UKRAINIAN CATHOLIC UNIVERSITY

BACHELOR THESIS

Knowledge profiling for personalized language learning

Author:
Anton TARASOV

Supervisor:
Oles DOBOSEVYCH

A thesis submitted in fulfillment of the requirements for the degree of Bachelor of Science

in the

Department of Computer Sciences
Faculty of Applied Sciences

Lviv 2021

Declaration of Authorship

I, Anton TARASOV, declare that this thesis titled, "Knowledge profiling for personalized language learning" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

"In the beginning was the Word, and the Word was with God, and the Word was God."

John 1:1

UKRAINIAN CATHOLIC UNIVERSITY

Faculty of Applied Sciences

Bachelor of Science

Knowledge profiling for personalized language learning

by Anton TARASOV

Abstract

There are 1.5 billion English-language learners worldwide. More and more of them use some digital tools and media to improve their skills. Online learning platforms show competitiveness compared to offline lessons. This work proposes a framework for creating a language learning platform for profiling user knowledge and providing personalized study materials. We describe the developed minimal viable product for vocabulary and tenses learning. We planned to test the solution through private tutoring, and, for now, we evaluated the recommendation system for prioritization of learned vocabulary on collected external data. Code of our MVP is stored in the GitHub repository.

Acknowledgements

I am grateful to my university teachers' patience. A special thanks to my close friends, without whose support I would not finish this work

Contents

List of Figures

List of Abbreviations

POS tagging	**P**art **O**f **S**peach tagging
NER	**N**amed **E**ntity **R**ecognition
NLP	**N**atural **L**anguage **P**rocessing
MVP	**M**inimum **V**iable **P**roduct
TF-IDF	**T**erm **F**requency - **I**nverse **D**ocument **F**requency
TF-IUVF	**T**erm **F**requency - **I**nverse **U**ser **V**ocabulary **F**requency
ML	**M**achine **L**earning
Frequency	
UD	**U**niversal **D**ependencies format
PLRS	**P**ersonalized **L**earning **R**ecommendation **S**ystem
ESL	**E**nglish as **S**econd **L**anguage
SLA	**S**econd **L**angauge **A**cquisition

Dedicated to my physics teacher, who did not just give the fish but taught me how to fish

Chapter 1

Introduction

1.1 Motivation

The problem of language learning appeared with the first word said to the human being. According to modern studies, there are almost 7000 languages in existence (Anderson, 2010). The largest online open dictionary contains more than 1 million entries. It would take 38 times the expected lifetime in Ukraine to read all of them 1 per day. Although we do not need such a rich vocabulary to communicate with others, there are times when one needs to enrich it with new words.

The most popular language in the world is English (Eberhard and Fennig, 2021). It has an interesting property: only 27% of speakers are native speakers, and 73% are those who learn this language as their second. It indicates a great need for the development of high-quality English learning resources, as more and more people learn this language every year for personal, scientific, or professional use (Parker, 2015).

Learning a language is a complex process that involves many different activities and methods. We need to re-evaluate educational methods and techniques in modern times as many of them have already become outdated or aren't based on current human psychology and mentality (Tomlinson, 2008).

Nowadays, technologies allow us to change the teaching and learning approach by including personalization and gamification techniques in creating educational materials for learners. Recent events connected with global pandemic show us that education can be quickly and efficiently redesigned as an online system. Many businesses and educational facilities started to develop new technologies that allow learners to get quality materials online. (Paudel, 2021)

Another thing that becomes increasingly popular is self-education. More and more resources offer students the possibility of learning without a teacher's physical or even verbal presence. This tendency will allow the education system to become more efficient in the future, as the constant need for professional educators to be in personal contact with the students will slowly decline. Instead, these people will focus more on creating educational materials that can be available online for everyone, which in theory will lead to cheaper and faster education. Nowadays, resources like Khanacademy, Udemy, Coursera, Duolingo show us that it is not necessary to pay colossal college tuition fees in order to master new skills (Jiang et al., 2020). Duolingo and other resources rely heavily on gamification, which somewhat fulfills the role of a teacher in motivating learners and understanding their needs (Flores, 2015).

We witnessed the great success of recommendation engines in large internet companies such as Amazon, Netflix, YouTube. Views from recommendations take from 30 percent of total views in Amazon up to 80 percent in Netflix (Smith and Linden, 2017). The scientific community tried to adapt the technology to language learning

to personalize learning in student needs. Jie Lue developed a framework for personalized learning recommender systems (PLRS) (Lu, 2004). Ildikó Pilán et al. (Pilán, Volodina, and Borin, 2017) applied machine learning to assess L2 complexity for exercise candidates for e-learning environment. These and other works emphasized the perspective of applying personalization to language learning.

1.2 Research questions

The scale of the language learning problem and the success of technologies in assisting student motivated us to find answers to the following questions:

- What makes the learning process efficient in terms of retention and speed?
- How collected user data can improve learning outcomes?
- How to build an e-learning system for personalized student learning?

1.3 Goals

1. Overview and describe existing works and findings regarding:
    - Foreign and Second Language acquisition personalization.
    - E-learning systems in the language domain.
    - Efficient learning techniques.
2. Develop a framework and MVP based on the done research.
3. Evaluate implemented e-learning system on real students and describe the results comparing to existing works.

Chapter 2

Related works

2.1 Existing products

2.1.1 Anki

Anki (*Anki*) is a program for memorising pieces of information from card decks. Card is a pair of question and answer. In the context of language learning it often looks like word as a question and its meaning as an answer. Deck is a group of cards usually united by theme. User can create deck manually or use the shared decks on languages, art, sciences and trivia.

Anki uses **Spaced repetition** (2.2.3) approach to make learning efficient. User learns from cards by repetitive task to recall answer to the question on the front of the card. After user thought on the question the answer is shown. Users should compare their answers to the right answer and self assess themselves. Based on how easy it was for user to recall the answer Anki schedules next review for the card.

The evolution of Anki scheduler algorithm led to counter-intuitive findings. The scheduler was originally based on the SuperMemo SM5 algorithm. However, Anki's default behaviour of revealing the next interval before answering a card revealed some fundamental problems with this algorithm. The main difference between SM2 and later versions of the algorithm is the following:

- SM2 uses learner's performance on a card to select the next time to schedule that card.
- SM3+ use it also to determine the next time to schedule similar cards.

In not just a single cards' performance, but in performance as a cards' group, it seems to be more accurate. This works correctly if a user is very consistent in learning and all cards have similar difficulty. On the other hand, when inconsistencies are introduced into the equation (cards of varying difficulty, choosing a different time for learning), SM3+ make more incorrect guesses at the next interval. As a result, cards are being scheduled too often or vice versa - too far from each other (*Anki 2.0 User Manual*).

2.1.2 Duolingo

Duolingo is one of the most popular language-learning applications. It uses both Grammar translation and audiolingual methods of second language acquisition(Savvani, 2019). This combination allows to provide a complex learning experience and optimize the process. Also, the learning course includes all the parts of language usage (grammar, vocabulary, reading, listening, speaking exercises), so a user has an impression of being immersed in the language environment.

In order to make learning less stressful, Duolingo uses bite-sized lessons. It offers to learn 15 minutes a day instead of getting longer lessons several times a week. This way of learning is also supported by multiple studies in second language acquisition and in cognitive science. They show a lot of advantages for 'distributed' as opposed to 'massed' practice in the target language.

Another significant part of Duolingo's method is making learning similar to entertainment. It incorporates features which are typical for games (e.g. streaks, crowns, gems, XPs, and leaderboards). This encourages Duolingo users to continue their learning. By doing so, Duolingo increases learner enjoyment, shown in several recent researches to be associated with higher willingness to communicate and reduced levels of anxiety.

Duolingo improved its most popular courses by aligning them with the Common European Framework of Reference (CEFR)(Nikolaeva, 2019). CEFR is an international standard used to describe language learners' abilities at different stages of proficiency. According to the CEFR guidelines, language courses shall be focused on communicative functions. That includes things which learners actually can do with a language, such as asking for directions or making an order in a cafe.

Another important part of Duolingo's method is personalization. There are several features that Duolingo provides in order to individualize learning experiences:

- Learners with previous experience in the target language are encouraged to take a placement test and start the course at an appropriate place.
- While working on exercise, learners receive immediate feedback when giving a wrong answer. Therefore, they receive an opportunity to apply it to next questions.
- Duolingo offers practice sessions that use spaced repetition algorithms personalized for each learner.

In addition, Duolingo offers variability of choosing lessons. After passing Level 1 difficulty in one lesson, a user can continue with the same lesson or proceed to the next one. Thus, users are able to choose the way of learning, depending on the goals and needs they have. Some users work to the highest difficulty level before moving on to the next lesson. Others only complete the first difficulty level and then proceed to the next part of the course. There are also users who combine these 2 strategies to create their own unique way of learning.

