CTU
CZECH TECHNICAL UNIVERSITY IN PRAGUE

F3

Faculty of Electrical Engineering
Department of Cybernetics

Bachelor's Thesis

NLP Methods for Product Review Analysis

Ondřej Jiří Beneš

May 2024
Supervisor: Ing. Jan Drchal, Ph.D.

BACHELOR‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Beneš Ondřej Jiří** Personal ID number: **498943**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Cybernetics**

Study program: **Open Informatics**

Specialisation: **Artificial Intelligence and Computer Science**

II. Bachelor’s thesis details

Bachelor’s thesis title in English:

NLP Methods for Product Review Analysis

Bachelor’s thesis title in Czech:

Metody zpracování přirozeného jazyka pro analýzu recenzí

Guidelines:

The task is to explore possibilities to extract structured information from product reviews to get statistics on the positive and negative attributes of the product. The problem is related to sentiment analysis and text clustering.

1. Review the state-of-the-art large language models, sentiment analysis, and text clustering methods.
2. Collect product review data of a selected domain (suggested by the supervisor) and preprocess them.
3. Analyse reviews and develop a method of extracting positive and negative product descriptors.
4. Evaluate the method using standard metrics and domain experts (if available).

Bibliography / sources:

[1] Reimers, Nils, and Iryna Gurevych. "Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation." Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2020.
[2] Wankhade, Mayur, Annavarapu Chandra Sekhara Rao, and Chaitanya Kulkarni. "A survey on sentiment analysis methods, applications, and challenges." Artificial Intelligence Review 55.7 (2022): 5731-5780.
[3] Ghosal, Attri, et al. "A short review on different clustering techniques and their applications." Emerging Technology in Modelling and Graphics: Proceedings of IEM Graph 2018 (2020): 69-83.

Name and workplace of bachelor’s thesis supervisor:

Ing. Jan Drchal, Ph.D. Artificial Intelligence Center FEE

Name and workplace of second bachelor’s thesis supervisor or consultant:

Date of bachelor’s thesis assignment: **07.02.2024** Deadline for bachelor thesis submission: **24.05.2024**

Assignment valid until: **21.09.2025**

Ing. Jan Drchal, Ph.D. Supervisor’s signature	prof. Dr. Ing. Jan Kybic Head of department’s signature	prof. Mgr. Petr Páta, Ph.D. Dean’s signature

III. Assignment receipt

The student acknowledges that the bachelor’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor’s thesis, the author must state the names of consultants and include a list of references.

Date of assignment receipt	Student’s signature

Acknowledgement / Declaration

I would like to thank my supervisor Ing. Jan Drchal, Ph.D. for giving me the opportunity to work on this project, his guidance, and his willingness.

Also, I would like to thank my family for their support and my girlfriend for her patience.

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, 24 May 2024

....

Abstrakt / Abstract

Tato práce testuje klasické NLP metody k získání nových indikací nebo vedlejších účinků molekul (léků) z jejich uživatelských recenzí. Prvně je vytvořen dataset, ze dvou webových stránek, který obsahuje názvy léků, uživatelské recenze a již zmapované indikace a vedlejší účinky. Poté je proveden clustering, který spojí podobné indikace nebo vedlejší účinky na základě kosinové podobnosti vektorů získaných z modelu SBERT. Představeny jsou čtyři přístupy: two-tower model, zero/few-shot prompting a fine-tuning jazykového modelu, na kterém je potom zero/few-shot prompting proveden znova. Nakonec je zadefinována ztrátová funkce a jsou zevaluovány výstupy jednotlivých přístupů. Ukázal jsem, že generativní přístupy dosáhly lepších výsledků než two-tower model.

Klíčová slova: nové indikace nebo vedlejší účinky, NLP metody, two-tower model, zero/few-shot prompting

This thesis tests the classic NLP methods for obtaining new indications or side effects of molecules (drugs) from their user reviews. Firstly, the dataset from two websites is made. This dataset contains names of molecules, user reviews, and well-known indications and side effects. The clustering is then performed to combine similar indications or side effects based on the cosine similarity of vectors obtained from the SBERT. Four approaches are presented: the two-tower model, the zero/few-shot prompting, and the supervised fine-tuning of the large language model on which the zero/few-shot prompting is performed again. Ultimately, the loss function is defined, and different approaches are evaluated. I have shown that the generative approaches achieved better results than the two-tower model.

Keywords: new indications or side effects, NLP methods, two-tower model, zero/few-shot prompting

/ Contents

Tables / Figures

Chapter 1
Introduction

Today's age allows the public to access a vast amount of information about drugs. Some of the most widely used sources include online pharmaceutical encyclopedias and community forums where users can share their user reviews on drugs. This almost informal flow of information creates a possibility for identifying new indications or side effects of drugs.

The development of new drugs (new molecular entities) is money and time expensive. The total cost usually includes research and development costs. In the U.S., the cost is estimated to be between $161 million and $4.54 milliard. Some specific groups of drugs have an even greater lower bound of the cost interval; for example, $944 million for anticancer drugs [1]. Before new drugs are marketed, there are four phases in the U.S.: Discovery and Development, Preclinical Research, Clinical Research, and FDA[1] Review. These phases take 10.5 years on average [2]. In Europe, the process, the cost, and the time are similar. Most drugs do not pass Phase I clinical trials because of high toxicity and inefficiency [3]. The approach called *drug repurposing* is researching new indications for existing drugs. One example could be sildenafil [4]. The original indication was high blood pressure and angina, but during clinical trials, a new indication has been discovered: erectile dysfunction. The main advantage of this approach is that a drug already has a complex clinical profile, thereby reducing risk and cost during development and the development time itself. It is, therefore, not surprising that the number of repurposed drugs on the market continues to increase. The WHO[2] has even called this approach *the underrated champion of sustainable innovation* [5].

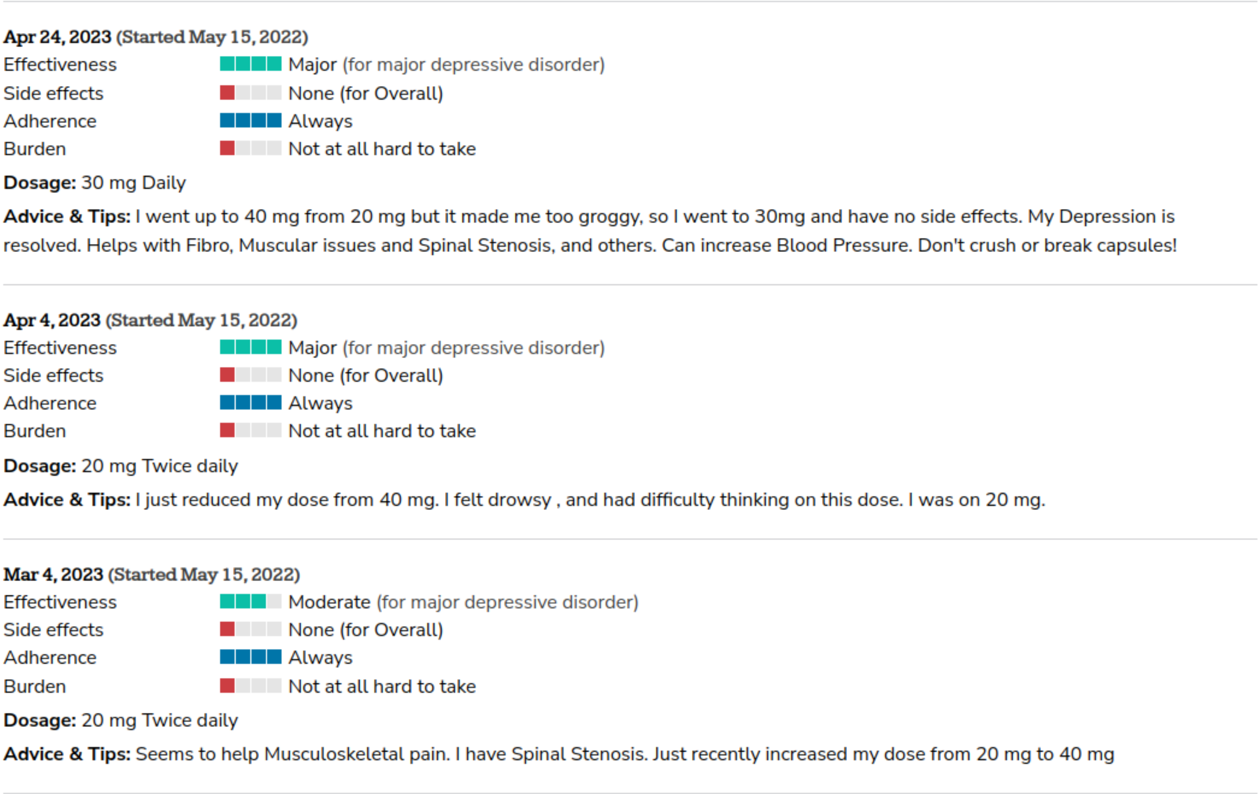
For 71 of the 222 newly approved drugs, from 2001 to 2010, new side effects that did not manifest during clinical trials were found [6]. Clinical trials need more time, sample size (usually ranging from a few hundred to a few thousand respondents), and rich sample distribution. Many drugs are approved with the condition that there will be post-market studies that will analyze medical reports–these studies take up to years. MedWatch[3] is one way to discover new side effects in already marketed drugs. It relies on voluntary input from users and healthcare professionals. Another example is efalizumab [7]. It was approved by the FDA in 2003 and withdrawn from the market in 2009 due to the potential risk of brain disease. Therefore, early detection of side effects is a crucial step to ensure safety. This may also result in reduced hospital costs.

The goal of this thesis is to analyze user reviews on drugs and try to extract new indications or side effects that are not well-known for those drugs.

[1] The FDA is the federal agency that regulates food, drugs, medical devices, and other products in the U.S.

[2] The World Health Organization (WHO) is a specialized agency of the United Nations responsible for international public health.

[3] MedWatch is the Food and Drug Administration's *Safety Information and Adverse Event Reporting Program.*



Apr 24, 2023 (Started May 15, 2022)

Effectiveness	🟩🟩🟩🟩 Major (for major depressive disorder)
Side effects	🟥⬜⬜⬜ None (for Overall)
Adherence	🟦🟦🟦🟦 Always
Burden	🟥⬜⬜⬜ Not at all hard to take

Dosage: 30 mg Daily

Advice & Tips: I went up to 40 mg from 20 mg but it made me too groggy, so I went to 30mg and have no side effects. My Depression is resolved. Helps with Fibro, Muscular issues and Spinal Stenosis, and others. Can increase Blood Pressure. Don't crush or break capsules!

Apr 4, 2023 (Started May 15, 2022)

Effectiveness	🟩🟩🟩🟩 Major (for major depressive disorder)
Side effects	🟥⬜⬜⬜ None (for Overall)
Adherence	🟦🟦🟦🟦 Always
Burden	🟥⬜⬜⬜ Not at all hard to take

Dosage: 20 mg Twice daily

Advice & Tips: I just reduced my dose from 40 mg. I felt drowsy , and had difficulty thinking on this dose. I was on 20 mg.

Mar 4, 2023 (Started May 15, 2022)

Effectiveness	🟩🟩🟩⬜ Moderate (for major depressive disorder)
Side effects	🟥⬜⬜⬜ None (for Overall)
Adherence	🟦🟦🟦🟦 Always
Burden	🟥⬜⬜⬜ Not at all hard to take

Dosage: 20 mg Twice daily

Advice & Tips: Seems to help Musculoskeletal pain. I have Spinal Stenosis. Just recently increased my dose from 20 mg to 40 mg

Figure 1.1. Sample of user reviews on duloxetine

User reviews are scraped from two sources: Drugs.com[4] and PatientsLikeMe[5]. I use a large language model (LLM)–artificial neural network built with the Transformer-based architecture and natural language processing (NLP) methods such as the two-tower model, the prompt engineering, and the supervised fine-tuning. The scraped data with the prompt forming the input that may look like:

```
Extract possible indications from the following text:

Seems to help Musculoskeletal pain. I have Spinal Stenosis.
Just recently increased my dose from 20 mg to 40 mg
```

This input is passed to the model, and the expected output may look like:

```
["musculoskeletal pain"]
```

Ultimately, I compare potentially new results with well-known indications or side effects.

[4] Drugs.com is an online pharmaceutical encyclopedia that provides drug information for consumers and healthcare professionals.

[5] PatientsLikeMe is an integrated community, health management, and real-world data platform.

Chapter 2
Review of State-of-the-Art

In this chapter, I describe how the attention mechanism works, mainly how sentence embeddings are obtained, the Llama 2 (chat) improvements, the low-rank adaption used for the fine-tuning, and how different approaches are made.

