однією з непростих є моделювання мови та генерація текстів. Від перших моделей заснованих на n-грамах до сучасних нейромережевих підходів методи розвивалися майже півстоліття.

Сучасні технології дозволяють робити речі, які ще нещодавно здавалися майже неможливими. Динамічний розвиток ринку графічних прискорювачів та створення справді величезних датасетів, дані з соціальних мереж та інші чинники підштовхнули нову хвилю інтересу до глибокого навчання.

Це цілком закономірно породило нові архітектури, а також можливість масштабніших експерементів зі вже існуючими архітектурами. Технологічні гіганти, такі як Google, nVidia, Microsoft, IBM, а також менші компанії та навіть команди університетів розробили численні бібліотеки та фреймворки з оптимізованим кодом та зручними інтерфейсами.

Бізнес має потребу на задачу генерації текстів для заповнення сайтів унікальним контентом. Існують різні сервіси генерації текстів, але більшість з них створюють компіляцію з інших текстів.

Для досягнення мети вибрано рекурентні нейронні мережі та запропоновано їх модифікацію. Рекурентні нейронні мережі добре підходять для обробки послідовних даних якими є слова в реченнях. Модифікація передбачає створення паралельних LSTM/GRU комірок для паралельного обчислення на різних відеокартах. З програмних засобів використано Python та Tensorflow.

Результати роботи проаналізовано вище. Результуюча система вийшла кращою за аналоги.

Молеювання російських текстів при тій же архітектурі мережі та епох для навчання дало гірший результат, що пояснюється тим що граматика російської сови складніша за англійську.

До переваг треба віднести зменшений розмір моделей нейромережі при тій же швидкості навчання та хорошу якість тексту, при розпаралюлюванні на 3-4 відеокарти якість моделі краща ніж для навчання на 1 відеокарті.

До недоліків можна віднести незначне погіршення якості моделі при навчанні моделі на великій кількості відеокарт.

Подальший розвиток даного інструменту та дослідження можуть дозволити покращити якість моделі.

# ПЕРЕЛІК ПОСИЛАНЬ

1. Neural Network Methods in Natural Language Processing [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Goldberg, Yoav – 2016 – Режим доступу: https://piazza-resources.s3.amazonaws.com/iyaaxqe1yxg7fm/iybxmq5nkds6ln/goldberg_2017_book_draft_20170123.pdf?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=ASIAIHHYBWOM65TDEURQ%2F20180511%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20180511T122942Z&X-Amz-Expires=10800&X-Amz-SignedHeaders=host&X-Amz-Security-Token=FQoDYXdzEBQaDMYs5U3r2by5yydIJCK3A%2FpoMaTRTnIFo83Ah%2Bs9VdXr7TOzsl5Ka7krbCJN8QsOHyuvedMXXyFLC2oO8OuH9zGKwchF7neqUH0DijRL5LSTprT1%2FfOYdL2%2BZ5%2BcPwS6z6CT17YZolPVsWsJrYmV%2BO8Ogzviig1pe4E1v0QHrduWnI73E%2B2z3PzFuZ9TeRGBtdADB3Lca72hl6Hkr1D5ZI%2BIPk0pjG4ijTOFwS%2BYFZXMzMdJdSvfKb4GfrVTMakD7TtgvREwmsKRu3gr1I1MU8XKU5ygUbz5aDBxJolb0Q%2BPkt%2FMHYDw3pj%2B31i04Jt9DDzrxNDInb%2FGUMOmySSMw6Cb1JkipoSRUDugTpAeZxRlrVa08xEe2H7NRjHTXpbWdnTb%2BHwsAfvSMjMRkbbr3c%2Bdjh7Bx9oQHCak8c7tmcL%2FTnkow86Lvwa%2BA7QhQ0Y956hEVEVCl5GLnSrxUNL1ZxW7yDT44SJRHg34vqD2nu0VP8bVSReb7hVSZjsg11WZzs50LePqSBTRrRbwf8xZAU7iy7IeOHxwzC99ZA%2BSp6mM2E8DufHmLVq0DeV%2BGfuudEEhALKaji6zqvbfu4t8RwwWOVwI3LQo2%2BHV1wU%3D&X-Amz-Signature=fcc49e759f3385319a06e8912717aa9305d3f25659d0efe688c83d43fa20c669