Jiang, Rollinson, Plonsky, and Pajak compared the results of using Duolingo with the results of getting language courses as a part of higher education. To do so, they assessed the reading and listening proficiency of Duolingo users learning Spanish and French, who had taken the beginning level courses. After that, scholars compared their scores to the results of students who had taken four semesters of these languages in their universities. The ACTFL proficiency scores showed the following results for Duolingo users:

- Novice High level in listening (for both Spanish and French learners).
- Intermediate Low in reading for Spanish learners.
- Intermediate Mid in reading for French learners.

These scores are at the same level as the proficiency scores of 4th semester university students. In other words, when Duolingo Spanish and French learners reached Checkpoint 5 they got the same result that can be reached completing four semesters of classes (Jiang et al., 2020).

2.2 Language learning strategies

2.2.1 Item response theory

Item response theory (IRT) is a system for the design and analysis of tests measuring the level of proficiency in skills. This theory of testing is based on the relationship between students performance on a test and their general performance in a certain field of study. The unique feature of this system is that each item value and importance is measured individually. This is why IRT is considered to be a lot more effective in evaluating students' level of knowledge than more traditional testing approaches (Thompson, 2009).

2.2.2 Active Recall

Active recall is a principle of learning based on the the need to actively stimulate memory during the studying process. It is quite different from passive review, in which the learning material is processed without immediate practical application of learned knowledge as it is in reading or watching educational materials. For example, reading a translation of a foreign word or even it's definition is a passive review if no active learning happens after it. Active recall method would require student to do an exercise using this new word or at least write a sentence with a new word in it.

Active recall is based on the psychological testing effect and has proven its effectiveness in consolidating long-term memory. Studies show that it is the quickest and the most effective way to study written materials when it comes to factual and problem-solving tests since it is extremely efficient for committing details and ideas into ones' memory (Karpicke and Blunt, 2011).

2.2.3 Spaced repetition

The spacing effect theory states that people tend to remember information more effectively if they use spaced repetition practice. This practise is based on using short study periods spread out over time. The phenomenon was first documented by Ebbinghaus (Ebbinghaus, 1885). His experiment consisted of comparing the results of two methods of studying: one cram-up study session and three study sessions spread over three-day period. Both methods led to very similar results on tests of his knowledge of learned 12-syllable sequences, but spaced repetition method took cumulatively 50% less time.

The lag effect (Melton, 1970) is the related observation that people learn even better if the spacing between practices increases over time. For example, a learning schedule might begin with review sessions a few seconds apart, then minutes, then hours, days, months, and so on, with each successive review stretching out over a longer and longer time interval (Settles and Meeder, 2016).

Studies also show that spaced repetition is very efficient in second language learning. It helps to improve results greatly, along with reducing the amount of testing needed to check students' knowledge (Metsämuuronen, 2013).

2.3 Complexity adjustment

One of the main challenges in e-learning is identifying language proficiency level to provide suitable learning material. We identified two groups of proposed solutions:

- Identifying text complexity of exercises based on the content.
- Modeling user needs based on the interactions.

2.3.1 Text complexity classification

Many text features were proposed in ESL and text-mining literature. Kurdi described how to identify and use set of features that can describe the phonological, morphological, lexical, syntactic, discursive, and psychological complexity for further text complexity classification(Kurdi, 2020). ldikó Pilán et al. (Pilán, Volodina, and Borin, 2017) proposed framework and provided empirical solution for selecting exercises based on the predicted text complexity and heuristic rules to match student proficiency level. They did comprehensive study and evaluated proposed solution with english teachers and students. The evaluation showed that automatic complexity classification gives good enough results to assist teaching professional or by providing possibility for self-learning for students by proposing suitable excercices.

Kurdi in his work (Kurdi, 2020) experimented with five machine learning algorithms and 118 features to build a classifier that can distinguish text complexity. They collected a corpus of texts of English. 6171 text documents were collected from six free professional websites and edited specially for ESL students. Each text labeled with one of three difficulty levels: 1, 2 and 3. These levels were collected from the websites as well and correspond respectively to A2, B1, and B2 in the Common European Framework of Reference for Languages.

This paper sheds light on what features are the most useful in task of text complexity classification. Kurdi made a comprehensive survey of the features from the literature. He extracted 118 features: 99 linguistic features, 12 psychological features and 7 readability formulas. Linguistic features covered five areas: **Phonology, Morphology, Syntax, Discourse, Lexicon**. The researcher compared F-scores (Fawcett, 2006) with each of these linguistic areas using the five ML algorithms as shown in Figure 2.1

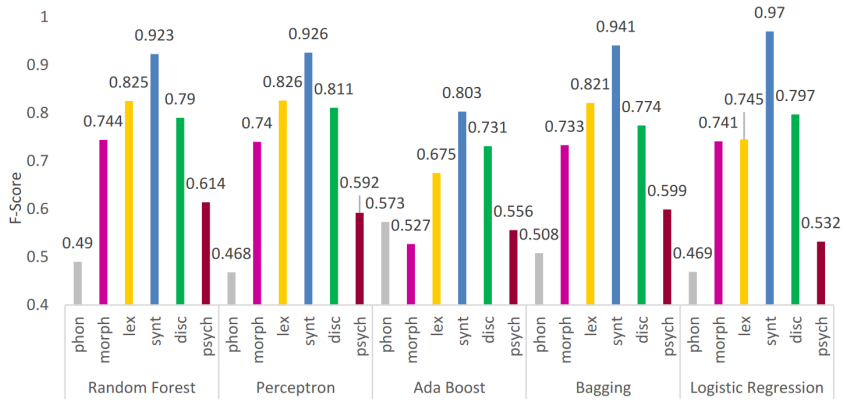

FIGURE 2.1: F-score results per linguistic area. Figure from (Kurdi, 2020)

The results of comparison showed that the biggest impact was made by syntax features. Lexical, discourse and morphology could also be used for accurate text complexity classification.

Although, text complexity classification may be used to match text with predefined complexity levels for users, such an approach ignore individual differences

in knowledge. Students with different background but the same level according to CERF may differ in their weak and strong skills. Moreover, they may have large differences in known lexicon due to professional background. To estimate text complexity for particular user it would be beneficial to include information about user knowledge as well.

2.3.2 User personalization

Using recommender system showed its efficiency in predicting customer needs on practice (Smith and Linden, 2017). Personalization may be applied to the wide number of products including e-learning environments for language learning. Jie Lue was one of the first who proposed a framework for personalized learning recommender systems (PLRS) (Lu, 2004). In this framework researcher described four main components:

1. **Getting student information** - this component aims to collect the student information both asking directly and by mining historical data.
2. **Student requirement identification** - this component should apply a multi-criteria student requirements analysis model (this model takes into account both similarity between students and direct user feedback) to identify learning needs of the student.
3. **Learning material matching analysis** - this component should use fuzzy matching rules to match user requirements and learning materials.
4. **Learning recommendation generation** - this component will should determine the number of recommended learning materials and prepare them for each user based on discovered associations by previous component.

The proposed framework integrates content-based and collaborative recommendations. System may generate recomendations for users based on individual attributes, history of learning, known personal interests, and other requirements. A learning recommender system based on this framework is designed to optimize recommendations and reduce the amount of false positive errors or suggestions consisting of materials that student doesn't like. Although, the work described high-level approach for matching students with suitable learning materials, it did not address the problem of identification of students requirements and matching it to material in details.

Language Acquisition Modelling

Burr Settles et al. presented shared task of Second Language Acquistion (SLA) modelling. The task was to predict future user errors on exercises from Duolingo based on his previous history (Settles et al., 2018)). For this task they prepared a dataset containing history of token-level errors made by the student in the learning language exercises. The goal was to predict future errors of the students.

They mostly focused on three Duoling exercises formats. These exercises require active recall from the students. They must answer in the second language through translation of transcription. Sample data from the corpus presented in Figure 2.2. Data format was inspired by Universal Dependencies (UD) format (Petrov, Das, and McDonald, 2011a). Each token is associated with student answer validity (correct or wrong), morpho-syntactic features and specific token (word).

```
# user:XEinXf5+  countries:CO  days:2.678  client:web  session:practice  format:reverse_translate  time:6
oMGsnnH/0101   When      ADV    PronType=Int|fPOS=ADV++WRB                                    advmod    4  1
oMGsnnH/0102   can       AUX    VerbForm=Fin|fPOS=AUX++MD                                     aux       4  0
oMGsnnH/0103   I         PRON   Case=Nom|Number=Sing|Person=1|PronType=Prs|fPOS=PRON++PRP      nsubj     4  1
oMGsnnH/0104   help      VERB   VerbForm=Inf|fPOS=VERB++VB                                    ROOT      0  0

# user:XEinXf5+  countries:CO  days:5.707  client:android  session:practice  format:reverse_translate  time:22
W+QU2fm70301   He        PRON   Case=Nom|Gender=Masc|Number=Sing|Person=3|PronType=Prs|fPOS=PRON++PRP  nsubj     3  0
W+QU2fm70302   's        AUX    Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin|fPOS=AUX++VBZ    aux       3  1
W+QU2fm70303   wearing   VERB   Tense=Pres|VerbForm=Part|fPOS=VERB++VBG                       ROOT      0  0
W+QU2fm70304   two       NUM    NumType=Card|fPOS=NUM++CD                                     nummod    5  0
W+QU2fm70305   shirts    NOUN   Number=Plur|fPOS=NOUN++NNS                                    dobj      3  0

# user:XEinXf5+  countries:CO  days:10.302  client:web  session:lesson  format:reverse_translate  time:28
vOeGrMgP0101   We        PRON   Case=Nom|Number=Plur|Person=1|PronType=Prs|fPOS=PRON++PRP      nsubj     2  0
vOeGrMgP0102   eat       VERB   Mood=Ind|Tense=Pres|VerbForm=Fin|fPOS=VERB++VBP               ROOT      0  1
vOeGrMgP0103   cheese    NOUN   Degree=Pos|fPOS=ADJ++JJ                                       dobj      2  1
vOeGrMgP0104   and       CONJ   fPOS=CONJ++CC                                                cc        2  0
vOeGrMgP0105   they      PRON   Case=Nom|Number=Plur|Person=3|PronType=Prs|fPOS=PRON++PRP      nsubj     6  0
vOeGrMgP0106   eat       VERB   Mood=Ind|Tense=Pres|VerbForm=Fin|fPOS=VERB++VBP               conj      2  1
vOeGrMgP0107   fish      NOUN   fPOS=X++FW                                                    dobj      6  0
```