2.1 Transformer

Before the Transformer, sequence models were based on recurrent or convolutional networks using the attention mechanism, where a sequence could be a text, an image, audio, or similar. In 2017, Google developed the Transformer that only uses the attention mechanism without the need for recurrence or convolution [8]. Removing recurrence allowed the possibility of better parallelization and accelerated the ability to train. Today's LLM architectures are built primarily on the Transformer.

The Transformer has two parts: the encoder and the decoder. Both parts comprise a stack of identical layers and use the (masked) multi-head attentions and feed-forward networks.

2.1.1 Attention Mechanism

Firstly, inputs are linearly transformed into Q, K, and V matrices. Then, the attention is computed as:

$$\text{self-attention}(Q, K, V) = \text{softmax}(Q \cdot K^{\mathrm{T}}) \cdot V$$

$$\text{attention}(X) = \text{self-attention}(W_{\mathrm{Q}} \cdot X, W_{\mathrm{K}} \cdot X, W_{\mathrm{V}} \cdot X)$$

Multi-head attention includes h attentions stacked onto each other, therefore performing h attention computations:

$$\text{multi-head attention}(X) = \text{concat}(\text{head}_1, \ldots, \text{head}_h) \cdot W_{\text{proj}}$$

$$\text{where } \text{head}_i = \text{attention}(X)$$

The masked multi-head attention is the basic multi-head attention with one change: the output from the decoder passed to the decoder again has masked all following sequences after the current sequence. This approach is used during training, where the model should not know the correct sequence ahead.

$$\text{self-attention}(Q, K, V) = \text{softmax}(Q \cdot K^{\mathrm{T}} + M) \cdot V$$

$$\text{where } M = \begin{pmatrix} 0 & -\infty & -\infty & \cdots & -\infty \\ 0 & 0 & -\infty & \cdots & -\infty \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

That one change filters out specific rows from the matrix V because the softmax assigns zero to all $-\infty$.

2.1.2 Sentence Embeddings

The Transformer model does not only have to be used for generating sequences but also for creating embeddings from input by using only the first part of the architecture: the encoder.

Obtaining a single word embedding from an output could be done by taking the last embedding from the encoder or averaging a couple of them since models like the BERT[1] use a stack of h encoders onto each other. Nevertheless, sentence embedding is a bit different. One way to do it is to pass a whole sentence through a stack of encoders and either average the outputs or use the last embedding of `[CLS]` token (or a mean of couple ones). There is, however, no convention about it, and embeddings are not that good.

The SBERT approaches this disadvantage and proposes the option to get sentence embeddings directly [9].

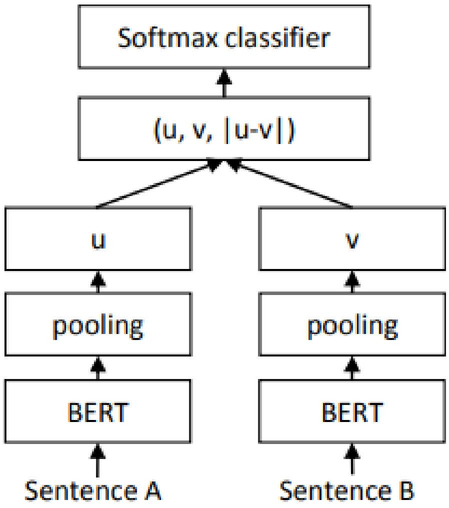


Figure 2.1. SBERT architecture with the classification objective function from the *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks* paper [9]

Both BERT models have tied weights (siamese networks) and have already been pre-trained to produce meaningful sentence embeddings. The fact that the BERT model is pre-trained significantly reduces fine-tuning time. The preferred pooling strategy is computing the mean of all output vectors. During the fine-tuning, the SBERT experimented with three objective functions:

1. Classification Objective Function (the NLI task using the combination of the SNLI[2] and the Multi-Genre NLI[3] dataset)

$$o = \mathrm{softmax}(W_{\mathrm{t}}(u, v, |u - v|)) \tag{see Figure 2.1}$$

where $W_{\mathrm{t}} \in \mathbb{R}^{3n \times k}$, n is the dimension of sentence embeddings, k the number of labels; for example: *contradiction*, *entailment*, and *neutral*, and $(u, v, |u - v|)$ is the concatenation of embeddings u, v, and the element-wise difference $|u - v|$. Since the

[1] https://arxiv.org/pdf/1810.04805
[2] https://nlp.stanford.edu/projects/snli
[3] https://cims.nyu.edu/ sbowman/multinli

Text	Judgment	Hypothesis
Children smiling and waving at camera.	neutral	They are smiling at their parents.
Children smiling and waving at camera.	entailment	There are children present.
Children smiling and waving at camera.	contradiction	The kids are frowning.

Table 2.1. Sample from the SNLI corpus

softmax classifier returns probabilities for each label, the SBERT minimizes the cross-entropy function: $H(o, p) = -\sum_i p_i \cdot \log(o_i)$, where p_i is the correct probability for a given judgment.

2. Regression Objective Function (the STS task using the STSb[4] dataset)

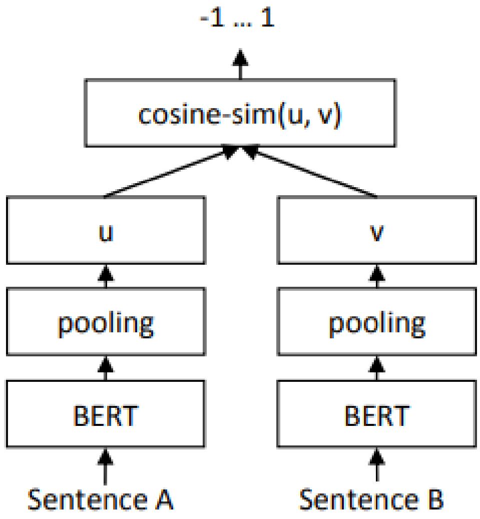


Figure 2.2. SBERT architecture with the regression objective function from the *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks* paper [9]

$$\text{cosine-sim}(u, v) = \frac{\langle u|v\rangle}{\|u\|_2 \cdot \|v\|_2}$$

Minimized is the mean squared error function: $\text{MSE}(c, y) = (c - y)^2$, where c is the cosine similarity value obtained by the model, and y is the normalized (to $[-1; 1]$) value obtained from the dataset. The SBERT experimented with two setups: only training on the STSb, and first training on the NLI, then on the STSb.

sentence1	sentence2	y
A man is smoking.	A man is skating.	0.5
Some men are fighting.	Two men are fighting.	4.25
A plane is taking off.	An air plane is taking off.	5

Table 2.2. Sample from the STSb corpus

[4] https://huggingface.co/datasets/nyu-mll/glue/viewer/stsb

3. Triplet Objective Function[5]

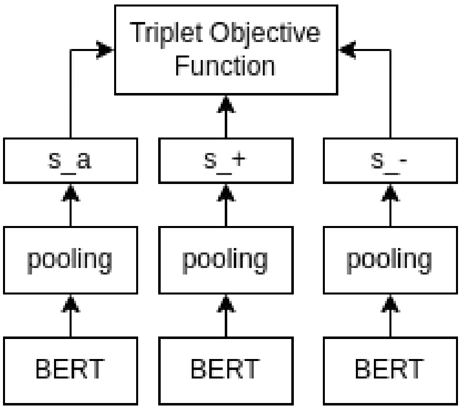


Figure 2.3. SBERT architecture with the triplet objective function

$$\max(\|s_a - s_+\|_2 - \|s_a - s_-\|_2 + \varepsilon, 0)$$

where s_a is the sentence embedding for an anchor sentence a, s_+ for a positive sentence, and s_- for a negative sentence. The function is minimized; meaning the distance between s_a and s_+ should be smaller than the distance between s_a and s_- by at least the margin ε.

The SentenceTransformers[6] Python library provides a way to compute the cosine similarity of embeddings obtained from the model.

$$\text{cos-sim}(\mathbf{A}_{m\times n}, \mathbf{B}_{o\times n}) = (\mathbf{A}_{\text{normalized}} \cdot \mathbf{B}^T_{\text{normalized}})_{m\times o}$$

$$\mathbf{A}_{\text{normalized}}[i,j] = \frac{\mathbf{A}[i,j]}{\|\mathbf{A}[i,:]\|_2}, \text{where } \mathbf{A}[i,:] \text{ is the } i\text{-th row of the matrix } \mathbf{A}$$

2.2 Llama 2

Unlike the BERT, the Llama 2 contains only the decoder part of the Transformer with several changes like more tokens, the bigger context window, and the usage of grouped-query attention [10]. To define the grouped query-attention, I have first to define the multi-query attention:

$$\text{multi-query attention}(X) = \text{concat}(\text{query}_1, \ldots, \text{query}_h) \cdot W_{\text{proj}}$$

$$\text{where query}_i = \text{self-attention}(W_{\text{Q}_i} \cdot X, W_{\text{K}} \cdot X, W_{\text{V}} \cdot X)$$

$$\text{grouped-query attention}(X) = \text{concat}(\text{head}_1, \ldots, \text{head}_h)$$

$$\text{where head}_i = \text{multi-query attention}(X)$$

The chat version is fine-tuned using the Reinforcement Learning with Human Feedback model, where annotators decide which responses are better. During multi-turn conversations, the loss of context occurs. The Llama 2 addresses this issue by concatenating the instruction to all user messages.

[5] using the dataset from https://aclanthology.org/P18-2009.pdf

[6] https://www.sbert.net

2.3 Low-Rank Adaptation

There are two problems when training a large model: computational and storage costs. A large model has many parameters, and both parameters and gradients must be stored. The LoRA solves these two problems [11].

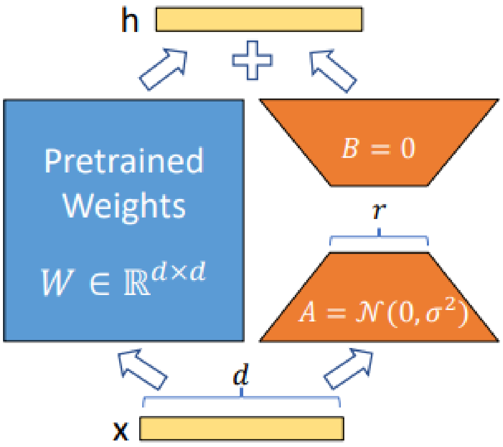

h
Pretrained
Weights
$W \in \mathbb{R}^{d \times d}$
$B = 0$
r
$A = \mathcal{N}(0, \sigma^2)$
d
x

Figure 2.4. LoRA approach from the *LoRA: Low-Rank Adaptation of Large Language Models* paper [11]

The LoRA adds extra weights and allows us to fine-tune only those weights while freezing the original ones. The great advantage of this approach is that the model does not forget what it has been trained on. In Figure 2.4, only A and B parameters are trained.

In a basic sense, training a neural network is adjusting weights matrices by its gradients: $W + W_{\text{grad}}$. The matrix W has a rank representing a number of linearly independent columns. If there are any linearly dependant columns, they can be removed without loss of information since they can be calculated again by combining the remaining columns from the matrix. The LoRA's idea is not to optimize full matrices, but rather low-ranked decomposed ones.

$$(W_{\text{grad}})_{m \times n} = B_{m \times r} \cdot A_{r \times n}$$

$$\text{where } m \times n \gg (m \times r) + (r \times n)$$

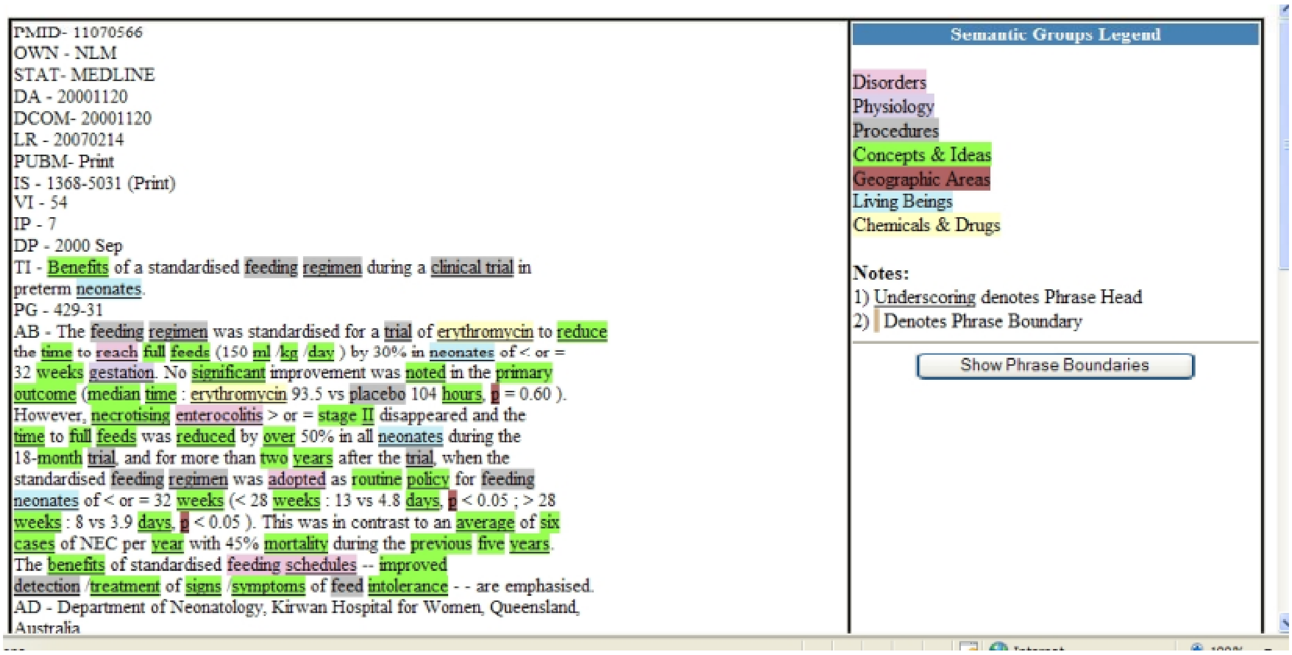
r is the hyperparameter to choose. Too low a value could lead to loss of information because of removing linearly independent columns.

