BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS
AND MULTIMEDIA

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

CORPUS PROCESSING FOR FOREIGN LANGUAGE LEARNING

ZPRACOVÁNÍ KORPUSŮ PRO VÝUKU CIZÍCH JAZYKŮ

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR DANIIL KHUDIAKOV
AUTOR PRÁCE

SUPERVISOR Doc. RNDr. PAVEL SMRŽ, Ph.D.
VEDOUCÍ PRÁCE

BRNO 2016

Abstract

The thesis deals with the computer assisted language learning. Special attention is given to the theme of language learning with usage of corpus. The process of developing browser extension for language learning is detailed described. The main feature of the extension is possibility to work with the personalized usage examples of the word. Personalization is based on the student vocabulary and interests. The news portals are used as source of usage examples.

Abstrakt

V Této práci se zkoumá problematika studování cizích jazyků pomocí moderních informačních technologií. Zvláštní pozornost je věnovaná především studovaní jazyků s použitím korpusu. V práci je popsán postup vývoje rozšíření pro internetový prohlížeč s účelem studování cizích jazyků. Specifikem rozšíření je možnost studenta využívat personalizované příklady použitých slov. Personalizace se opírá o slovní zásobu studenta a preferované sféře použití. Příklady použití sebraný ze zpravodajských portálů.

Keywords

corpus processing, database, computer assisted language learning, data driven learning, concordance, elasticsearch, chrome extension

Klíčová slova

zpracování korpusů, databáze, computer assisted language learning, data driven learning, konkordance, elasticsearch, chrome extension

Reference

KHUDIAKOV, Daniil. *Corpus processing for foreign language learning.* Brno, 2016. Bachelor’s thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Smrž Pavel.

Corpus processing for foreign language learning

Declaration

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Doc. RNDr. Pavla Smrže Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

....
Daniil Khudiakov
May 17, 2016

Acknowledgements

Chtěl bych poděkovat docentu Pavlu Smržovi za konzultace a vedení této práce. Taky děkují inženýrům Aleksandru a Irine Deyneka za konzultace v lingvistických otázkách.

Contents

Chapter 1

Introduction

Nowadays, in the epoch of globalization, learning foreign languages is becoming a very important task for many people. In a condition of permanent lack of time for a learner, the process of learning should be as fast and efficient as possible. There are a lot of researches aimed to improve the learning process. One approach that showed its efficiency is to use corpora in language learning process.

Corpora can be used in language learning in different ways, which will be mentioned in the first part of the thesis. Also, the comparative analysis of existing solutions will be made focusing on their benefits and drawbacks.

As a result of studying this problematic, I will create my own extension that uses the corpus for learning foreign languages which should help students with the learning process and expand the functions of existing solutions resolving their gaps. My extension should be easy to use, personalized for each user and could support as many languages as possible. The extension will be integrated into existing language learning portal SpeakASAP.

In the next three chapters of the thesis, the process of development will be described, as well as used technologies and design of our solution.

In the last part of the thesis our solution will be compared with the existing ones. Some pros and cons will be described and the ways for future development will be defined.

Chapter 2

State of the art

In this part we will analyze the role of the computer in language learning. The different ways of corpora usage in language learning process will be described, considering the existing solutions and their drawbacks.

2.1 Computer-assisted language learning applications

Computer-assisted language learning(CALL), was defined by Michael Levy as "the search for and study of applications of the computer in language teaching and learning"[7].

Highlighted 8 generic CALL applications: Word processing, Games, Literature, Corpus linguistic, Computed-mediated communication (CMC), WWW resources, Adapting other materials for CALL, Personal Digital Assistants (PDAs) and mobile telephones[5].

2.1.1 Word processing

Word processing is the type of assistant when program checks user spelling and grammar. Most common example is word processor in Microsoft Office.

In spite of the fact that the word processing is a part of computer language learning, usually the applications of such type are not oriented on language learning. This can be observed particularly in the programs that offer a spelling correction but do not provide any explanation. Thus foreign language learners frequently misspell a word, then choose the first offered correction without considering whether it is appropriate or not.

However, there are such solutions which offer not only the correction of mistakes but also provide the rules with explanation for correct spelling and help to improve user the writing skills.