FIGURE 2.2: Sample exercise data from an English learner over time: roughly two, five, and ten days into the course. Figure from (Settles et al., 2018)

Totally 15 teams participated in the task. Osika et al. showed the best results with an ensemble model which combined the prediction from a Gradient Boosted Decision Tree (GBDT) and a recurrent neural network model (RNN). (Osika et al., 2018) Although their solution was not evaluated in a realistic production environment, they achieved high predictive performance. Osika et al. noted that each model separately would not have yielded first place in the task. Researchers stated that RNN and the GBDT show different performance on different types of word mistakes. There is a very high chance that the temporal dynamics modelled by the neural network model will be able to complement the GBDT predictions making possible for the ensemble to generalise unseen user events a lot better than its initial component parts.

They also compared features importance by ranking GBDT features by information gain. The unique user identifier was ranked as second most important feature, right after token. This observation suggests that GBTD may build a separate subtree for each user, which leads to the problem with generalization for new users. We also see this as an indicator of importance of addressing individual differences between students.

Chapter 3

Background information

3.1 Natural Language Processing

The most common medium for data storage and transfer between people is text with inherent language "encoding." Nowadays, we have advanced computer technologies able to process a large amount of data and processing information. But the ordinary human-readable text, even if stored in machine-readable form, does not exhibit the same amount of *information* for machines as it does for a human. To make such information available for standard computer processing methods, it has to be extracted (or "mined") with specific techniques. The Natural Language Processing (NLP) is an interdisciplinary field of knowledge on the verge of linguistics, computer science, and artificial intelligence concerned with interactions between computers and human language; human-language text "understanding" is one of the primary tasks (Feldman and Sanger, 2006). In this particular work context, we rely on NLP not only in the context of one-way processing human->computer. We also care about other details extracted from this transition: lexical features in particular, as our aim is to work not only with text but with people's perception of this text.

Today there are a plethora of solutions already available for NLP problem-solving. In this work, we don't implement such algorithms from scratch but use ready solutions instead. Following the advice from the reviews of open-source NLP libraries, our choices fell on the SpaCy: an open-source software library for NLP aimed at industrial usage.Honnibal et al., 2020

SpaCy is written in the programming languages Python (Martelli, 2005) and Cython (Behnel et al., 2011), so it provides ease of development together with speed of execution(Choi, Tetreault, and Stent, 2015).

NLP libraries give the convenience of working with linguistic knowledge extracted from raw text, but working with them as black boxes is infeasible; some NLP-domain-specific knowledge is required from the user. It is common practice for such libraries to provide abstractions of modules related to solving one specific NLP task. The SpaCy follows this pattern providing 'Components' (Honnibal et al., 2020) that could be used isolated or, what is more natural for real-world problems, combined in a text analysis pipeline.

3.1.1 Text mining

SpaCy's design aims to process a raw text to the *Doc* object that comes with a variety of annotations. From a given set of components, the user has to compose the text processing pipeline. Each of Components could be modified, skipped, replaced, but the basic configuration usually resembles one depicted at 3.1.
We will describe the most relevant NLP tasks on an example of their place in such standard pipeline:

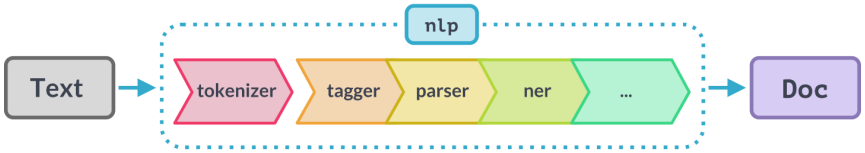

FIGURE 3.1: Principal scheme of typical NLP pipeline used by SpaCy. Picture from (Honnibal et al., 2020)

- **Tokenization**
  Tokenization is the preprocessing phase that transforms the sequence of characters into a series of tokens (meaningful strings). Tokenization splits text into sentences and sentences into words, based on inherent properties of language (for English, it is pretty simple as sentences are separated with punctuation and words are separated with delimiters). Delimiters like whitespaces, tabs and line breaks are being dropped in this phase, but punctuation usually is stored as separate tokens for future analysis step purposes (**Farrel_1995**).

- **Part-of-speech tagging (POS)**
  The **Part-of-speech tagging** is the process of specification corresponding parts of speech to the words. It requires processing context, as it is the only way to disambiguate words belonging to the different parts of speech (Petrov, Das, and McDonald, 2011b) (For example, the word *"**set**"*, which can be a *verb*, *adjective* or a *noun*).

- **DependencyParserser**
  The DependancyParsing component recognizes the sentence grammar (hence, logical) structure. It segments the sentence and labels words with their dependency tree information. Having the grammatical roles in place, we can merge over-segmented tokens into a single lexical unit. Resulting dependency structure could be depicted as shown at 3.2

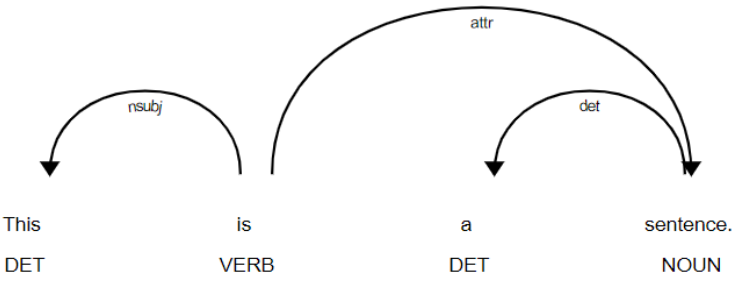

FIGURE 3.2: Strucutre of parsed sentence. Scheme from (Honnibal et al., 2020)

- **Named Entities Recognition (NER)**
  This stage is responsible for recognizing and categorizing proper names: peoples, places, entities, time expressions, quantities, etc. In the context of the English language, NER is simplified by capitalization; still, it has some caveats, i.e., cases when entity name consists of several words and only one of them is capitalized (Nadeau and Sekine, 2007).

- **Lemmatization**
  The Lemmatization stage reduces words to their lemmas: dictionary, canonical form, or citation form. (Zgusta, 2006) This is the form used in standard dictionaries. In the same way, we can codify them into a metaphorical programmatic dictionary in our system,

  As the 'canonical' human language dictionary has a countable and fairly limited set of words (lemmas), we can replace the lemmatized representation with a detailed index standing for the position in such an immutable dictionary.

After the processing with NLP-library we have our initial text transformed into structured document format, enriched with metadata and lemmas relations.

3.1.2 Bag Of Words

As we stated above 3.1.1, words after lemmatization could be replaced with the indices in some predefined dictionary for further processing purposes. Developing this approach even further, we can summarize the text as an occurrences count for each dictionary entry. Expression in the form of integer vector will lose most of the information and drop all the structure and context. On the other hand, it is condensed and fast for computer processing, storage, and comparisons. Such representation in NLP is called the **bag-of-words** model (Feldman and Sanger, 2006). Of course, such vector will contain mostly zeroes at indices of absent words, so usually, it is implemented as a set of value pairs: index in dictionary and number of occurrences. As analyzed documents have different sizes, the absolute values in bag-of-words representation are not helpful for comparisons. Therefore they are usually normalized so that each vector's values sum up to 1. These representation values could be interpreted as a probability to find the given lexeme by random pick from the text.

3.1.3 Term Frequency - Inverse Document Frequency(TF-IDF):

TF-IDF is a numerical statistic that reflects how important a word is to a document in a collection or corpus (Rajaraman and Ullman, 2011).