2.4 Different Approaches

One of the popular professional tools is MetaMap[7]. It has been developed at the National Library of Medicine (NLM) and maps biomedical text to the UMLS Metathesaurus, so called a named-entity recognition task. UMLS[8] stands for Unified Medical Language System, and Metathesaurus[9], the semantic network, links similar

[7] https://lhncbc.nlm.nih.gov/ii/tools/MetaMap.html
[8] https://www.nlm.nih.gov/research/umls/index.html
[9] https://www.nlm.nih.gov/research/umls/knowledge_sources/metathesaurus/index.html

names for the same concept from many vocabularies. MetaMap is one of the building blocks for the Medical Text Indexer[10] (MTI).



PMID- 11070566
OWN - NLM
STAT- MEDLINE
DA - 20001120
DCOM- 20001120
LR - 20070214
PUBM- Print
IS - 1368-5031 (Print)
VI - 54
IP - 7
DP - 2000 Sep
TI - Benefits of a standardised feeding regimen during a clinical trial in preterm neonates.
PG - 429-31
AB - The feeding regimen was standardised for a trial of erythromycin to reduce the time to reach full feeds (150 ml /kg /day) by 30% in neonates of < or = 32 weeks gestation. No significant improvement was noted in the primary outcome (median time : erythromycin 93.5 vs placebo 104 hours, p = 0.60). However, necrotising enterocolitis > or = stage II disappeared and the time to full feeds was reduced by over 50% in all neonates during the 18-month trial, and for more than two years after the trial, when the standardised feeding regimen was adopted as routine policy for feeding neonates of < or = 32 weeks (< 28 weeks : 13 vs 4.8 days, p < 0.05 ; > 28 weeks : 8 vs 3.9 days, p < 0.05). This was in contrast to an average of six cases of NEC per year with 45% mortality during the previous five years. The benefits of standardised feeding schedules -- improved detection /treatment of signs /symptoms of feed intolerance - - are emphasised.
AD - Department of Neonatology, Kirwan Hospital for Women, Queensland, Australia

Semantic Groups Legend

Disorders
Physiology
Procedures
Concepts & Ideas
Geographic Areas
Living Beings
Chemicals & Drugs

Notes:
1) Underscoring denotes Phrase Head
2) | Denotes Phrase Boundary

Figure 2.5. Sample from the MetaMap 3D from the *UMLS Webcast: The Currect State of MetaMap and MMTx* [12]

Another approach is from the *Large Language Models are Few-Shot Clinical Information Extractors* paper [13]. They use InstructGPT[11] and the zero/one-shot prompting for extracting clinical information like a list of medications, interventions, or attributes like dosage or reason.

[10] https://lhncbc.nlm.nih.gov/ii/tools/MTI.html
[11] https://arxiv.org/pdf/2203.02155

Chapter 3
Approach

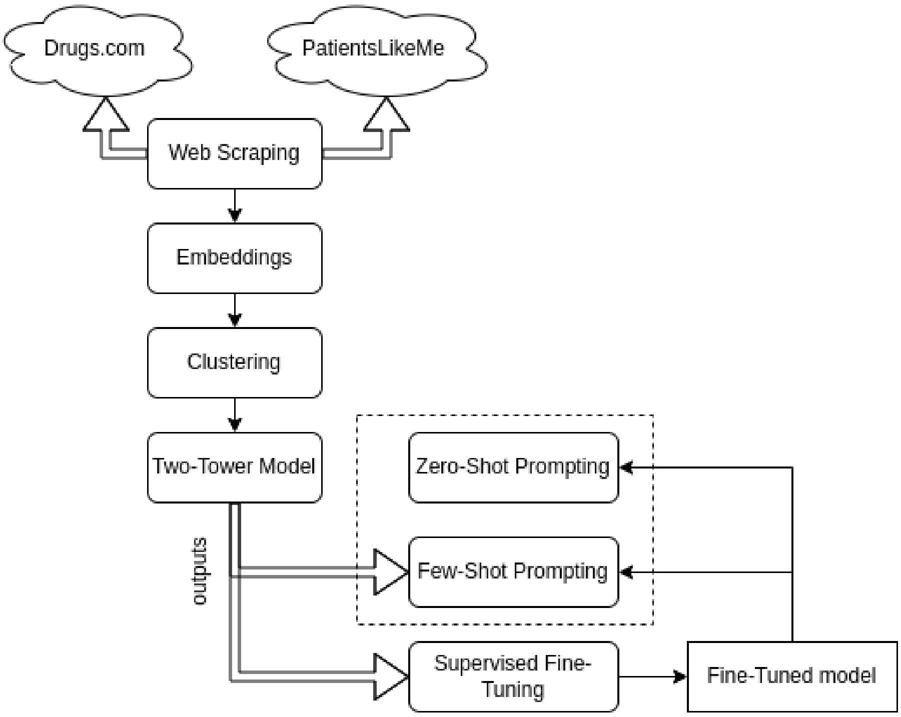


Figure 3.1. Design of the approach

My approach includes (1) web scraping data from Drugs.com and PatientsLikeMe, (2) transforming the data (indications, side effects, and user reviews) to a vector space, (3) finding out similarities between those vectors and their clustering, (4) finding out new indications or side effects using the two-tower model, (5) zero-shot prompting a LLM, (6) few-shot prompting a LLM with the usage of the two-tower model outputs, (7) supervised fine-tuning with the usage of the two-tower model outputs and zero/few-shot prompting the fine-tuned model.

(1) For web scraping, I use the Scrapy[1] framework for its simplicity and variety, such as asynchronism, scheduling requests, possible built-in caching, and auto-throttling extension. I use the scrapy-selenium[2] package, which is a bit old and contains bugs, but I have partially fixed it. Molecule's URLs must be manually found because both platforms have many inconsistencies in the format. After that, I process data to remove unwanted characters or redundancy.

(2) I need to work with vectors when using NLP approaches. The model used for transforming texts into a vector space is *all-mpnet-base-v2*[3] from Microsoft. Its size is 420 MB, and it has been trained on over one milliard training pairs (see Section 2.1.2).

[1] https://scrapy.org
[2] https://github.com/clemfromspace/scrapy-selenium
[3] https://huggingface.co/microsoft/mpnet-base

(3) Since data could still have redundancy, I need to do clustering. Firstly, I visualize embeddings using the t-SNE algorithm because it focuses on local neighbors. The clustering is done using the `community_detection` function provided by the SentenceTransformers library. It uses the cosine similarity and the algorithm to extract communities.

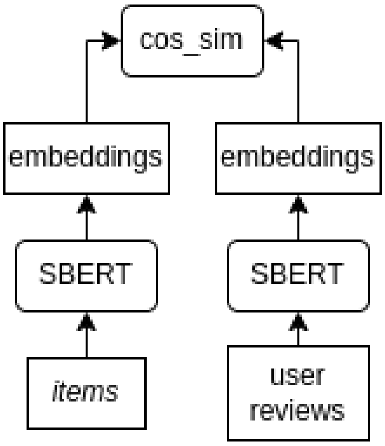


Figure 3.2. Two-Tower Model

(4) This approach (see Figure 3.2) uses the cosine similarity to compute similarities between indications/side effects (hereinafter: *items*) and user reviews. Before that, I remove well-known *items* from them to prevent the cosine similarity outputs *items* that are not new. The most similar *item* is returned as output for each user review.

(5, 6) The two-tower model approach computes similarities between each *item* and user reviews. This approach is not very flexible since the user review similarity could be highly influenced by any word in it. I use the attention mechanism to find more relations between words in user reviews. The zero/few-shot prompting approaches use the pre-defined prompt templates for each *item*.

(6) Since the few-shot approach needs examples, I provide the outputs from the two-tower model approach. I experiment with one, five, 10, and 30 examples.

(7) The model used for prompting is the open-sourced fine-tuned Llama 2[4] chat version of 7 milliard parameters from Microsoft. For inference, I use the vLLM[5], which effectively manages a KV cache memory via PagedAttention [14].

Ultimately, I need to evaluate the model outputs using the (5), (6), and (7) approaches. I define the loss function as:

$$\text{model}(\texttt{prompt}) = [\text{predicted}_1, \ldots, \text{predicted}_j]$$

$$L = -\frac{1}{t \cdot j} \sum_i \max\{\text{cos-sim}(\mathbf{e}_{\text{predicted}_i}, \mathbf{E}_{\text{well-known } items})\}$$

$$\text{where } (\mathbf{e}_{\text{predicted}_i})_{1\times 768} \text{ is an embedded string predicted}_i,$$
$$(\mathbf{E}_{\text{well-known } items})_{n\times 768} \text{ is a matrix of embeddings,}$$
$$\text{and } t \text{ is a dataset size}$$

[4] https://huggingface.co/meta-llama/Llama-2-7b-chat-hf

[5] https://docs.vllm.ai/en/latest

Chapter 4
Implementation

4.1 Web Scraping

Web scraping is a technique used to extract data from websites automatically. More specifically, it downloads a page in HTML format and extracts valuable data defined by the user.

4.1.1 Design of the Scraping Process

I have the molecules on which I am interested in data. For these molecules, I have to obtain URL addresses for both websites. Firstly, I tried automating this process by adjusting URL addresses like *www.drugs.com/mtm/<molecule>.html*. One problem with this approach is that the */mtm/* part does not always have to be the same. Sometimes, a molecule is located at */cdi/*, */cons/*, */npp/*, and that part does not even have to be there at all. Another problem is that some URL addresses also contain a type of drug application; for example, *formoterol* is located at */mtm/formoterol-inhalation.html*. The last problem is that a molecule could have different synonyms or is presented with another substance that helps with the pharmaceutical properties; for example, *aciclovir* is located at */acyclovir.html*, or *cetirizine* is located at */cetirizine-hcl.html* where *hcl* is a type of salt that helps with better absorption of the substance. Then, it sometimes happens that a molecule is located at multiple URL addresses but has a different amount of data. Unfortunately, a similar problem occurred with PatientsLikeMe. Ultimately, I was forced to manually find each molecule on the websites or check where more data was located.

If there is a molecule located at two URL addresses, and one of them is */<molecule>-<type of application>.html*, I choose the URL address without a type of application because I assume that it could contain side effects regarding a type of application other than a molecule itself. I also assume the identity of different types of one vitamin; for example, *vitamin D(2, 3) = alfacalcidol = colecalciferol = ergocalciferol*. As mentioned in the paragraph above, I assume the identity for molecules that are presented with a helper substance for better pharmaceutical properties or molecules that are presented in different forms; for example, *candesartan-cilexetil* is dissolving to *candesartan* in the body. See Appendix B for more

Both websites have accessible all user reviews only after login, and, in addition, login is not easy and often requires CSRF token generation, GDPR confirmation, and setting *httpOnly* cookies. So, I must simulate a web browser login during the initial scraping. PatientsLikeMe was consistently banning my accounts because I could not find any affable scraping delay. Therefore, I had to use different VPNs, which I was constantly switching.

The whole scraping is strictly for academic purposes as proof of concept.