One of the examples of such applications is **Grammarly**.

The main feature of Grammarly for learning process of foreign language is an explanation of made mistakes. It shows not only where mistake and offers correction, but also adds description of rule relative to this mistake. It checks over 250 points of grammar, supports contextual spellchecks, punctuation checks, style checks and controls structure of a sentence. Except this, it offers the alternatives for the used word in the context. Such feature helps to improve the vocabulary.

2.1.2 Games

Games in language learning can be divided into two types:

1. Games which have positive impact on language learning
2. Games developed for language learning

The example for the first type of games is **Hangman**. In this game player guesses the word by letters and has the limited number of wrong guesses for one word. Playing Hangman user expands vocabulary and improve grammar.

The example for the second type of the games is **Influent**, which is developed from the beginning as a game for language learning. It is possible to describe this game in the following way. One of the often used methods for learning foreign language vocabulary is to use stickers with the name of the item in this language on every item at home. As a result the learner step by step will remember all names of these items. The game Influent uses the same method. The player is at the place where all things are named with the stickers. After remembering the names of all things, the player can try to find these items in the game by their names in learned language.

2.1.3 Literature

Literature in computer language learning generally represents the same as common paper-based literature, except the fact that for better learning the computer literature can be extended by translation of words, definitions of words or references on related materials etc.

For example service **Readlang** offers the texts library and during the process of reading it is possible to translate any word or phrase inside the text. Besides of using the library texts, there is a possibility to use the Readling browser extension, which gives you an opportunity to translate by the same way any text in internet.

Another example is **Google Dictionary**. Google Dictionary is a browser extension, which offers a definition of the selected word from the text.

2.1.4 Computer-mediated communication

Computer-mediated communication (CMC) includes the communication by e-mails, bulletin boards, chatlines within MOO (Multi-user domains, Object Oriented) environments and uses the social networking services such as FaceBook and Twitter. In other words, CMC is a way of language learning by communication.

Except common general social networks such as FaceBook and Twitter, there are special social networks for learning language, which offer special features.

For example **BUSUU** is language learning social network, which has 60 million users. Except the standard exchange messages between users, BUSUU offers the correction of texts, written by the learners and corrected by the native speakers. Also there are language lessons, training of vocabulary and repeating of corrected user mistakes.

2.1.5 WWW resources

WWW resources are web portals for language learning. There are a lot of portals with different ways of teaching. The most popular of them are mentioned below.

Babbel

Babbel offers paid courses and it is mainly based on “drill-and-practice” technique. During the lesson user learns new words and phrases and after repeats them in different ways. For example by associating the phrase from mother language with the phrase from learning language or building the phrase in learning language by letters. After that the user can repeat the learned words or phrases from his Babbel vocabulary.

The main feature of this portal is supporting of many languages. It supports 14 languages, such as German, English, Russian, Spanish, etc.

Duolingo

Duolingo offers free courses which are set of text lessons with exercises. Lessons in course can be opened only in certain order. Thus a new lesson will become active only after you complete the previous one. Another special interesting thing in Duolingo is “Strength Bar”. As soon as user finishes the new lesson the “Strength Bar” will show the full indicator stripe, which symbolized the level of user knowledge. But after some time this stripe starts to decrease, which reminds user to repeat the lesson. After lesson repeated the stripe becomes full again. Also Duolingo uses gamification for user motivation. User will get the bonus points “lingots” after lessons completed. These points can be spent by user on special tasks such as training on time or extended language tests and others.

As drawback Duolingo sometimes uses in lessons unnatural sentences, which usually are not being used in speech.

Duolingo was researched for its effectiveness by Roumen Vesselinov and John Grego. The result of the research has shown that the vast majority of the participants in the study liked the product and most of them succeeded in improving their knowledge of Spanish[10].

Rosseta Stone

Next popular portal is Rosetta Stone. Rosetta Stone offers only paid courses. They offer their own method to learn language. Main idea of this method is that you need to learn the second language as the first language. It means, for example, that you need to learn the new words not by their translation, but by association these words with images. Also Rosetta Stone provides the training of pronunciation by recognition of the user speech.