Term frequency (TF) is a frequency of term (lemma) t occurrences in a given document d. There are several interpretations of "frequency" in this context, but for our needs, we use the "Boolean" TF that only indicates whether the word is present in the considered document 3.1. (Manning, Raghavan, and Schütze, 2008):

$$\begin{aligned} tf(d,f) &= 1 \text{ if document contains this term} \\ tf(d,f) &= 0 \text{ otherwise} \end{aligned} \tag{3.1}$$

Inverse Document Frequency (IDF) is a metric that describes how much information each term bears in the context of a given documents corpus (Robertson, 2004). Let's consider two examples: the words "the", "and", "is" are present in all of the documents; therefore, presence of such word does not bring much information about document specificities - those have to have very low IDF value. As an opposite example, the words like "platypus", "schizophasia," or "Hippopotomonstrosesquippedaliophobia" are rarely met and bring a highly specific context with them. Therefore documents containing the same word of such kind are likely to share the same topic; such words have to have a large IDF value. To hold those

properties, IDF could be expressed as a logarithmically scaled inverse fraction of the documents that contain the word: the ratio of documents containing this word to the total number of documents, taken with logarithm.

$$Idf(t, D) = log(\frac{N}{1 + |d \in D : t \in D|}) \tag{3.2}$$

Where:

N : is the number of all documents:$|D|$

$|d \in D : t \in D|$: number of documents d having such term t

$(" + 1" :$ stands for correction that handles case of division by zero$)$

TF-IDF combines the TF and IDF metrics to describe the amount of information each word brings to a selected document in the context of a given documents corpus 3.3:

$$tfidf(t, d, D) = tf(t, d) * idf(t, D) \tag{3.3}$$

IDF filters-out common terms: ratio of documents with such terms to all the documents will get close to 1, IDF will get close to $log(1) = 0$ while being always $>= 0$.

We are using described metric to compare user vocabularies instead of usual documents. Such adaptation is justified, as the "Boolean" version of TF 3.1 metric we are using does not distinguish one or several occurrences of the same word. So as any document is indistinguishable from its vocabulary for such metric, comparison of documents and vocabularies are interchangeable.

Our adaptation of TF-IDF metrics approach Term Frequency - is Inverse User Vocabulary Frequency (TF-IUVF). This metric will be used as values for each lemma to compare users by their vocabularies: vocabulary will be a vector of TF-IUVF values.

$$tfiuvf(t, uv, UV) = tf(t, uv) * iuvf(t, UV) \tag{3.4}$$

Where:

uv stands for individual user vocabulary

UV is a set of all uvs in our system

$tf(t, uv)$ is defined similar to $tf(t, d)$3.1 and $iuvf(t, UV)$ similar to $idf(t, D)$3.2

3.2 Recommender systems

Recommender systems are a class of information filtering systems that seek to predict the "rating" or "preference" a user would give to an item (Ricci, Rokach, and Shapira, 2010). Such systems are used when we have a problem of providing relevant items (content, services, products) to users with a goal of satisfaction from provided items maximization. To provide the recommendations, such system needs to build some user profiling and target item classification.

As we are considering a system that should provide learning goals for users, it lays to the category of Content recommendation systems. For content-based filtering, we should provide a series of discrete, pre-tagged characteristics of an item (text, learning challenges in our case) in order to recommend additional items with similar properties (Mooney and Roy, 1999). For this purposes, the ordinary TF-IDF 3.3 metric could be used.

But the main part of our system will be devoted to the users' similarity profiling based on their previous behavior and characteristics estimated by our system. This part will be based on a collaborative filtering approach 3.2.1.

3.2.1 Collaborative filtering

Collaborative filtering is a method of making predictions (filtering) about the interests of a user by interpolating them based on collected preferences or taste information from many users (collaborating). The cornerstone assumption for collaborative filtering is that users with similar experiences that had similar preferences in the past will continue to agree on their opinions in the future. System based on this method generates recommendations based on which items user liked in the past. This method is focused not on target item features but on the user ratings for the same items (Ricci, Rokach, and Shapira, 2010).

Collaborative filtering approaches are classified into model-based and memory-based approaches (Breese, Heckerman, and Kadie, 2013). For the purposes of this work we are interested in *memory-based* collaborative filtering systems. Such system builds user profiles from recorded item's "rating" data. We will use mostly the user vocabularies for this purpose.

Gathered users' vocabulary data is convenient to represent in the form of a matrix of size $|D| \times |U|$, where $|D|$ is the length of our lemmas dictionary and $|U|$ are the number of all active users. Each column of such matrix will stand for the user vocabulary vector as individual entries would stand TF-IUVF 3.4 scores for each word. Such representation will play the role of **Utility Matrix** described by Rajaraman and Ulrich (Rajaraman and Ullman, 2011). Typically such utility matrix contains some implicit measure of user preferences towards represented items or user explicit feedback in form rating.

Recommendation system task is to predict missing "ratings" in utility matrix for further selection of the most relevant items. In case of vocabulary data our "ratings" in constructed utility matrix are TF-IUVF values. We calculate rating prediction for the user U_i and lemma L as shown in 3.5

$$R_{U_i,L} = \frac{\sum_{j \in I} simil(U_j, U_i) * R_{U_j,L}}{\sum_{j \in I} simil(U_j, U_i)} \tag{3.5}$$

Where:

U_j : is an user with lemma L in vocabulary

R_{U_j,L_j} : is a rating of lemma L and user U_j

$simil(U_j, U_i)$: is a similarity between users U_j and U_i

For similarity measurement between two users we using a cosine-based approach, defined as following 3.6 (Breese, Heckerman, and Kadie, 2013):

$$simil(x,y) = cos(\vec{x},\vec{y}) = \frac{\vec{x}\cdot\vec{y}}{||\vec{x}|| \times ||\vec{y}||} = \frac{\sum_{i\in I_{xy}} r_{x,i} r_{y,i}}{\sqrt{\sum_{i\in I_x} r_{x,i}^2}\sqrt{\sum_{i\in I_y} r_{y,i}^2}} \tag{3.6}$$

Where:

I_{xy} : is set of all words learned by both users x and y

Memory-based collaborative filtering approach advantages includes:

- Results are openly interpretable, so we can tune the metrics easily.
- Implementation and integration of such approach is simple.
- Does not require interpretation of added items content: only users experience from this item matters for us.
- Small overhead on adding new data.

Yet, it has several problems that has to be addressed in implemented systems:

- **Cold Start**: for the new user, there is not enough data to provide a good recommendation. The approach to handle this problem will be considered in Proposed Approach part of our work (4).
- **Scalability**: when the amount of users in such system becomes large, the explicit exact computation of best-N matches becomes expensive. This could be handled with Local-sensitive hashing (Feldman and Sanger, 2006) and other algorithms and data structures approaches.
- **Data sparsity**: this problem is the most severe for systems with lots of items, where active users cannot effectively provide ratings for all of them, like e-commerce use-cases. However, in our project, the amount of words in vocabulary is limited, and the size of the popular vocabulary subset is limited even more, so this problem is not a limiting issue for us.

Chapter 4

Proposed Approach

To test our ideas on personalizing language learning for self-education or as a part of a study course, we built an MVP website for private tutoring. Based on done research, we identify three primary goals for our system:

- Collect information about the user language proficiency.
- Challenge users to reinforce their knowledge.
- Personalize user learning content and adapt the complexity.

To achieve these, we designed a few learning activities and a framework for collecting the user interactions to model their knowledge and personalize learning. We consider training exercises as a building block for our system. We call it a **challenge**. The developed system implements only three challenges for vocabulary and tenses skills. However, we believe that it is possible to build a self-contained learning platform based on the proposed framework.

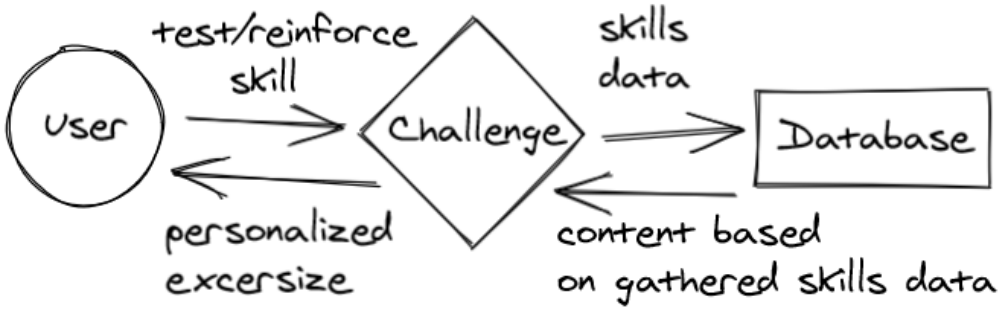

FIGURE 4.1: Framework components

Figure 4.1 represents the flow of the data throughout the system. The database contains learning materials such as manually prepared exercises, generated exercises, reading texts. This data is used to challenge the user with some task that tests his vocabulary, grammar, tenses and reinforces gained knowledge. Users proceed with tasks and collect metrics useful to determine how successful the user is in completing each type of task. The system uses gathered skills data to adapt challenges and introduce new content to get the best recall and amount of learned information.