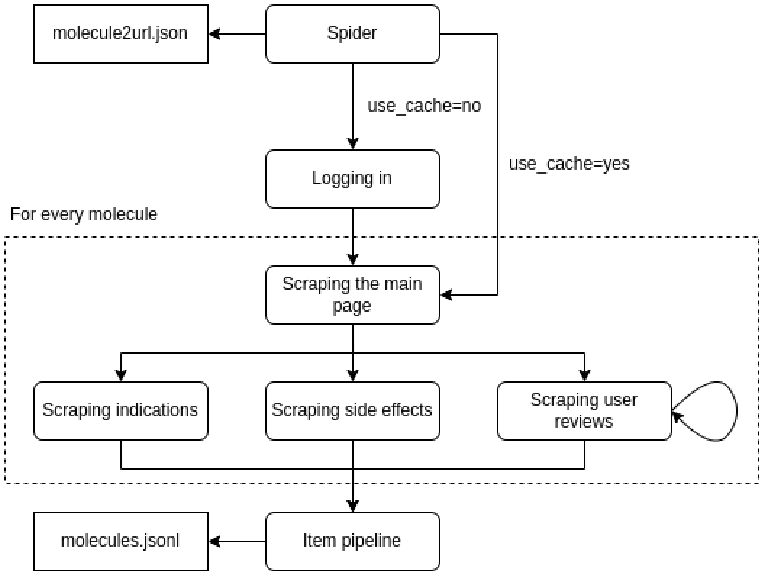
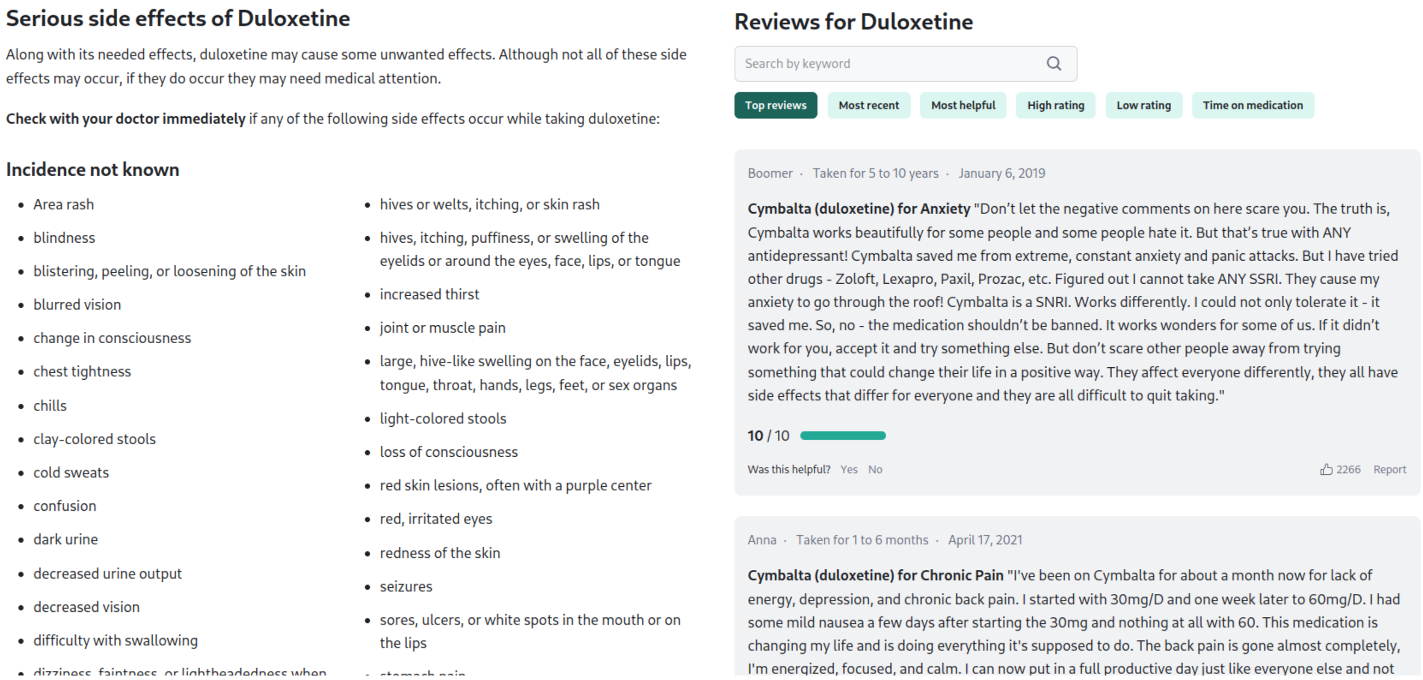


Figure 4.1. Design of the scraping process

The whole process looks like the Spider module reads the URL addresses from the file, if necessary, simulates logging in via web browser, and sends a request to each URL address. On the main page of a molecule, URL addresses for indications, side effects, and user reviews are found, and a request will be sent to them again. In the case of user review websites, this is done until the last page is reached. Every request is followed by selecting correct data from the HTML code and sending them to the Item pipeline module that processes data like lower-casing, removing non-ASCII characters, and so on.

4.1.2 Scraping Data from Drugs.com



Serious side effects of Duloxetine

Along with its needed effects, duloxetine may cause some unwanted effects. Although not all of these side effects may occur, if they do occur they may need medical attention.

Check with your doctor immediately if any of the following side effects occur while taking duloxetine:

Incidence not known

- Area rash
- blindness
- blistering, peeling, or loosening of the skin
- blurred vision
- change in consciousness
- chest tightness
- chills
- clay-colored stools
- cold sweats
- confusion
- dark urine
- decreased urine output
- decreased vision
- difficulty with swallowing
- dizziness, faintness, or lightheadedness when
- hives or welts, itching, or skin rash
- hives, itching, puffiness, or swelling of the eyelids or around the eyes, face, lips, or tongue
- increased thirst
- joint or muscle pain
- large, hive-like swelling on the face, eyelids, lips, tongue, throat, hands, legs, feet, or sex organs
- light-colored stools
- loss of consciousness
- red skin lesions, often with a purple center
- red, irritated eyes
- redness of the skin
- seizures
- sores, ulcers, or white spots in the mouth or on the lips
- stomach pain

Reviews for Duloxetine

Search by keyword

Top reviews | Most recent | Most helpful | High rating | Low rating | Time on medication

Boomer · Taken for 5 to 10 years · January 6, 2019

Cymbalta (duloxetine) for Anxiety "Don't let the negative comments on here scare you. The truth is, Cymbalta works beautifully for some people and some people hate it. But that's true with ANY antidepressant! Cymbalta saved me from extreme, constant anxiety and panic attacks. But I have tried other drugs - Zoloft, Lexapro, Paxil, Prozac, etc. Figured out I cannot take ANY SSRI. They cause my anxiety to go through the roof! Cymbalta is a SNRI. Works differently. I could not only tolerate it - it saved me. So, no - the medication shouldn't be banned. It works wonders for some of us. If it didn't work for you, accept it and try something else. But don't scare other people away from trying something that could change their life in a positive way. They affect everyone differently, they all have side effects that differ for everyone and they are all difficult to quit taking."

10 / 10

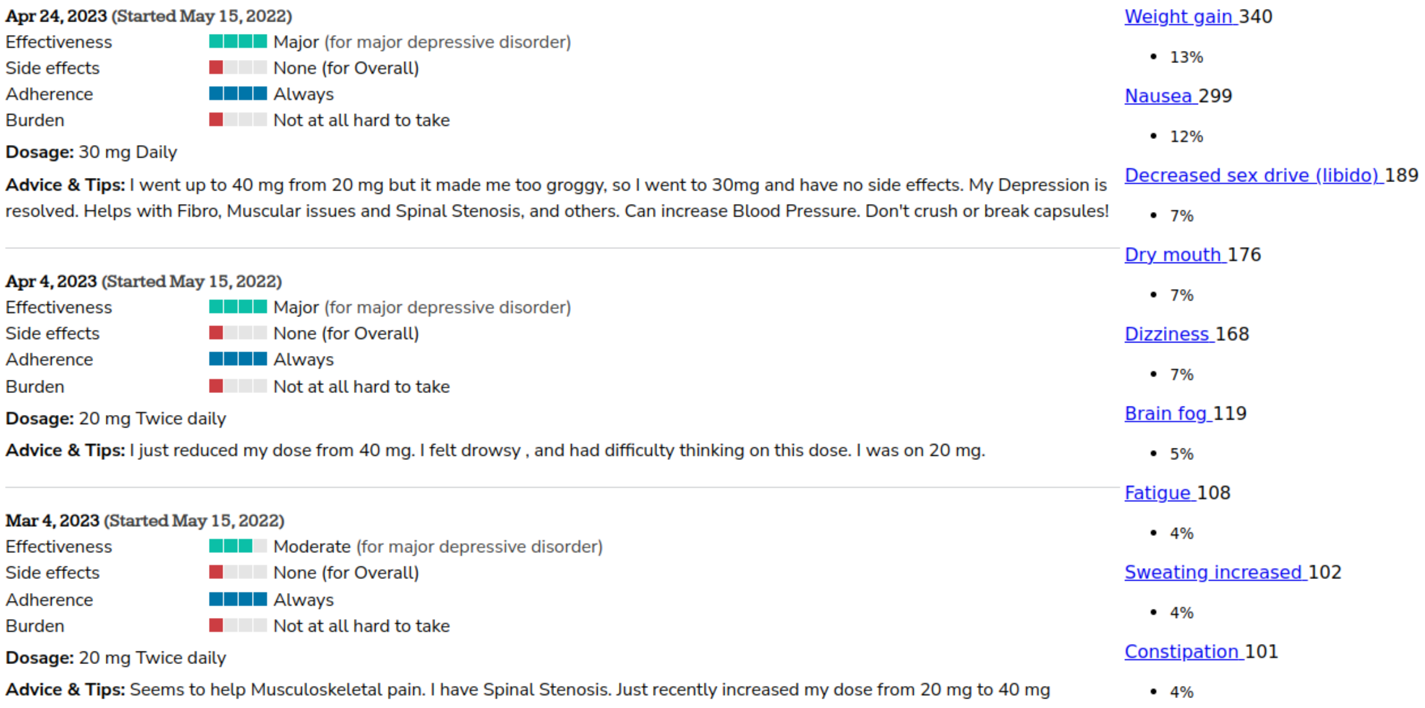
Was this helpful? Yes No 2266 Report

Anna · Taken for 1 to 6 months · April 17, 2021

Cymbalta (duloxetine) for Chronic Pain "I've been on Cymbalta for about a month now for lack of energy, depression, and chronic back pain. I started with 30mg/D and one week later to 60mg/D. I had some mild nausea a few days after starting the 30mg and nothing at all with 60. This medication is changing my life and is doing everything it's supposed to do. The back pain is gone almost completely, I'm energized, focused, and calm. I can now put in a full productive day just like everyone else and not

Figure 4.2. Side effects and user reviews of the molecule on Drugs.com. See Appendix C for more.

4.1.3 Scraping Data from PatientsLikeMe



Apr 24, 2023 (Started May 15, 2022)

Effectiveness	🟩🟩🟩🟩 Major (for major depressive disorder)
Side effects	🟥⬜⬜⬜ None (for Overall)
Adherence	🟦🟦🟦🟦 Always
Burden	🟥⬜⬜⬜ Not at all hard to take

Dosage: 30 mg Daily

Advice & Tips: I went up to 40 mg from 20 mg but it made me too groggy, so I went to 30mg and have no side effects. My Depression is resolved. Helps with Fibro, Muscular issues and Spinal Stenosis, and others. Can increase Blood Pressure. Don't crush or break capsules!

Apr 4, 2023 (Started May 15, 2022)

Effectiveness	🟩🟩🟩🟩 Major (for major depressive disorder)
Side effects	🟥⬜⬜⬜ None (for Overall)
Adherence	🟦🟦🟦🟦 Always
Burden	🟥⬜⬜⬜ Not at all hard to take

Dosage: 20 mg Twice daily

Advice & Tips: I just reduced my dose from 40 mg. I felt drowsy , and had difficulty thinking on this dose. I was on 20 mg.

Mar 4, 2023 (Started May 15, 2022)

Effectiveness	🟩🟩🟩⬜ Moderate (for major depressive disorder)
Side effects	🟥⬜⬜⬜ None (for Overall)
Adherence	🟦🟦🟦🟦 Always
Burden	🟥⬜⬜⬜ Not at all hard to take

Dosage: 20 mg Twice daily

Advice & Tips: Seems to help Musculoskeletal pain. I have Spinal Stenosis. Just recently increased my dose from 20 mg to 40 mg

Weight gain 340
- 13%

Nausea 299
- 12%

Decreased sex drive (libido) 189
- 7%

Dry mouth 176
- 7%

Dizziness 168
- 7%

Brain fog 119
- 5%

Fatigue 108
- 4%

Sweating increased 102
- 4%

Constipation 101
- 4%

Figure 4.3. User reviews and side effects of the molecule on PatientsLikeMe. See Appendix D for more.