The drawback of the system is the price. Rosetta Stone is not cheap and moreover they don’t have demo courses.

Rosetta Stone was also researched for its effectiveness by Roumen Vesselinov. The result of the research has shown that this portal is extremely easy to use, very helpful, and enjoyable to work with[10].

Lingualeo

Lingualeo is language learning portal for Russian learners. It started as the site for training and expanding the vocabulary, but now it also has the paid learning courses.

Lingualeo is fully based on gamification. The user level can be increased by systematic learning. User also gets his points for learning good. These points can be spent on special features.

It is interesting to learn language by Lingualeo, because of opportunity to increase the vocabulary by repeating of words in different ways. Also Lingualeo has a big storage of text, audio and video resources in English with the possibility to translate and add unknown words to the vocabulary. As a source of the video, it supports resources such as TED Talks and Coursera.

The disadvantage of the portal is that it supports only learning of English language. Also it has a low level of students involvement. In 2013 only 1% of registered users visited the site every day [4].

SpeakASAP

The last example of WWW resource is portal SpeakASAP. It is also portal for Russian students, that offers the free courses, as well as the paid ones. SpeakASAP offers 16 languages for learning.

The free courses include a base course of language divided into 7 lessons and grammar text and video lessons. Base course is distributed in text, audio and video formats.

The paid courses include two types of courses. The first type is conversation by Skype with Russian speaker teacher and native speaker teacher. The second type is language marathon.

SpeakASAP offers special methodologies of teaching languages. It is based on the following idea: “Language is not the goal, language is a tool for goal achievement”. That is why the main purpose of the course is to teach a student how to speak in foreign language as soon as possible.

Based on the users reviews there are such disadvantages of the system as lack of translations for the words from lessons.

2.1.6 Personals Digital Assistants(PDAs) and mobile telephones

Nowadays a market of language learning applications for mobile platforms is growing very fast. It offers a lot of solutions for learning languages. Each portal mentioned in the part 2.1.5 offers their mobile applications for learning languages. These applications have either mostly the same functionality or limited site functionality.

There are a lot of different types of applications for language learning. It can be set of audio lessons, grammar articles, applications for memorizing words, different chats, etc.

Because portal derived applications have mostly same functionality, we will consider an application which was developed and implemented for mobile telephones.

For example **ChineseSkill** is for learning mandarin language. As mobile application it offers special features such as writing Chinese characters by touching a mobile screen and practicing in pronunciation by using a mobile microphone.

2.1.7 Adapting other materials for CALL

There can be plenty of other materials, which may be adapted for language learning. For example, it can be just surfing in foreign internet pages, watching foreign movies or listening to foreign music.

2.2 Corpus processing in language learning

The main goal of this thesis is to build extension that uses corpus for learning of foreign language. That is why we will discuss this type of CALL in more detail.

2.2.1 Classification of corpora for language learning

Corpus is a collection of written texts, especially the entire works of a particular author or a body of writing on a particular subject.

Corpora for language learning purpose are divided into two types[8].

- General corpus, strand in applied corpus research that aims to inform the teaching syllabus and also stresses the importance of frequency of occurrence, examines language items in actual language use and compares the distributions and patterns found in general reference corpora (of speech and/or writing) with the presentations of the same items in teaching materials (coursebooks, grammars, usage handbooks).
- Specialized corpus, which can be divided into three subtypes:
  - Language for specialized purpose(LSP), aimed to aggregate text data from specialized sources, for example corpus of Italian business letters. It can be used for creation of specific courses in specific area.
  - Lerner corpora aggregates the language produced by language learners. By corpora of this type common mistakes of learners can be determined and the lesson can be altered correspondingly to avoid such mistakes in the future.
  - Parallel corpora represents the language units of one language and their equivalents in another one. It can be used to help students with language items, that cause translation problems.

Applications of corpus in language learning can be divided into two following types:

- indirect applications of corpora
- direct applications of corpora

Indirect application means, that student or teacher does not work with corpus directly, but corpus is used as a way to improve lessons.

For direct application the corpus is used directly by student or teacher. Language learners and teachers get their hands on corpora and concordancers (concordancer gives a list of several words, phrases, or distributed structures along with immediate contexts from a corpus or other collection of texts assembled for language study) themselves and find out about language patterning and the behavior of words and phrases in an “autonomous” way[6]. This type of corpus usage is called Data Driven Learning(DDL).