Those components perfectly lie in Model-View-Controller (Leff and Rayfield, 2001) software design pattern. Users get challenge through View components. They pass the challenge by interacting with Controller. Model receives measured skills data and then sends adapted content back to View to challenge the user again.

4.1 Challenges

The proposed framework **challenge** consists of:

- User interface for displaying task and receiving answers.
- Indicators of user success on passing the challenge (e.g., number of wrong answers).
- Type of challenge, associated skills.

We implemented three challenges in our MVP:

- Reading challenge.
- Flashcard challenge.
- Tenses challenge.

4.1.1 Reading challenge

Vocabulary is essential for both understanding and using the language. We see reading exercises as easy and straightforward way to collect information on the user vocabulary. That is why the first challenge users encounter in our MVP is reading. Reading challenge is the initial point for learning more about user language proficiency and adapting later challenges.

User interface

We developed an interactive reader which displays possibly unknown words in the list on the right (Figure 4.2). The user challenge here is to mark words as known or add them to the learning list.

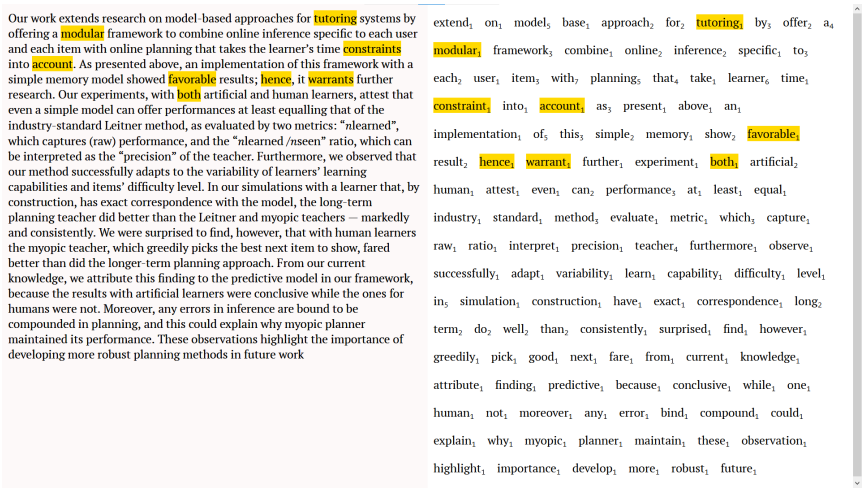

FIGURE 4.2: Reading challenge. Highlighted words were added to the learning list. Click on the cross marks them as known and deletes them from the list

Reading texts may be recommended by the system or pasted by the user. This way we allow adding new words in the learning vocabulary from the sources user intends to read so they start learning needed words right away. It would be even

more comfortable to provide a user with a browser plugin like Duolingo (*Duolingo*) for this purpose. However, we also recommend texts of optimal complexity to learn new useful individually picked words. When the user opens the reader, it selects unread text suitable for user vocabulary with the highest recommendation score of new words from the manually formed corpus of texts. System tokenizes the given text and returns the list of tokens (lemma and part-of-speech-tag) possibly unknown by the user. These tokens are rendered in the list of words on the right, where the user can mark them as known or add them to the learning queue.

Collected indicators

To adapt this and other challenges, we collect data from the user vocabulary formed after reading exercises in the form of database entry with information on:

- **User ID**.
- **Challenge ID** - interactive reader.
- **Skill ID** - vocabulary.
- **Item ID** - here: a word ID identifying unique combination of lemma and part-of-speech tag.
- **Quality** - marking user success on skill challenge with this particular item (Boolean identifying if user knew the word).
- **Challenge time**.

FIGURE 4.3: Reading Challenge schema

Each indicator except time and quality has its dimension table with the definition of the indicator values. This information is stored in the database table and used by workers to personalize challenging content.

4.1.2 Flashcard challenge

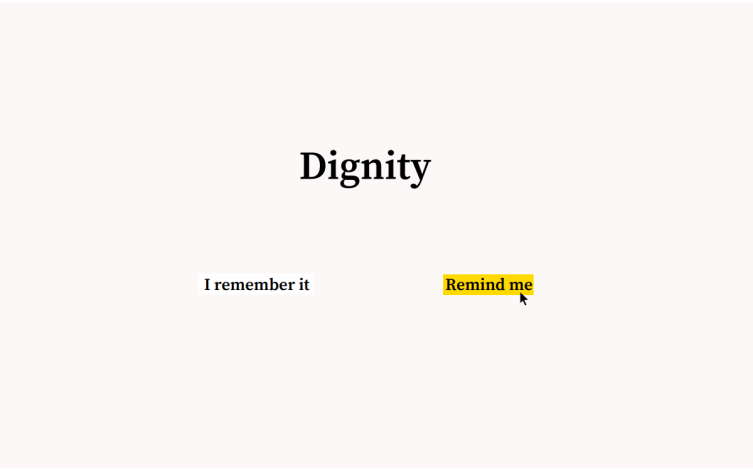

FIGURE 4.4: Flashcard challenge. Recall stage

To memorise new words user needs to challenge their learned vocabulary. Inspired by Anki, we implemented our own vocabulary challenge with flashcards. Using reading challenge user collects words they will learn in the flashcard challenge.

User interface

Their task in this challenge to recall word meaning and to answer if they succeed on this task (Figure 4.4).

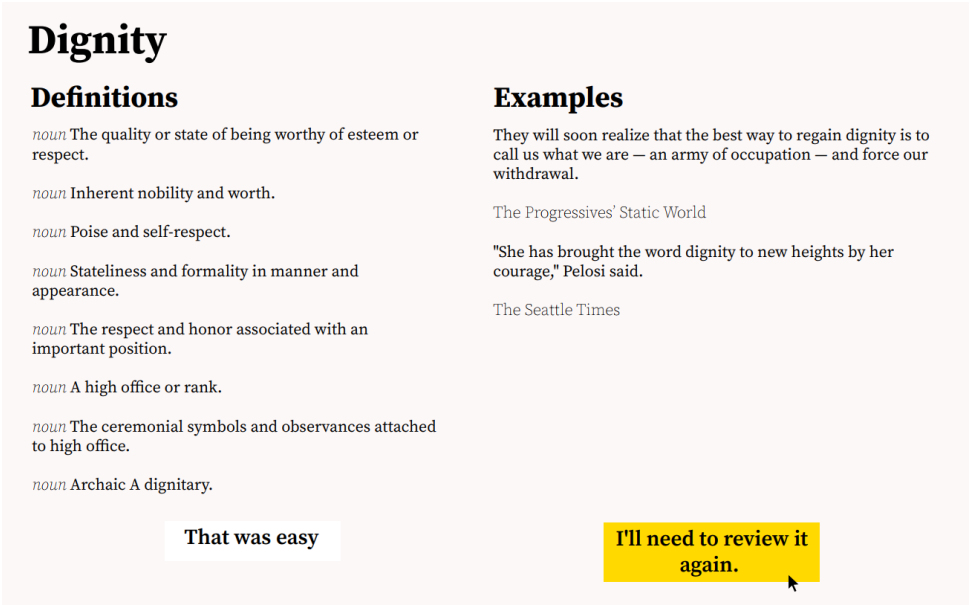

FIGURE 4.5: Flashcard challenge. Self-assessment stage

After the user's response, the system will display definitions and usage examples of the word. In case if user claimed to remember the word, they will be asked if he remembered it right (Figure 4.5).

Collected indicators

This challenge collects the same information as the reading challenge with the difference in **Challenge ID** identifying flashcard challenge.

4.1.3 Tenses challenge

To cover more than just lexical exercises and test how the approach would work on English grammar learning, we implemented a tenses challenge. We manually prepared 100 Present Simple sentences into the database for this challenge in our MVP.

User interface

To test how well the user knows the tenses, we give them a task to put the word from the brackets in the right form (Figure 4.6). We have the right answers in the database and compare them with those given by the user. The user sees their mistakes and learns a rule associated with this kind of task (Figure 4.7).