4.1.4 Data processing

Data processing is handled by the Item pipeline module mentioned in Figure 4.1. It makes *items* lowercase, removes unnecessary white and non-ASCII characters, removes redundancy, and asserts checking if all user reviews have been scraped. Some user reviews on PatientsLikeMe may contain just *none* or *none.*, so I am ignoring those. Before a user inserts data into Drugs.com, there is a check from the pharmacist, so I assume *items* are correct there. This check does not happen on PatientsLikeMe, so it may contain wrong data in the molecule. I handle this situation with a naive approach, where I define the *threshold* for a number of reports per *item*:

$$threshold = \mathrm{ceil}(t \cdot \frac{p}{h})$$

where p is the number of users that reported any *item* for a given molecule; h is the number of *items*; $\frac{p}{h}$ is the average of reports of *items* per given molecule; t is fixed constant, in my case 5 %. The *threshold* is used as:

$$\text{number of reports per } item > threshold$$

molecule	indications	side_effects	user_reviews
abacavir-lamivudine	[hiv infection, …	[fruit-like breath odor, …	[my doctor and …
abiraterone	[prostate cancer]	[thickening of …	[Was diagnosed with …
…	…	…	…
zolpidem	[difficulty sleeping, …	[severe sunburn, …	[I have been …

Table 4.1. Merged data of both scraping

4.2 Visualisation of Embeddings

The SBERT model transforms each string to a vector $v \in \mathbb{R}^{768}$. The resulting matrix for each text set is $\mathrm{M}_{n \times 768}$, where n is the number of elements in a text set.

Now, I visualize vectors for indications and side effects to find out if there are any clusters between them, which I might use for another phase of removing redundancy and speeding up calculations. I already handle redundancy in Section 4.1.4, but that is only a naive approach in the sense of `string1 == string2`, where strings like *muscle spasm* and *muscle spasms* may escape filtering. I use the t-SNE algorithm for visualization.

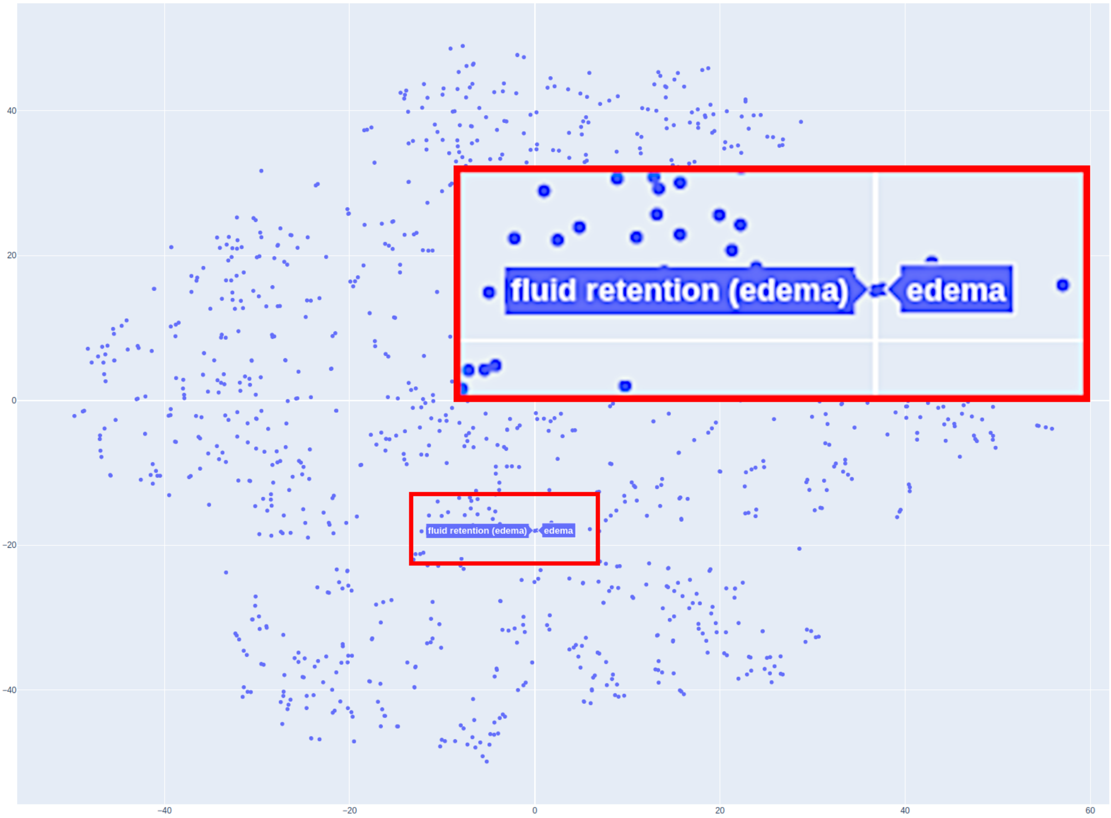


Figure 4.4. Example of visualization of indications and possible clusters using interactive Python tool Plotly[1]

The visualizations show possible hints of clusters. I am going to do clustering and look into them.

4.3 Clustering

As mentioned in Chapter 3, I use the function `community_detection` with predefined *threshold* = 0.75 that makes clusters with neighbors having similarity greater than *threshold* among each one. The function returns `List[List[int]]`, where the external list contains all clusters, and the internal list contains elements representing indices of row vectors from an embedding matrix.

[1] https://plotly.com/python

There are meaningful clusters, but there are also clusters that are poorly done; for example:

```
kidney transplant
liver transplant
heart transplant
lung transplant
```

This could obviously be a problem because each transplant is a wholly different process. In this case, I would take *kidney transplant* as the name of the cluster since it is the central point of the cluster. Then, it could happen that I would match *kidney transplant* as the potentially new indication for some user reviews. The catch is that I do not know what specific transplant a user experienced.

Final results are always human-checked with respect to well-known *items*, but those glitches should be handled better.

Ultimately, I add those clusters to proper rows in data mentioned in Table 4.1, and I create a mapping from *item* to its central point in the cluster and embed those central points the same way as in Section 4.2.

4.4 Finding New Indications and Side Effects

From now on, I will only work with already clustered *items*. Firstly, the matrix $\mathbf{E}_{items}$ contains vectors of all embedded *items*. Secondly, the matrix $\mathbf{E}_{items\text{ per molecule}}$ contains vectors of embedded *items* only per given molecule. A similar principle holds for other uses.

4.4.1 Two-Tower Model

Before showing the pseudo-code for finding out new *items*, let me explain one used operation on the third line:

$$\mathbf{E}_{items} \oslash \mathbf{E}_{items\text{ per molecule}} = \mathbf{R}$$

$$\mathbf{R}[i,:] = \mathbf{E}_{items}[j,:] \text{ if } (\mathbf{E}_{items}[j,:] \notin \mathbf{E}_{items\text{ per molecule}} \text{ and } \mathbf{E}_{items}[j,:] \notin \mathbf{R})$$

$$\mathbf{E}[i,:] \in \mathbf{E} = \text{true if the } i\text{-th row of the matrix } \mathbf{E} \text{ is in the matrix } \mathbf{E}$$

```
1  $matches = [\,]$
2  for $molecule$, $user\_reviews$, $items$ in $data$ do
3      $\mathbf{R} = \mathbf{E}_{items} \oslash \mathbf{E}_{items\text{ per molecule}}$
4      $similarities = \text{cos-sim}(\mathbf{R}, \mathbf{E}_{\text{user reviews per molecule}})$
5      for $i$, $row$ in enumerate($similarities$) do
6          $most\_similar\_item = \max(row)$
7          $matches$.append({
8              $molecule$,
9              "user_review": $user\_reviews[i]$,
10             "similarity": $most\_similar\_item.similarity$,
11             "most_similar_item": $most\_similar\_item.name$,
12             "well_known_items": $items$
13         })
14     end for
15 end for
```

In Section 4.3, I mention some clusters that could be a problem. Another example is the cluster containing *side effects*. I would match a user review containing the string *side effects* with high similarity. Because of that, I ignore that clustered *items* when appending to the *matches*.

The example from the output, more specifically the one item from `matches`, looks like:

```
{'molecule': 'memantine',
 'user_review': 'increased anxiety',
 'similarity': 1.000000238418579,
 'most_similar_item': 'increased anxiety',
 'well_known_items': ['new daily persistent headache', 'memory loss',
 'corticobasal degeneration', 'progressive supranuclear palsy',
 'amyotrophic lateral sclerosis', 'bipolar disorder', 'brain fog',
 'involuntary eye movement (nystagmus)', 'muscle stiffness', 'autism',
 'central pain syndrome', 'pain', 'early onset dementia', 'migraine',
 'short term memory problems', 'cognitive impairment', 'fibromyalgia',
 'depressed mood', 'obsessive-compulsive disorder', 'memory problems',
 'emotional lability']}
```

The cosine similarity approach looks for similarities between each item and a whole sentence, not the individual word in the sentence. If a user review is a slightly more complex sentence that does not directly mention a similar item, this approach will not find it. However, the attention mechanism could help me to do that.

4.4.2 Zero-Shot Prompting

When designing the prompt template, I follow the correct formatting that was used when fine-tuning[2] the model. This is what the prompt looks like:

```
<s>[INST] <<SYS>>\n
You are an assistant helping extract indications from the given text.
Return Python list. When no indications are extracted,
return an empty list.\n
<</SYS>>
\n
\n
<user review> [/INST] ['
```

I deliberately use `['` at the end of the prompt so that the model can better respond in a Python list format and I can parse it easily. If that part was not there, the model could occasionally return a Python list, bullet points, or even an answer in one long sentence.

Because I have over 100 thousand user reviews, I select only 10 user reviews per molecule (making it two thousand in total). Moreover, the model has the context window for 4096 tokens. Therefore before generating, I tokenize all prompts and filter out those whose length is greater than 4096. Now, I let the model generate an output for each user review.

I experiment with several settings for `temperature` and `top_p` values; the ones I use are `temperature = 0.7` and `top_p = 0.9`. Both values influence the creativity of the model: $\text{softmax}(\frac{z}{\text{temperature}})$; `top_p` takes into account only tokens where their

[2] https://llama.meta.com/docs/model-cards-and-prompt-formats/meta-llama-2

cumulative probability is greater (tokens are sorted by their probability in descending order). I also use the stop string] because the model may still generate after the end of a Python list.

Here is the sample from the output:

```
<s>[INST] <<SYS>>\n
You are an assistant helping extract indications from the given text.
Return Python list. When no indications are extracted,
return an empty list.\n
<</SYS>>
\n
\n
It usually gives me a bloated feeling so I take it with a full stomach
[/INST] ['indications': ['bloated feeling', 'full stomach']
```

The model still has problems with following the correct format, but it will be improved in the few-shot prompting. Unfortunately, the model returns nonsense as well:

```
I have had lots of energy and felt good until recently. My PSA initially
went from 66 to 0.1, after 20 months it has started to rise, now it is
0.8 The medicine was easy to take. I did have a decrease in bone
thickness. My bones have now shown and increase in mets.

['Indications']
```

4.4.3 Few-Shot Prompting

This approach can be used to enable in-context learning. It provides a few examples of the task at inference time. It has been shown that it could dramatically improve the performance [15].

As before, I have to prepare the prompt template:

```
<s>[INST] <<SYS>>\n
You are an assistant helping extract indications from the given text.
Return Python list. When no indications are extracted,
return an empty list.\n
<</SYS>>
\n
\n
<user review> [/INST] <output> </s>
<s>[INST] <user review> [/INST] <output> </s>
...
<s>[INST] <user review> [/INST] ['
```

For the examples, I select random user reviews from a different dataset than those used for prompting. As an output for each user review, I use data from the *matches* in Section 4.4.1. I modify the algorithm to return the three most similar *items*. An output has a randomized length to let the model know it may return different list lengths. I experiment with 1/5/10/30 examples.

```
I have had lots of energy and felt good until recently. My PSA initially
went from 66 to 0.1, after 20 months it has started to rise, now it is
0.8 The medicine was easy to take. I did have a decrease in bone
thickness. My bones have now shown and increase in mets.

n=1 ['PSA', 'prostate cancer', 'bone metastases']
n=5/10 ['prostate cancer', 'hormone therapy', 'bone metastases']
```

The model tends to output more sensical things, but they are still sometimes not related to user reviews.

4.4.4 Supervised Fine-Tuning

I fine-tune the model on the NVIDIA A100-SXM4-80GB in the RCI[3] cluster using the LoRA for one epoch. The targeted modules are matrices W_{Q} and W_{V} (see Section 2.1.1). I set $r = 8$ and $alpha = 16$ since the scaling factor of LoRA's weights added back to the original ones is $\frac{alpha}{r}$ and $dropout = 0.05$ that indicates the dropout probability for LoRA layers. Because some LLM models lack padding tokens, I manually set the padding token to `</s>` and use padding from the left side. The dataset used for the fine-tuning consists of over seven thousand user reviews that do not occur in the dataset used for prompting. The learning rate is $5e-5$, and as the optimization method AdamW[4], has been chosen. The model has about seven milliard tokens, and because of the LoRA, I fine-tune only four million of them (0.06 %).