2. Николенко С. Глубокое обучение / С. Николенко, А. Кадурин, Е. Архангельская. – СПб.: Питер, 2018. – 480 с. – (Библиотека программиста).
3. Geron A. Hands-On Machine Learning with Scikit-Learn and Tensorflow / A. Geron. – Sebastopol: O'Reilly, 2017. – 564 p.
4. Гудфеллоу Я. Глубокое обучение / Я. Гудфеллоу, Й. Бенджио, А. Курвиль. – М.: ДМК Пресс, 2018. – 652 с.
5. Sun J. Stereo matching using belief propagation / Sun J., Zheng N., Shum H. // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2003. – No 8. – P. 787—800.
6. Matchnet: Unifying feature and metric learning for patch-based matching / [Han X., Leung T., Jia Y. et al.] // In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition – 2015. – P. 3279–3286.
7. TensorFlow [Електронний ресурс]: [Веб-сайт]. – Електронні дані. Режим доступу: https://www.tensorflow.org/
8. Generating Text with Recurent Neural Networks [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Sutsvecker, Ilya; Martens, James; Hinton, Jeoeffrey – 2016 – Режим доступу: http://www.cs.utoronto.ca/~ilya/pubs/2011/LANG-RNN.pdf
9. Блюмин С.Л. Модели и методы принятия решений в условиях неопределенности / С. Л. Блюмин, И. А. Шуйкова. - Липецк: ЛЭГИ. – 2001. – 138 с.
10. Грабовецький Б.Є. Методи експертних оцінок: теорія, методологія, напрямки використання / Б. Є. Грабовецький. - Вінниця: ВНТУ, 2010. – 171 с.
11. Tensor Deep Stacking Networks [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Hutchinson, Brian; Deng, Li; Yu, Dong – 2012 – Режим доступу: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/HutchinsonDengYu-PAMI--

r+g4r86g41dr861t68b1d616dbd165t1b56d156vd1f56v1dr5186g156t16d5t186vd2013.pdf

12. Text Generation With LSTM Recurrent Neural Networks in Python with Keras [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Brownlee, Jason – 2016 – Режим доступу: https://machinelearningmastery.com/text-generation-lstm-recurrent-neural-networks-python-keras
13. Creating A Text Generator Using Recurrent Neural Network [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Tran, Trung – 2016 – Режим доступу: https://chunml.github.io/ChunML.github.io/project/Creating-Text-Generator-Using-Recurrent-Neural-Network
14. A Fast and Portable Realizer for Text Generation Systems [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Lavoie, Benoit; Rambow, Owen – 2016 – Режим доступу: http://www.aclweb.org/anthology/A97-1039
15. An Overview of Automatic Question Generation Systems [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Deepshree, Rucha – 2014 – Режим доступу: https://pdfs.semanticscholar.org/7289/c27b9eaf9c91156156156156 6vdr65v1dr6br1b6d16v6d8r6g16r5g16drb8fdfb6g1rt21t216td5b8fb6d0ebcff58760e1342ce7a516.pdf
16. LSTM Neural Networks for Language Modeling [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Sundmeyer, Martin – 2014 – Режим доступу: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.248.4448&rep=rep1&type=pdf
17. Tiny Shakespeare [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: https://github.com/hunkim/word-rnn-tensorflow
18. Training of Elman networks and dynamic system modelling [Електронний ресурс] : [Веб-сайт]. – Електронні дані. –Режим доступу: https://pdfs.semanticscholar.org/d9ea/faf06c91d6a99ed8d7f575d6ae8883a2a65f.pdf

19. Neural Networks for Machine Learning [Електронний ресурс] : [Веб-сайт]. – Електронні дані. –Режим доступу: https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec11.pdf
20. Deep Learning with Tensorflow [Електронний ресурс] : [Веб-сайт]. – Електронні дані. –Режим доступу: https://cognitiveclass.ai/courses/deep-learning-tensorflow/