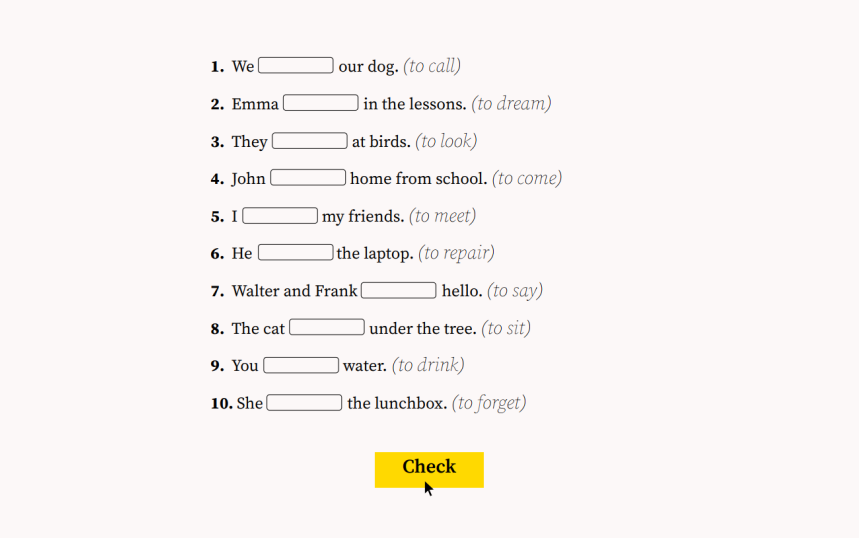

FIGURE 4.6: Tenses challenge. User input

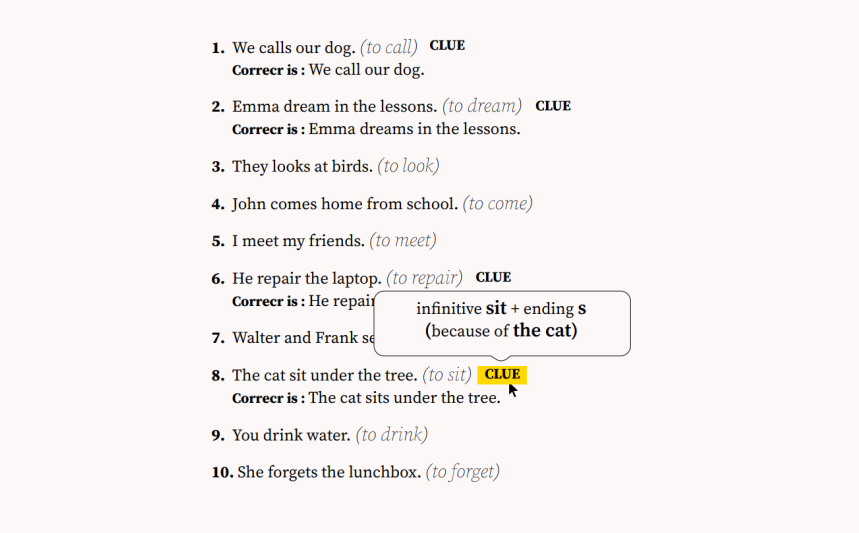

FIGURE 4.7: Tenses challenge. Correct answers

Collected indicators

We use the same scheme for all challenges as we described in the reading challenge (4.3). The main difference of tenses challenge is the value behind **Item ID**. In this case, it does not identify a word but a testing sentence in the database.

4.2 Challenges adaptation

For each challenge, we aggregate indicators of how hard or easy the challenge is. Indicators for the reading challenge are known and unknown words; for tenses they are mistakes in each tense. Those measurements are stored for each user to see the dynamics of their skill acquiring. This data could be used to adjust challenges complexity and focus the user on skills that need improvements.

4.2.1 General approach

The task of complexity adaption in our case is a task of complexity prediction as we select existing materials instead of adapting them. We need to model user knowledge based on the challenges, but before they start the next ones.

The way we collect challenges data is similar to the Duolingo exercises corpus (Settles, 2018). It contains identifiers of the challenges, time, and the measurement of user success in the challenge. With such a format, we could use such an approach as Osika used in the Second Language Acquisition Modeling Task (Osika et al., 2018). However, we could not use the exact data for our challenges because Duolingo has other exercises. Furthermore, we still need time to collect enough actual usage data to train our own models.

Instead of using machine learning models on historical data to balance too easy and too hard, we developed few empirical algorithms for each challenge. Collected data is used to change the selection of prepared exercises and learning materials to teach a user in a designed, personalized way.

4.2.2 Vocabulary challenges

The user learns new words and repeats the learned ones. To make the process efficient, we put learned words to order in the **Flashcard challenge** 4.1.2 to remember them best and calculate the value of each word for the user using the recommendation engine.

Spaced repetition

To maximize the retention in the long run, we implemented a concepts scheduler based on spaced repetition research. We used the SM-2 algorithm (Wozniak, 1990), which was used in the first versions of the SuperMemo software package. The scheduler model works in a next way:

1. Challenges with vocabulary related tasks sends the quality of recall from a scale of 0 to 5 regarding each challenged word (we map collected indicators for words as 0 - the user did not know the word, 5 - the user easily remembered the word, 1 - the user remembered the word wrong)
2. Next review date is computed by SM-2 algorithm.
3. Words for vocabulary related tasks will be selected by their computer review date in ascending order.

Recommendation score

There are many words the user can learn, but what are the words which worth learning right now? James Tauber defines learning value as a measure of sentences that words allow to read(Tauber, 2005). He models the task of vocabulary learning as a Travelling Salesman Problem (Cormen, 2009). However, we should point that such a measurement does not distinguish between sentences. We still face the problem where we should decide what sentences are worth reading.

In our opinion, a key to this problem is the fact that language is a social phenomenon. The main task we solve with this tool is communicating with others. Hence we decided to recommend user words to learn based on other users' vocabularies.

We use collaborative-filtering 3.2.1 approach here. We consider words in user vocabulary as items they rated positively and words the user doesn't know have zero-rating in the utility matrix. So, at first the new user will learn words which most users know. Those words should cover the basic vocabulary needed for common situations. At the same time, the more advanced user who wants to learn specific vocabulary (e.g., professional or some dialect) will learn new words based on his specific interests.

This approach is prone to the cold start problem. However, we use this score only to rank learning materials (see Reading challenge recommendations 4.2.3) after the user finished the introductory session in the reading challenge with default or personal texts to gather the first information on vocabulary. The problem could affect only words that are not used by anyone, which we solved by adding "superuser" , who knows all existing words, to the database.

4.2.3 Reading challenge personalization

For our reading challenge, we suggest reading user texts. This is our way to enrich users' vocabulary with new and useful words. We believe that we should have a small number of new words in the familiar context for better retention. We also want users to learn words that are relevant to their interests and existing knowledge.

Complexity adaption

To select texts which are suitable for reading with a little struggle because of new words but easy enough to understand the context, we calculate the ratio between known and unknown words in the text. This simple heuristic (4.1)

$$U = W_{unknown} / W_{text} \tag{4.1}$$

Our baseline for optimal percent of unknown words in the text is 0.2. This **words novelty coefficient** may be tuned for other users. We select the top hundred texts which are the closest to the 0.2 using formula for ordering (4.2)

$$V = 1/U * (0.2 - U)^2 \tag{4.2}$$

Recommendations

Next thing we calculate is a recommendations score based on the recommendations scores of all unknown words in the sample. We select text with the largest sum of all unknown word scores.

4.2.4 Tenses challenge personalization

To make learning tenses free from excessive unknown information, we select sentences based on user vocabulary. We use the same heuristic (4.1) to select sentences with the smallest number of unknown words.

4.3 Architecture

4.3.1 Data storage

To make the system extendable, we needed to build a way to save samples needed for challenges and collect challenges success indicators. The challenge model allows us to store user interactions as facts and store words, exercises, and any learning material info in dimension tables. By using Star schema (Corr and Stagnitto, 2011) in our database, we ensure that at any time, we can add more indicators to the challenges or new items for the challenges.

4.3.2 Technology stack

To make the challenges available for our users, we implemented them using:

- Python and Typescript as main programming languages.
- PostgreSQL as a database.
- Google Cloud Platform as hosting service.
- Gunicorn as the HTTP server.
- APScheduler library for making aggregations and updating challenge content jobs.
- Spacy for NLP.

Chapter 5

Results

We implemented MVP and manually tested the system. We have not evaluated the proposed approach with students yet. However, developed challenges mostly were inspired by findings from cognitive science and existing solutions described in Related Works 2, which have shown their efficiency in second language learning and broader educational context.

The developed system will be used by students of private tutors in Lviv. We expect at least 20 users of ages from 14 to 40. We plan to collect data and questionnaire feedback from them. It will be used to tune our existing challenges and develop new ones.

5.1 Recommendations

Although we could not measure the efficiency of our approach for memorizing words and learning tenses, we evaluated our recommendation engine with a collected corpus of texts. It has been shown that offline evaluations have a low correlation for user studies or A/B tests (Beel et al., 2013). Moreover, instead of actual historical data, we collected data from other sources. Hence, provided analysis does not suggest that the proposed recommendations will satisfy students. However, this analysis helps us to understand if our assumptions regarding recommendations ome true.

Dataset

We collected data using arxiv papers scrapper (*https://github.com/karpathy/arxiv-sanity-preserver*). We fetched papers of 6 categories:**cs.AI, math.ST, stat.ML, q-bio.NC, q-bio.QM, q-bio.TO**. Each paper contained metadata, including the authors of the paper. Totally 8000 authors of 3000 papers are presented in the collected corpus. These works contain both shared lexicon and domain-specific lemmas, simulating users with different learning backgrounds.

Experiment

We registered unique authors as users in our system, extracted 5000 words from their papers, and added to their vocabularies as if they marked known words in the reading challenge 4.1.1. Based on these words, we calculated TF-IUVF described in Background chapter 3.4. We used scikit-learn TF-IDF implementation and filtered common words using the default English stop words list from that library (*scikit-learn docs*).