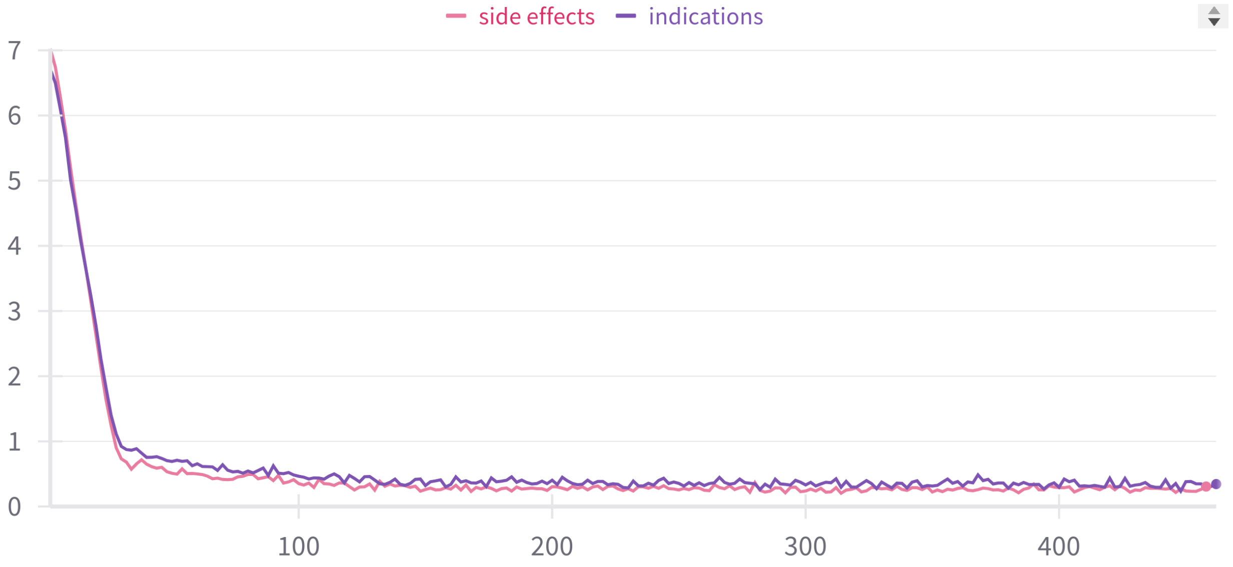


Figure 4.5. Training loss

[3] https://rci.cvut.cz

[4] https://pytorch.org/docs/stable/generated/torch.optim.AdamW.html

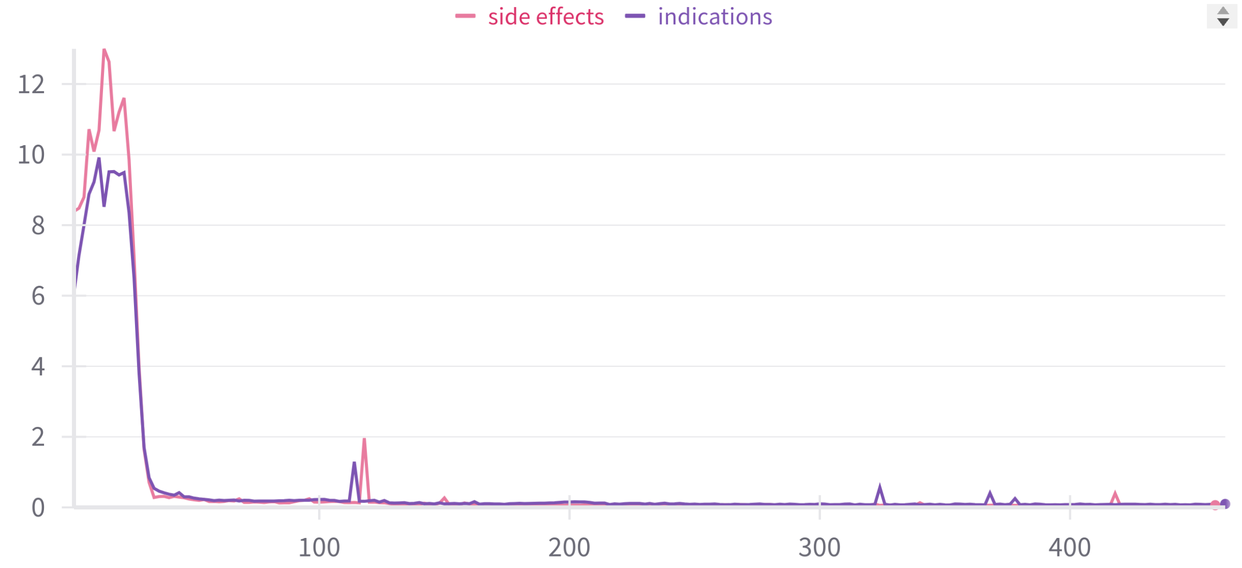


Figure 4.6. Gradient norm

I report the fine-tuning process to the Weights & Biases[5] platform, which shows me important data like: a training loss, a gradient norm, or even the size of allocated memory on the graphics card and the time spent accessing memory. These first two are especially important in analyzing the fine-tuning and its stability.

The fine-tuning is done for both *items*: indications and side effects. Ultimately, I perform the zero/few-shot prompting again.

[5] https://wandb.ai

Chapter 5
Evaluation

I evaluate the cosine similarity approach (see Section 4.4.1), the zero/few-shot prompting (see Sections 4.4.2 and 4.4.3) on the base model, and then again on the fine-tuned model (see Section 4.4.4). The dataset used for the evaluation is the same as in Section 4.4.2, which is the different one used for the fine-tuning to prevent the fine-tuned model from learning the correct results by heart. The loss function is defined in Chapter 3.

indications	-0.411
side effects	-0.541

Table 5.1. Evaluation of the Two-Tower Model

Generated outputs are manually fixed when there are no square brackets or apostrophes in a leading or trailing part. Each NLP approach was performed thrice due to the volatility rising from the pre-defined `temperature` and `top_p` values; the resulting loss value is the average.

	base model				fine-tuned model			
	attempts				attempts			
	0	1	2	mean	0	1	2	mean
zero-shot	-0.362	-0.363	-0.362	**-0.362**	-0.431	-0.428	-0.430	**-0.430**
few-shot[n=1]	-0.481	-0.447	-0.448	-0.459	-0.450	-0.449	-0.449	-0.449
few-shot[n=5]	-0.437	-0.437	-0.435	-0.436	-0.541	-0.540	-0.538	-0.540
few-shot[n=10]	-0.494	-0.498	-0.490	-0.494	-0.483	-0.482	-0.481	-0.482
few-shot[n=30]	-0.481	-0.474	-0.478	-0.478	-0.473	-0.477	-0.477	-0.476

Table 5.2. Evaluation of all NLP approaches for indications

As shown in Tables 5.2 and 5.3, only the zero-shot approach for the base model overcomes the two-tower model losses in Table 5.1. The few-shot approach is doing worse, and both prompting approaches for the fine-tuned model are even worse when compared to the base model. I discuss possible hypotheses and examples from the approaches below.

	base model				fine-tuned model			
	attempts				attempts			
	0	1	2	mean	0	1	2	mean
zero-shot	-0.478	-0.481	-0.475	**-0.478**	-0.558	-0.555	-0.555	**-0.556**
few-shot[n=1]	-0.541	-0.537	-0.536	-0.538	-0.612	-0.610	-0.612	-0.611
few-shot[n=5]	-0.591	-0.588	-0.591	-0.590	-0.592	-0.593	-0.594	-0.593
few-shot[n=10]	-0.597	-0.596	-0.588	-0.594	-0.616	-0.618	-0.619	-0.618
few-shot[n=30]	-0.567	-0.573	-0.573	-0.571	-0.653	-0.654	-0.657	-0.655

Table 5.3. Evaluation of all NLP approaches for side effects

Interestingly, the fine-tuned model is performing worse than the base model. I looked into the generated outputs, and it seems like the model has learned a lot about using correct *items*'s names from the clusters. However, there are still many imperfections in finding relations between them and user reviews. Overall, only the zero-shot prompting, where the model has more flexibility than using the few-shot prompting, overcomes the two-tower model approach. It may be due to insufficient fine-tuning or low model flexibility. When comparing the losses from both tables, it is clear that prompting for side effects is worse than prompting for indications. The only explanation I can think of is that the model has difficulty telling the difference between a symptom and a root cause.

Here are the good and bad examples of the fine-tuned model with 30 examples:

```
Looks like it is doing what it is supposed to do...
numbers getting better and better.
predictions=['blood pressure (hypertension)']
```

```
I'm unsure of weight gain however appetite definitely improved. Nausea
in morning is gone. Energy in mornings better - could be current steroid
use. Overall joint pain appears to have decreased. Side effects are
above, however App wouldn't let me add them.

well_known_items=['persistent depressive disorder', 'fibromyalgia',
'anxiety attacks', 'depressed mood']
predictions=['joint pain']
```

Chapter 6
Conclusion

This thesis consists of four parts: web scraping, creating embeddings, visualization/clustering, and finding novel *items*: indications and side effects.

I have been given the molecules (circa 250) by my supervisor. Each molecule needs to be matched with the corresponding molecule on both websites. The problem arose when I discovered that the molecule names were unambiguous, like different synonyms or the molecules could be bound with the helper molecules, which helps with better pharmaceutical properties. I had to go through each molecule manually to resolve these differences (see Appendix B). A web browser was simulated using the scrapy-selenium package, which contains a bug because it assumes an older version of the Selenium framework. I have partially fixed this bug. Drugs.com's format on side effects pages is inconsistent, so I manually found all possible formats and adapted my selector for the scraping process. The last problem was that both websites could have more pages for one molecule, but those pages do not contain the same amount of data. I went through all the pages I found and selected ones that had more data.

For creating embeddings, I studied how the SBERT model works and chose the suitable model according to benchmarks on the SentenceTransformers.

I have chosen the appropriate algorithm (t-SNE) to visualize embeddings in high-dimensional space. SentenceTransformers also provides the function of finding clusters between embeddings with the pre-defined threshold. Unfortunately, I found later that the model was not appropriately adapted to the specific domain and created imperfect embeddings for *items* (see Section 4.3), leaving insufficient time to change or improve the model.

I introduced two NLP approaches for finding novel *items*: the cosine similarity and generative approaches (the zero/few-shot prompting and the fine-tuned model). The first approach uses embeddings and computes the cosine similarity between them. It is the naive approach since the whole sentence is embedded into a single vector like *item*; it may often happen that if a user mentions something that does not relate to the molecule but relates to the *item*, it will be matched with a high similarity value. Generative approaches are better since they use the attention mechanism, which considers relations between individual words in a user review and has a chance to filter out meaningless mentions. For this approach, I studied how the Transformer architecture and the fine-tuning using the LoRA works or how I am supposed to format a prompt template. The zero-shot prompting gives the model the user review and the instruction on what to do with it. The few-shot prompting differs in that I add a few examples with the instruction. The examples were obtained from the cosine similarity approach. Ultimately, I fine-tuned the model to adapt it more to the specified task and did the prompting again. The fine-tuned model did worse overall than the bare model, which I mention in Chapter 5 with the possible hypothesis.

In the Evaluation, I have shown that the generative approach has the potential to be better than the cosine similarity approach.

6.1 Possible improvements

- Collaboration with a domain expert.
- Although, in this thesis, the whole dataset is not used for fine-tuning, many websites may still be web scraped; for example, the DailyStrength[1].
- The SBERT model should be more adapted to the domain. Created clusters are not precise. Therefore, merging *items* unrelated to each other creates wrong data the model is prompted and fine-tuned on.
- Since the fine-tuning has not helped, many experiments may still be done such as using a pre-trained model on the domain like the BioBERT[2].
- A sort of summarization layer may be used to get the key points from user reviews. Afterward, summarized results will be prompted.
- Better evaluation on choosing:
  - `temperature` and `top_p` hyperparameters for the model generation.
  - `alpha` and `r` since they scale the LoRA's weights merged to the original ones.

[1] https://www.dailystrength.org

[2] https://arxiv.org/pdf/1901.08746

Appendix A
Glossary

BERT	Bidirectional Encoder Representations from Transformers
LLM	Large Language Mode
LoRA	Low-Rank Adaptation of Large Language Models
NLI	Natural Language Inference
PEFT	Parameter-Efficient Fine-Tuning
SBERT	Sentence-BERT
STS	Sentence Text Similarity
t-SNE	t-distributed stochastic neighbor embedding

References

[1] Michael Schlander, Karla Hernandez-Villafuerte, Chih-Yuan Cheng, Jorge Mestre-Ferrandiz, and Michael Baumann. *How Much Does It Cost to Research and Develop a New Drug? A Systematic Review and Assessment*. 2021. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8516790/.

[2] BIO, QLS Advisors, and Informa UK Ltd. *Clinical Development Success Rates and Contributing Factors 2011-2020*. 2021. https://go.bio.org/rs/490-EHZ-999/images/ClinicalDevelopmentSuccessRates2011_2020.pdf.