Data Driven Learning was recognized by many researchers as the very efficient way of foreign language learning[12]. Vocabulary learning must be combined with the context where particular words occur. It is very important for students to acquire the context information. Only in a combination of translation with context the word can be fully understood, and later freely used[11].

In my work I focus on the implementation of Data Driven Learning tool. But before let’s consider existing solution of this type.

2.2.2 Corpus based tools for language learning

In this section, we will analyze existing DDL tools and define their advantages and drawbacks.

SkELL

SkELL is the solution for students and teachers for language learning by using corpus. It stands on corpus manager Sketch Engine.

Except usage examples SkELL offers other ways to explore words such as word sketch and similar words. Word sketch is used for searching collocates of the word. Collocates are the words, which often occur with searched word.

The advantages are big corpus, simple interface and supporting of search by a sequence of words.

As disadvantages can be highlighted supporting of only English language, mistakes in examples, only 40 usage examples for word or phrase.

Foboko sentence dictionary

Foboko sentence dictionary provides examples of word usage for English, Spanish, German, Italian, Portuguese and French languages.

As unusual features, Foboko offers usage examples for word in its different forms, similar expressions and definitions of the word.

The disadvantage is that it can show examples only for single words and it doesn't support the phrases with two or more words.

Tangorin

Tangorin provides usage examples for English or Japanese language with their equivalent in the second language. For providing this functionality it uses parallel corpus - Tanaka Corpus.

Tanaka Corpus is a corpus of parallel Japanese-English sentences collected by people. It is continuously being expanded and corrected by its community. The collection has approximately 150,000 pairs of sentences.

Main features of service, which related to Japanese language, are mentioned below.

- Furigana - Japanese reading aid, that indicates word pronunciation
- Separating of Japanese sentence by words
- Kanji elements on the page for input of search request

Other features are dictionary translation of Japanese words in English and supporting parallel examples for other languages, such as French, Spanish, German, Polish, Chinese, Russian and Italian.

The disadvantage of the system is only partially supporting of other languages. There are only some sentences translated into other languages.

Tatoeba

Tatoeba also provides usage examples with a translation based on a parallel corpus.

The big advantage of this system is supporting of almost all languages. Corpus is completely filled by site community. Except usage examples it also provides the examples of audio pronunciation for some sentences. Pronunciation examples are also made by site community.

In order to increase the number of examples with their translation, they use a special algorithm with indirectly association of sentences. Their algorithm connects two sentences in different languages, which have the same translation.

Most of the time their algorithm works, but not always. In some cases the algorithm does not work, for example figure 2.1.

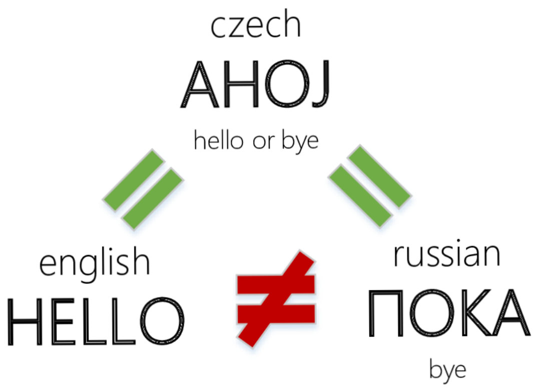


Figure 2.1: Tatoeba incorrect association.

Thus this is a disadvantage of the system.

Chapter 3

Design

My goal is to develop extension based on Data Driving Learning, which should become an assistant for studying of foreign languages. The main purpose of extension is to assist in learning process, as well as to help with understanding of foreign text and expand vocabulary. As main functionality our extension should give users the opportunity to see the personalized usage examples of learned words.

In this section we will define requirements for extension. After that based on these requirements we will develop the design and describe the implementation.

3.1 Requirements

The following requirements are determined for my extension.

Personalization

To improve quality of learning and make the learning process more efficient the extension must be adaptive to the particular user who will learn the foreign language.