After that, we calculated user similarities and missing ratings in the utility matrix. We limited the number of recommended words to 100 and explored few popular metrics for recommender systems, and visualized characteristic cases of recommendations.

Evaluation metrics

We calculated two validation metrics to characterize the recommendations:

- **Coverage** is the percent of items in the data the model is able to recommend on inference. 100 most relevant recommendations for each user form a union of 1240 unique words out of 5000, which is 24.8% of the all extracted words.
- **Personalization** is the dissimilarity (1 - cosine similarity) between users' lists of recommendations. This metric can help to determine if the model recommends many of the same words to different users. The highest this score, the more personalized recommendations are. Our personalization score is 0.7315596374733893

Those metrics in combination suggest that the recommendation system tends to select a small subset of popular words out of the whole dictionary and at the same time meets individual user interests.

Recommendations exploration

We chose demonstration examples of researchers' vocabularies and recommended word lists. We selected two categories: **q-bio.NC** (Neurons and Cognition) and **stat.ML** (Machine Learning). We extracted the most characterizing words for these categories based on TF-IDF, for which we treated categories as documents. We visualized these words as a word cloud, where word size corresponds to the TF-IDF value. We did the same with known and recommended words for users using calculated ratings based on TF-IUVF filled utility matrix. (see Background 3.2.1

Using these visualizations, we discovered that recommended words for researchers mostly come from the same domain and can contain popular words from the whole dictionary. Author of papers in **stat.ML** category knows words (Figure5.2) from his category (Figure 5.1) and gets in recommendations more words from that domain (Figure 5.3).

At the same time **q-bio.NC** researcher vocabulary (Figure 5.6)) comes from his domain (Figure 5.5), while recommended words (Figure 5.7) are selected mostly from his domain, but also contain words used in ML domain.

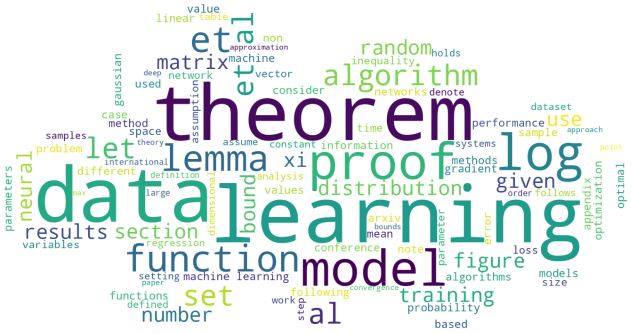

FIGURE 5.1: stat.ML most used words

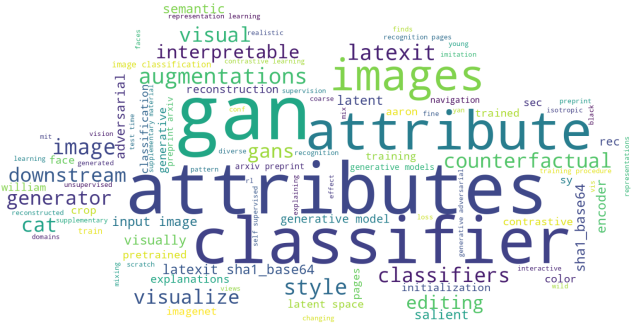

FIGURE 5.2: stat.ML researcher known words

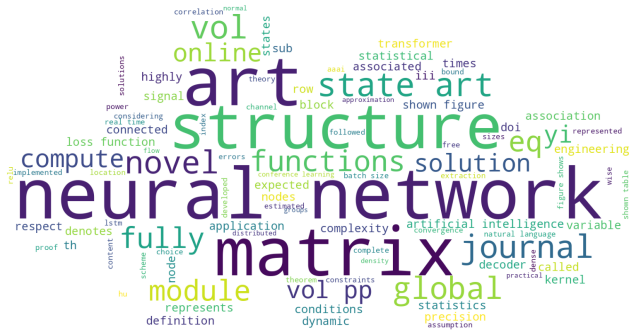

FIGURE 5.3: stat.ML researcher recommended words

FIGURE 5.4: Words cloud for stat.ML category and researcher

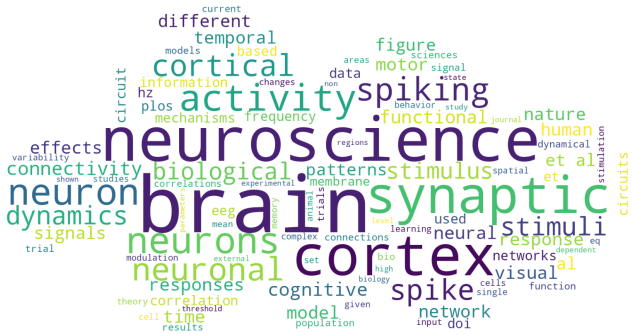

FIGURE 5.5: q-bio.NC most used words

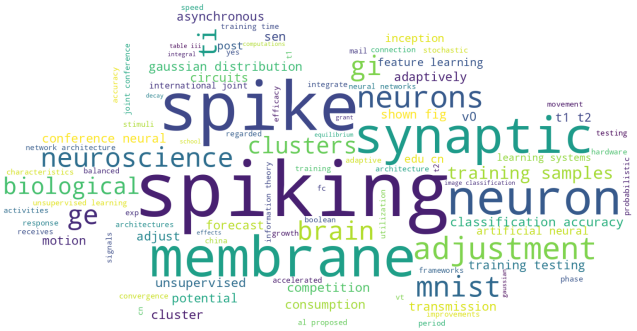

FIGURE 5.6: q-bio.NC researcher known words

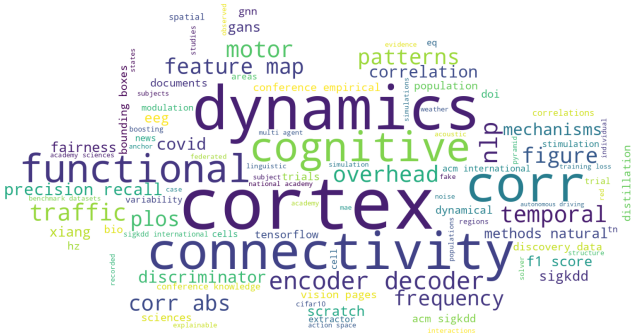

FIGURE 5.7: q-bio.NC researcher recommended words

FIGURE 5.8: Words cloud for q-bio.NC category and researcher

5.2 Framework

The proposed framework was designed relying on the existing solutions example. Its main component challenge consist of user interface, adaptive content model and student performance indicators. It allows us to measure response to each individual learning item, which allow us to benefit from Item Response Theory 2.2.1 when implementing challenges. We can also easily extend the developed MVP with variety of learning activities. Star schema, factual and dimensionality tables allow us to add new item tags, adaption schemes and measurements for challenges.

5.3 Conclusion

In this work, we overviewed existing approaches to personalized and efficient language learning. We investigated what learning techniques and user data can improve the learning process. We proposed a framework for creating an e-learning platform for profiling user knowledge and providing personalized study materials and implemented MVP. The developed solution consists of heuristics and NLP pipelines for processing text and user data, extendable database and architecture, collaborative-filtering recommendation system, learning exercises, and simple user interface. Although we did not evaluate the whole solution, it synthesized efficient learning techniques and well-recommended algorithms. We also described and evaluated using collaborative filtering on user vocabulary for prioritizing word learning. Based on the done evaluation, we also expect this method to be useful in a production e-learning environment.

5.3.1 Future work

Although this research allowed us to transform gained knowledge into MVP for language learning, we are still interested in our main research question: **What makes the learning process efficient in terms of retention and speed?** We will continue our work on extending and testing the existing system. Our main priorities in the near future are:

- **Creating new challenges** - we plan to implement more exercise formats starting with adapted existing exercises from educational resources.
- **Defining learning performance indicators** - based on the collected challenges data, we could gain insights on learning system improvement by measuring students' progress.
- **Collecting feedback from students** - we plan to add questionnaires for our beta-users to gather feedback right into the database with other data.
- **A/B testing** - we will develop a management system for splitting our users into groups and providing them with different versions of algorithms.
- **Tuning heuristics and introducing machine learning models** - with such infrastructure and user feedback, we could tune our existing heuristics and try to train some ML models instead (e.g., models that were proposed for the shared SLA task (Osika et al., 2018))

Bibliography

Anderson, Stephen R. (2010). *How many languages are there in the world?* URL: https://www.linguisticsociety.org/content/how-many-languages-are-there-world.

Anki. *Anki*. URL: https://apps.ankiweb.net/.

Beel, Joeran et al. (2013). "A Comparative Analysis of Offline and Online Evaluations and Discussion of Research Paper Recommender System Evaluation". In: *Proceedings of the International Workshop on Reproducibility and Replication in Recommender Systems Evaluation*. RepSys '13. Hong Kong, China: Association for Computing Machinery, 7–14. ISBN: 9781450324656. DOI: 10.1145/2532508.2532511. URL: https://doi.org/10.1145/2532508.2532511.

Behnel, Stefan et al. (2011). "Cython: The Best of Both Worlds". In: *Computing in Science Engineering* 13.2, pp. 31–39. DOI: 10.1109/MCSE.2010.118.