[3] Robert M. Plenge, Edward M. Scolnick, and David Altshuler. *Validating therapeutic targets through human genetics*. 2013. https://pubmed.ncbi.nlm.nih.gov/23868113/.

[4] Irwin Goldstein, Arthur L. Burnett, Raymond C. Rosen, Peter W. Park, and Vera J. Stecher. *The Serendipitous Story of Sildenafil: An Unexpected Oral Therapy for Erectile Dysfunction*. 2019. https://pubmed.ncbi.nlm.nih.gov/30301707/.

[5] WHO. *Repurposing of medicines - the underrated champion of sustainable innovation: policy brief*. 2021. https://www.who.int/europe/publications/i/item/WHO-EURO-2021-2807-42565-59178.

[6] Nicholas S. Downing, Nilay D. Shah, Jenerius A. Aminawung, and et al . *Postmarket Safety Events Among Novel Therapeutics Approved by the US Food and Drug Administration Between 2001 and 2010*. 2017. https://jamanetwork.com/journals/jama/fullarticle/2625319.

[7] FDA. *FDA Statement on the Voluntary Withdrawal of Raptiva From the U.S. Market*. 2009. https://www.fda.gov/drugs/postmarket-drug-safety-information-patients-and-providers/fda-statement-voluntary-withdrawal-raptiva-us-market.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, and et al . *Attention Is All You Need*. 2017. https://arxiv.org/pdf/1706.03762.

[9] Nils Reimers, and Iryna Gurevych. *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*. 2019. https://arxiv.org/pdf/1908.10084.

[10] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, and et al . *Llama 2: Open Foundation and Fine-Tuned Chat Models*. 2023. https://arxiv.org/pdf/2307.09288.

[11] Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, and et al . *LoRA: Low-Rank Adaptation of Large Language Models*. 2021. https://arxiv.org/pdf/2106.09685.

[12] Alan (Lan) R. Aronson. *UMLS Webcast: The Currect State of MetaMap and MMTx*. 2009.
https://lhncbc.nlm.nih.gov/ii/information/Papers/09.08.20.MetaMap-MMTx.updated.pdf.

[13] Monica Agrawal, Stefan Hegselmann, and Hunter Lang. *Large Language Models are Few-Shot Clinical Information Extractors*. 2022.
https://arxiv.org/pdf/2205.12689.

[14] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, and et al . *Efficient Memory Management for Large Language Model Serving with PagedAttention*. 2023.
https://arxiv.org/abs/2309.06180.

[15] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, and et al . *Language Models are Few-Shot Learners*. 2020.
https://arxiv.org/pdf/2005.14165.

Appendix B
Assumptions about molecules

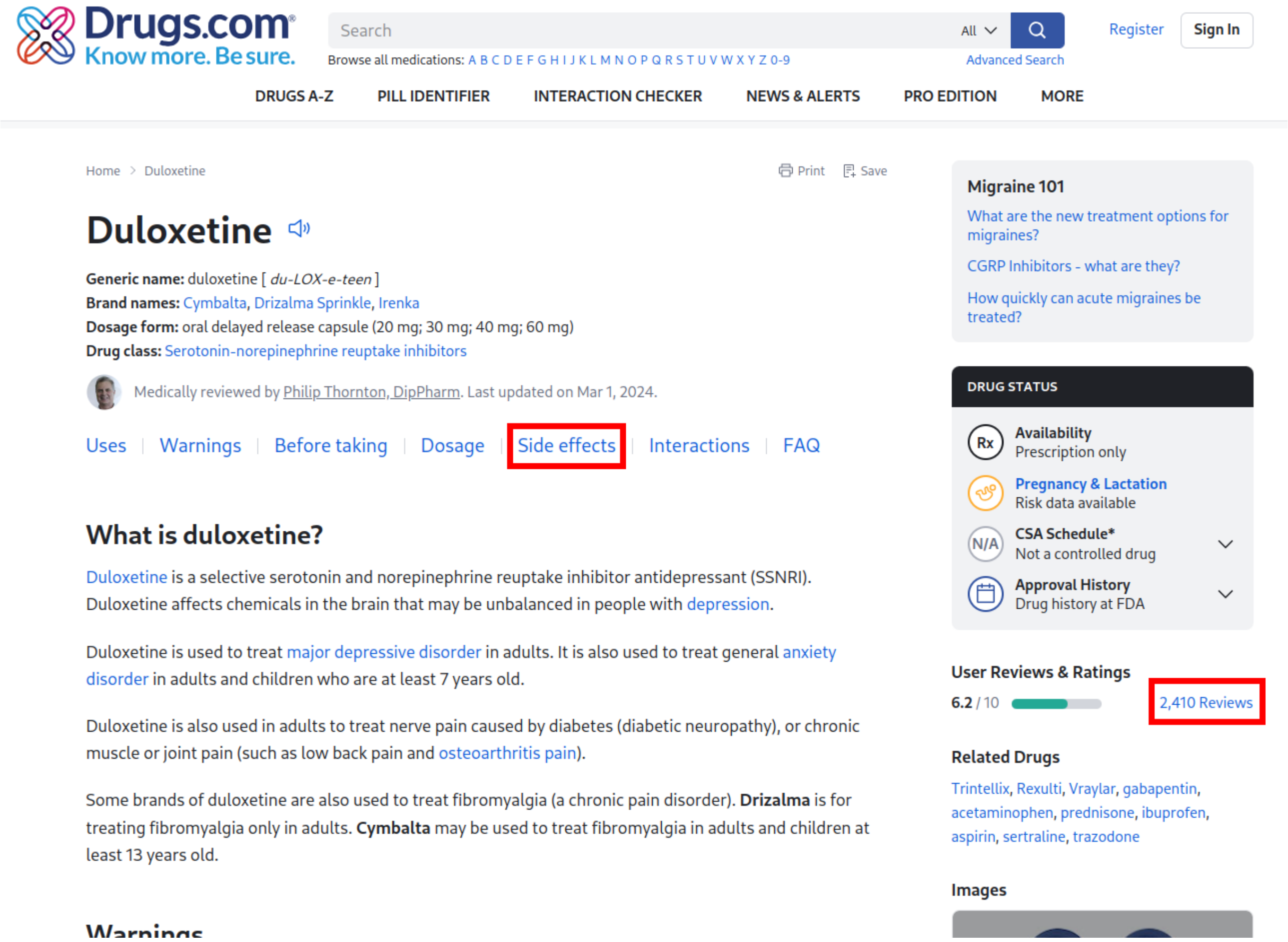
I made all assumptions/observations below on my own. It must be added that I am not an expert in this domain, so it may contain inaccuracies or even misjudgments.

bortezomib and *bortezomib-mannitol* are the same because mannitol is a type of sugar that improves pharmaceutic properties (for example, better drug solubility and stability). A similar principle holds for *losartan* and *losartan-potassium* or *iloprost* and *iloprost-trometamol.candesartan* and *candesartan-cilexetil* are the same because *candesartan-cilexetil* is the form of molecule, which is disolving to *candersartan* in the body. A similar principle holds for *mometasone* and *mometasone furoate* or *mycophenolate* and *mycophenolate mofetil* or *olmesartan* and *olmesartan-medoxomil* or *pemetrexed* and *pemetrexed-diarginine* or *perindopril-erbumine-indapamide* and *perindopril-indapamide* or *solifenacin* and *solifenacin succinate* or *sorafenib* and *sorafenib tosylate* or *sumatriptan* and *sumatriptan succinate* or *sunitinib* and *sunitinib malate* or *tenofovir* and *tenofovir-disoproxil* or *ulipristal* and *ulipristal acetate.* I usually use the shortest name for the molecule while preferring the more popular one.

Some molecules; for example, *amlodipine-indapamide-perindopril* work similarly to *amlodipine-perindopril*, but I do not assume the identity because *indapamide* is the active substance, not the helper for better pharmaceutic properties. A similar principle holds for; for example, *desogestrel* and *desogestrel-ethinyl estradiol* or *dienogest* and *dienogest-estradiol.*

A scraping of side effects has a higher recall than a precision. Some side effects might contain withdrawal symptoms or overdose symptoms as well. I also do not distinguish between side effects when the molecule is combined with statins (used for better synergic effect).

Appendix C
Scraping Data from Drugs.com



Duloxetine

Generic name: duloxetine [*du-LOX-e-teen*]
Brand names: Cymbalta, Drizalma Sprinkle, Irenka
Dosage form: oral delayed release capsule (20 mg; 30 mg; 40 mg; 60 mg)
Drug class: Serotonin-norepinephrine reuptake inhibitors

Medically reviewed by Philip Thornton, DipPharm. Last updated on Mar 1, 2024.

Uses | Warnings | Before taking | Dosage | Side effects | Interactions | FAQ

What is duloxetine?

Duloxetine is a selective serotonin and norepinephrine reuptake inhibitor antidepressant (SSNRI). Duloxetine affects chemicals in the brain that may be unbalanced in people with depression.

Duloxetine is used to treat major depressive disorder in adults. It is also used to treat general anxiety disorder in adults and children who are at least 7 years old.

Duloxetine is also used in adults to treat nerve pain caused by diabetes (diabetic neuropathy), or chronic muscle or joint pain (such as low back pain and osteoarthritis pain).

Some brands of duloxetine are also used to treat fibromyalgia (a chronic pain disorder). **Drizalma** is for treating fibromyalgia only in adults. **Cymbalta** may be used to treat fibromyalgia in adults and children at least 13 years old.

Warnings

Migraine 101

What are the new treatment options for migraines?

CGRP Inhibitors - what are they?

How quickly can acute migraines be treated?

DRUG STATUS

Availability
Prescription only

Pregnancy & Lactation
Risk data available

CSA Schedule*
Not a controlled drug

Approval History
Drug history at FDA

User Reviews & Ratings

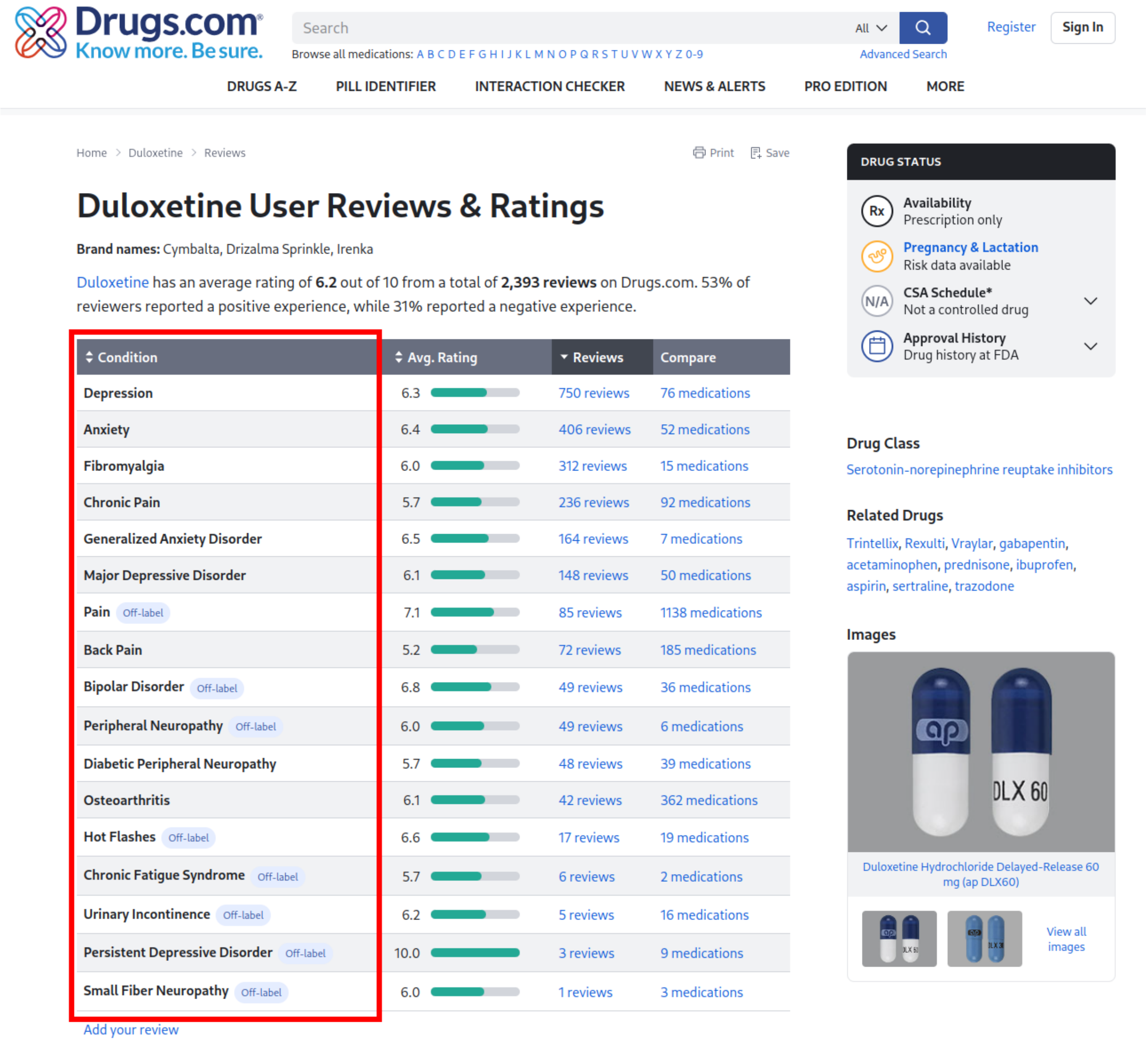
6.2 / 10 2,410 Reviews

Related Drugs

Trintellix, Rexulti, Vraylar, gabapentin, acetaminophen, prednisone, ibuprofen, aspirin, sertraline, trazodone

Images

Figure C.1. Example page of one molecule on Drugs.com



Duloxetine User Reviews & Ratings

Brand names: Cymbalta, Drizalma Sprinkle, Irenka

Duloxetine has an average rating of **6.2** out of 10 from a total of **2,393 reviews** on Drugs.com. 53% of reviewers reported a positive experience, while 31% reported a negative experience.