Multilingualism

To be universal for many people learning different languages, the extension should support as many languages as possible. Moreover our extension will be integrated into the portal SpeakASAP, which supports more than 15 languages.

Relevancy

Continuous transformation of any language, consisting of arising of new words or expressions or change of context, brings us the requirement to stay “up to date” with the language. This means to use maximally relevant and regularly updated sources of words examples. This will increase the involvement of the user because he will study on the base of relevant to current period of time examples.

Convenient usage

Our extension should be integrated in learning process. Unlike existing solution, where user must go to another web page in order to get help, our solution must help without interruption of learning process.

Simplicity

To be universal for many people independently from their age, computer experience and special knowledge, our solution must be easy to use.

Integration with language learning portal SpeakASAP

Our extension will be integrated into one of the most rapidly growing platform for learning of foreign languages - SpeakASAP. The extension will provide a useful tool and additional functionality into their approach of learning foreign languages.

3.2 Personalization of usage examples

There are several possible ways to provide the personalized solution.

In language learning student often needs to learn language from specific area or there are some areas which are much more interesting for him. From this perspective, one dimension of personalization must be personalization of usage examples by specific area . For this purpose sources of usage examples from different areas are needed.

Another way of personalization is to adjust usage examples on the base of the words from user vocabulary. It increases readability of usage example and also helps students to remember faster already learned words and see usage of these words in different forms.

3.3 Source of usage examples

In order to provide user with personalized example of usage, I have chosen the following approaches.

Source selection

There are many potential sources of examples which we can be used, such as books, user forums, news article, social networks, and other resources that contain text data.

To choose the suitable source of usage examples we need to determine the requirements for it:

1. The source must not contain grammatical mistakes in the text. Any mistakes will have a bad impact on learning process.

2. Content should exist in many languages due to multilingualism of my extension.

3. Content should be relevant for today, the user must be familiar with the words in used context.

4. There should be the possibility to classify the usage examples according to preferred user area.

Taking into account the first point, we can exclude forums and social networks, because there is no redaction in these sources and mistakes are very common. Fiction and non-fiction books are also not suitable, because they can be written in a specific style or they can describe the events in another period of time. Thus the used language constructions or even words are not relevant for today.

In case of news portals, they have an editorial staff and rarely have the literature stylizations. Also we have the opportunity to classify the content based on preferred area. They are always relevant to current time and easily accessible. Moreover such portals usually provide language selection.

That is why based on our requirements we have chosen the News Portals as main source of the examples.

In the extension as a storage of text data from news portals we will use corpus.

Corpus

Corpus is an aggregated collection of text data that was structured.

We can choose between using the existing corpus or creating our own corpus of news articles.

If we use the existing corpus, we will get the huge amount of data at the beginning, but not all of this data will be relevant for today. We will also face a problem to find the existing corpora for all our languages. That is why I have chosen to create my own corpus with the possibility to add sources in different languages and gradually fill it. It will give me more control and more information about source of examples. Information about source will be used for personalization of usage examples.

Aggregation

For creation of my own corpus I need to have the method to aggregate news articles from news portals.

The one of the difficulties related to the aggregation from web resources is the fact that only content of article should be collected ignoring all other contents, for example, pictures, advertisement tags, etc.

3.4 System architecture

Our extension is developed as a client–server model distributed application.

To meet all stated requirement the client side will be implemented as a browser extension.

Realization in form of browser extension will enable to integrate it conveniently into learning process, which in most cases takes place online. Another benefit is simple usage that browser extension brings.

The figure 3.1 shows the interaction between the client and the server part of the extension as well as the communication with the external services.