Breese, John S., David Heckerman, and Carl Kadie (2013). *Empirical Analysis of Predictive Algorithms for Collaborative Filtering*. arXiv: 1301.7363 [cs.IR].

Choi, Jinho D., Joel Tetreault, and Amanda Stent (July 2015). "It Depends: Dependency Parser Comparison Using A Web-based Evaluation Tool". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, pp. 387–396. DOI: 10.3115/v1/P15-1038. URL: https://www.aclweb.org/anthology/P15-1038.

Cormen, Thomas (2009). *Introduction to algorithms*. Cambridge, Mass: MIT Press. ISBN: 9780262033848.

Corr, Lawrence and Jim Stagnitto (2011). *Agile data warehouse design: Collaborative dimensional modeling, from whiteboard to star schema*. DecisionOne Consulting.

Duolingo. *Duolingo*. URL: https://www.duolingo.com/.

Ebbinghaus, Hermann (1885). *Über das gedächtnis: untersuchungen zur experimentellen psychologie*. Duncker & Humblot.

Eberhard David M., Gary F. Simons and Charles D. Fennig (2021). *What are the top 200 most spoken languages?* Dallas, Texas. URL: https://www.ethnologue.com/guides/ethnologue200.

Elmes, Damien. *Anki 2.0 User Manual*. URL: https://www.webcitation.org/6E8NpPAT3?url=http://ankisrs.net/docs/manual.html#what-spaced-repetition-algorithm-does-anki-use,lastchecked={01.05.2021}.

Fawcett, Tom (2006). "An introduction to ROC analysis". In: *Pattern Recognition Letters* 27.8. ROC Analysis in Pattern Recognition, pp. 861–874. ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2005.10.010. URL: https://www.sciencedirect.com/science/article/pii/S016786550500303X.

Feldman, Ronen and James Sanger (2006). *The Text Mining Handbook*. Cambridge University Press. DOI: 10.1017/cbo9780511546914. URL: https://doi.org/10.1017/cbo9780511546914.

Flores, Jorge Francisco Figueroa (2015). "Using gamification to enhance second language learning". In: *Digital Education Review* 27, pp. 32–54.

Honnibal, Matthew et al. (2020). *spaCy: Industrial-strength Natural Language Processing in Python*. DOI: 10.5281/zenodo.1212303. URL: https://doi.org/10.5281/zenodo.1212303.

Jiang, Xiangying et al. (2020). "Duolingo efficacy study: Beginning-level courses equivalent to four university semesters". In:

Karpathy, Andrej. *https://github.com/karpathy/arxiv-sanity-preserver*. URL: https://github.com/karpathy/arxiv-sanity-preserver.

Karpicke, Jeffrey D. and Janell R. Blunt (2011). "Retrieval Practice Produces More Learning than Elaborative Studying with Concept Mapping". In: *Science* 331.6018, pp. 772–775. ISSN: 0036-8075. DOI: 10.1126/science.1199327. eprint: https://science.sciencemag.org/content/331/6018/772.full.pdf. URL: https://science.sciencemag.org/content/331/6018/772.

Kurdi, M. Zakaria (2020). *Text Complexity Classification Based on Linguistic Information: Application to Intelligent Tutoring of ESL*. arXiv: 2001.01863 [cs.CL].

learn, scikit. *scikit-learn docs*. URL: https://scikit-learn.org/0.24/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html?highlight=tf#sklearn.feature_extraction.text.TfidfVectorizer.

Leff, A. and J. Rayfield (2001). "Web-application development using the Model/View/Controller design pattern". In: *Proceedings Fifth IEEE International Enterprise Distributed Object Computing Conference*, pp. 118–127.

Lu, Jie (2004). "A personalized e-learning material recommender system". In: *International Conference on Information Technology and Applications*. Macquarie Scientific Publishing.

Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze (2008). "Scoring, term weighting, and the vector space model". In: *Introduction to Information Retrieval*. Cambridge University Press, 100–123. DOI: 10.1017/CBO9780511809071.007.

Martelli, Alex (2005). *Python cookbook*. Beijing Sebastopol, CA: O'Reilly. ISBN: 978-0-596-00797-3.

Melton, Arthur W (1970). "The situation with respect to the spacing of repetitions and memory". In: *Journal of Verbal Learning and Verbal Behavior* 9.5, pp. 596–606.

Metsämuuronen, Jari (Jan. 2013). "Effect of Repeated Testing on the Development of Secondary Language Proficiency". In: *Journal of Educational and Developmental Psychology* 3.1. DOI: 10.5539/jedp.v3n1p10. URL: https://doi.org/10.5539/jedp.v3n1p10.

Mooney, Raymond J. and Loriene Roy (1999). "Content-Based Book Recommending Using Learning for Text Categorization". In: *CoRR* cs.DL/9902011. URL: https://arxiv.org/abs/cs/9902011.

Nadeau, David and Satoshi Sekine (Aug. 2007). "A survey of named entity recognition and classification". In: *Lingvisticæ Investigationes. International Journal of Linguistics and Language Resources* 30.1, pp. 3–26. DOI: 10.1075/li.30.1.03nad. URL: https://doi.org/10.1075/li.30.1.03nad.

Nikolaeva, Sofiya (June 2019). "THE COMMON EUROPEAN FRAMEWORK OF REFERENCE FOR LANGUAGES: PAST, PRESENT AND FUTURE". In: *Advanced Education* 6.12, pp. 12–20. DOI: 10.20535/2410-8286.154993. URL: https://doi.org/10.20535/2410-8286.154993.

Osika, Anton et al. (June 2018). "Second Language Acquisition Modeling: An Ensemble Approach". In: *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 217–222. DOI: 10.18653/v1/W18-0525. URL: https://www.aclweb.org/anthology/W18-0525.

Parker, Bridget (2015). *More than any other foreign language, European youths learn English*. URL: https://www.pewresearch.org/fact-tank/2015/10/08/more-than-any-other-foreign-language-european-youths-learn-english/.

Paudel, Pitambar (2021). "Online education: Benefits, challenges and strategies during and after COVID-19 in higher education". In: *International Journal on Studies in Education* 3.2, pp. 70–85.

Petrov, Slav, Dipanjan Das, and Ryan McDonald (2011a). *A Universal Part-of-Speech Tagset*. arXiv: 1104.2086 [cs.CL].

— (2011b). *A Universal Part-of-Speech Tagset*. arXiv: 1104.2086 [cs.CL].

Pilán, Ildikó, Elena Volodina, and Lars Borin (2017). *Candidate sentence selection for language learning exercises: from a comprehensive framework to an empirical evaluation*. arXiv: 1706.03530 [cs.CL].

Rajaraman, Anand and Jeffrey David Ullman (2011). "Data Mining". In: *Mining of Massive Datasets*. Cambridge University Press, 1–17. DOI: 10.1017/CBO9781139058452.002.

Ricci, Francesco, Lior Rokach, and Bracha Shapira (Oct. 2010). "Recommender Systems Handbook". In: pp. 1–35. DOI: 10.1007/978-0-387-85820-3_1.

Robertson, Stephen (Oct. 2004). "Understanding inverse document frequency: on theoretical arguments for IDF". In: *Journal of Documentation* 60.5, pp. 503–520. DOI: 10.1108/00220410410560582. URL: https://doi.org/10.1108/00220410410560582.

Savvani, Stamatia (2019). "State-of-the-Art Duolingo Features and Applications". In: *The Challenges of the Digital Transformation in Education*. Ed. by Michael E. Auer and Thrasyvoulos Tsiatsos. Cham: Springer International Publishing, pp. 139–148. ISBN: 978-3-030-11935-5.

Settles, Burr (2018). *Data for the 2018 Duolingo Shared Task on Second Language Acquisition Modeling (SLAM)*. Version V4. DOI: 10.7910/DVN/8SWHNO. URL: https://doi.org/10.7910/DVN/8SWHNO.

Settles, Burr and Brendan Meeder (2016). "A trainable spaced repetition model for language learning". In: *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 1848–1858.

Settles, Burr et al. (2018). "Second language acquisition modeling". In: *Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications*, pp. 56–65.

Smith, Brent and Greg Linden (2017). "Two decades of recommender systems at Amazon. com". In: *Ieee internet computing* 21.3, pp. 12–18.

Tauber, James (2005). *Programmed Vocabulary Learning as a Travelling Salesman Problem*. URL: https://www.gwern.net/docs/www/jtauber.com/f2ca8373664f682d34ebe7ff4b0829a14516c.html.

Thompson, Nathan A (2009). "Ability estimation with item response theory". In: *Assessment Systems Corporation* 20.

Tomlinson, Brian (2008). "English language learning materials: A critical review". In:

Wozniak, Piotr A (1990). "Optimization of learning". In: *Unpublished master's thesis, Poznan University of Technology. Poznan, Poland*.

Zgusta, Ladislav (2006). *Lexicography then and now : selected essays*. Tbingen: Max Niemeyer. ISBN: 3484391294.