Condition	Avg. Rating	Reviews	Compare
Depression	6.3	750 reviews	76 medications
Anxiety	6.4	406 reviews	52 medications
Fibromyalgia	6.0	312 reviews	15 medications
Chronic Pain	5.7	236 reviews	92 medications
Generalized Anxiety Disorder	6.5	164 reviews	7 medications
Major Depressive Disorder	6.1	148 reviews	50 medications
Pain Off-label	7.1	85 reviews	1138 medications
Back Pain	5.2	72 reviews	185 medications
Bipolar Disorder Off-label	6.8	49 reviews	36 medications
Peripheral Neuropathy Off-label	6.0	49 reviews	6 medications
Diabetic Peripheral Neuropathy	5.7	48 reviews	39 medications
Osteoarthritis	6.1	42 reviews	362 medications
Hot Flashes Off-label	6.6	17 reviews	19 medications
Chronic Fatigue Syndrome Off-label	5.7	6 reviews	2 medications
Urinary Incontinence Off-label	6.2	5 reviews	16 medications
Persistent Depressive Disorder Off-label	10.0	3 reviews	9 medications
Small Fiber Neuropathy Off-label	6.0	1 reviews	3 medications

Add your review

DRUG STATUS

Availability
Prescription only

Pregnancy & Lactation
Risk data available

CSA Schedule*
Not a controlled drug

Approval History
Drug history at FDA

Drug Class

Serotonin-norepinephrine reuptake inhibitors

Related Drugs

Trintellix, Rexulti, Vraylar, gabapentin, acetaminophen, prednisone, ibuprofen, aspirin, sertraline, trazodone

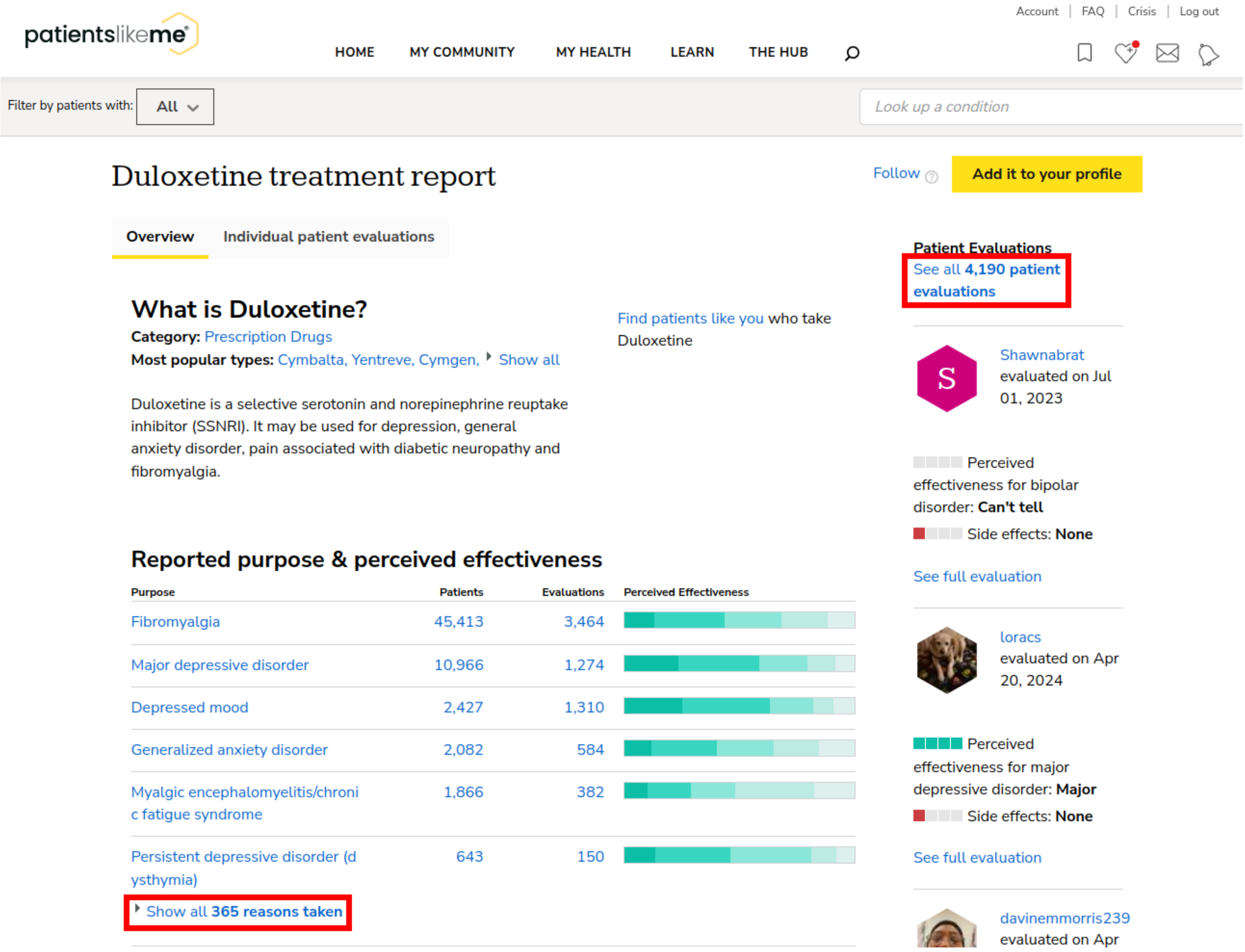
Images

Duloxetine Hydrochloride Delayed-Release 60 mg (ap DLX60)

View all images

Figure C.2. Indications of one molecule on Drugs.com

Appendix D
Scraping Data from PatientsLikeMe



Account | FAQ | Crisis | Log out

Filter by patients with: All

Look up a condition

Duloxetine treatment report

Follow

Add it to your profile

Overview | Individual patient evaluations

What is Duloxetine?

Category: Prescription Drugs

Most popular types: Cymbalta, Yentreve, Cymgen, ‣ Show all

Duloxetine is a selective serotonin and norepinephrine reuptake inhibitor (SSNRI). It may be used for depression, general anxiety disorder, pain associated with diabetic neuropathy and fibromyalgia.

Find patients like you who take Duloxetine

Reported purpose & perceived effectiveness

Purpose	Patients	Evaluations	Perceived Effectiveness
Fibromyalgia	45,413	3,464	
Major depressive disorder	10,966	1,274	
Depressed mood	2,427	1,310	
Generalized anxiety disorder	2,082	584	
Myalgic encephalomyelitis/chronic fatigue syndrome	1,866	382	
Persistent depressive disorder (dysthymia)	643	150	

‣ Show all **365 reasons taken**

Patient Evaluations

See all **4,190 patient evaluations**

Shawnabrat evaluated on Jul 01, 2023

Perceived effectiveness for bipolar disorder: **Can't tell**

Side effects: **None**

See full evaluation

loracs evaluated on Apr 20, 2024

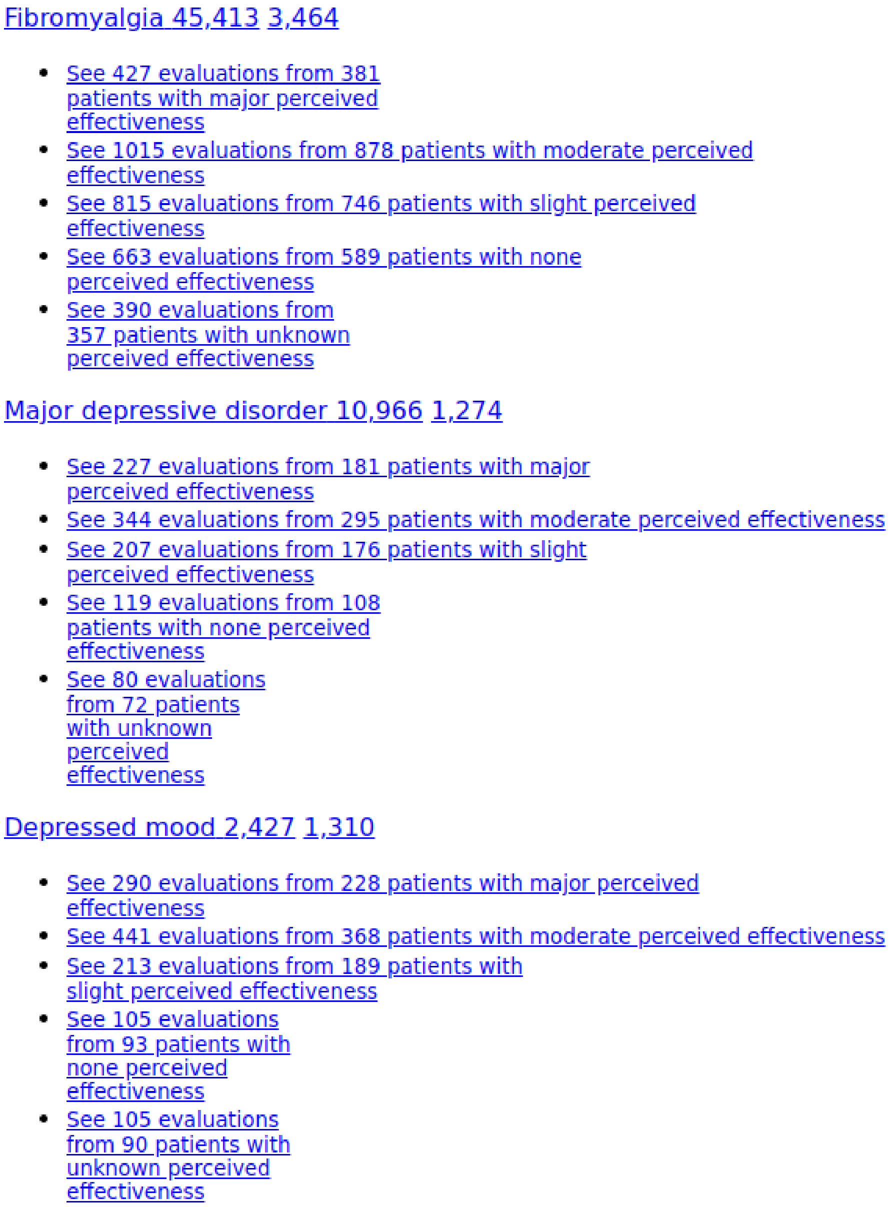
Perceived effectiveness for major depressive disorder: **Major**

Side effects: **None**

See full evaluation

davinemmorris239 evaluated on Apr

Figure D.3. Example page of one molecule on PatientsLikeMe



Fibromyalgia 45,413 3,464

- See 427 evaluations from 381 patients with major perceived effectiveness
- See 1015 evaluations from 878 patients with moderate perceived effectiveness
- See 815 evaluations from 746 patients with slight perceived effectiveness
- See 663 evaluations from 589 patients with none perceived effectiveness
- See 390 evaluations from 357 patients with unknown perceived effectiveness

Major depressive disorder 10,966 1,274

- See 227 evaluations from 181 patients with major perceived effectiveness
- See 344 evaluations from 295 patients with moderate perceived effectiveness
- See 207 evaluations from 176 patients with slight perceived effectiveness
- See 119 evaluations from 108 patients with none perceived effectiveness
- See 80 evaluations from 72 patients with unknown perceived effectiveness

Depressed mood 2,427 1,310

- See 290 evaluations from 228 patients with major perceived effectiveness
- See 441 evaluations from 368 patients with moderate perceived effectiveness
- See 213 evaluations from 189 patients with slight perceived effectiveness
- See 105 evaluations from 93 patients with none perceived effectiveness
- See 105 evaluations from 90 patients with unknown perceived effectiveness

Figure D.4. Indications of one molecule on PatientsLikeMe

Appendix E
Code

The code used for this thesis is in attachments.