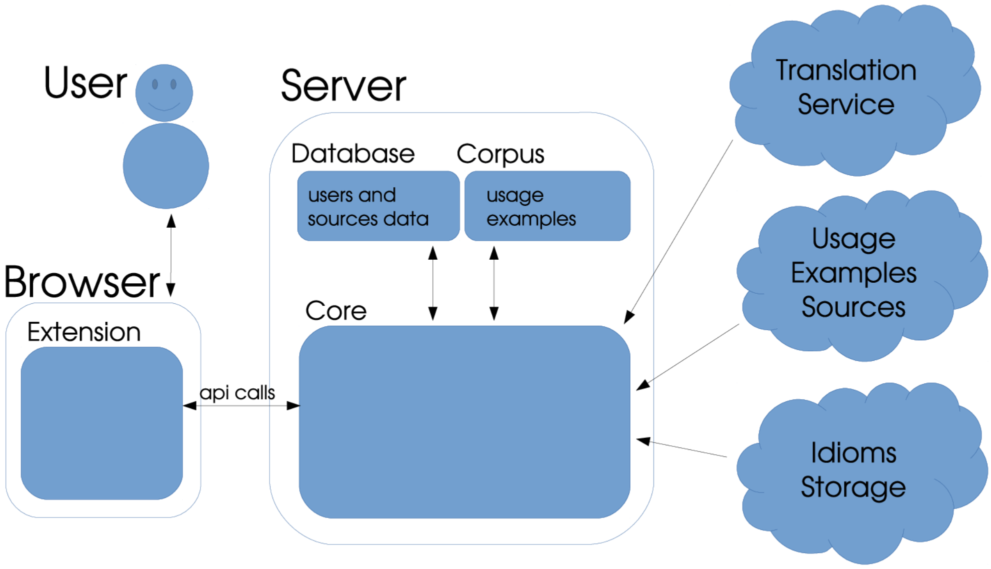


Figure 3.1: System architecture.

3.4.1 Server part

Server part is handling all data processing and communication with the external sources and services. It consists of 2 storages and core element.

The first storage is corpus of news articles which serve as a source of usage examples. To store the corpus we need a storage, that is aimed to work with the big size of a text data. This means that the storage should enable fast search in the text. For this purpose a special search server should be used, which will provide the ability of full text indexing and support the wide range of the searching capabilities.

The second storage is for other data, such as user data including the user dictionary.

Core element is responsible for the whole logic of the system and also for the communication with the following external services:

1. News sources will be used for aggregation of their articles which will serve as a source of usage examples.
2. Translation service will be used as additional functionality of extension for translation of words, phrases and whole sentences.
3. Idioms storage will be used as additional functionality of extension to obtain idioms collection for detection and translation of possible idioms in word context.

It will also provide API for request of usage examples from corpus and other data.

3.4.2 Client part

The extensions are supported by almost all browsers including Internet Explorer, Firefox, Opera, Google Chrome, Safari and Microsoft Edge. For my purposes I have chosen to create the extension for Google Chrome because it has almost 48% of browser market coverage. Moreover more than 50% of visitors of portal SpeakASAP, into which our extension will be integrated, are using Google Chrome, as you can see from the table 3.1.

Browser	Sessions	Sessions, %
1. Chrome	196 709	53,26 %
2. Safari	52 759	14,29 %
3. Opera	38 295	10,37 %
4. Firefox	27 807	7,53 %
5. YaBrowser	24 690	6,69 %

Table 3.1: SpeakASAP browser-sessions statistics.

Communication with the server will be implemented by using the server API. All data procession will be made on server-side. Client-side will serve only as provider and presenter of data.

Chapter 4

Used technologies

In implementation of extension the following technologies were used.

4.1 Elasticsearch

Elasticsearch is an open-source search engine built on top of Apache Lucene^{TM}, a full-text search-engine library[3].

Elasticsearch provides RESTfull API for working with data, High Availability by clustering and Long Term Persistency of data.

Lucene is information retrieval software library. It provides full text indexing and searching capability.

For our purpose we will use Elasticsearch because of the following reasons:

- It is already used by many big portals like Wikipedia, StackOverflow or GitHub
- It has good documentation
- It is easy to use
- It is an open-source
- It has helper library for different languages

4.2 jusText

jusText is a tool for removing boilerplate content, such as navigation links, headers, and footers from HTML pages. It is designed to preserve mainly text containing full sentences and it is therefore well suited for creating linguistic resources such as Web corpora[1].

jusText algorithm can be divided into the following three parts:

Preprocessing, where the elements with header, style and script tags are removed from the page. All elements with select tag are labeled as bad (boilerplate).

Context-free classification, where all elements are classified into different classes:

- bad – boilerplate block,
- good – main content block,
- short – block, which is too short for making decision

- near-good – somewhere in-between short and good.

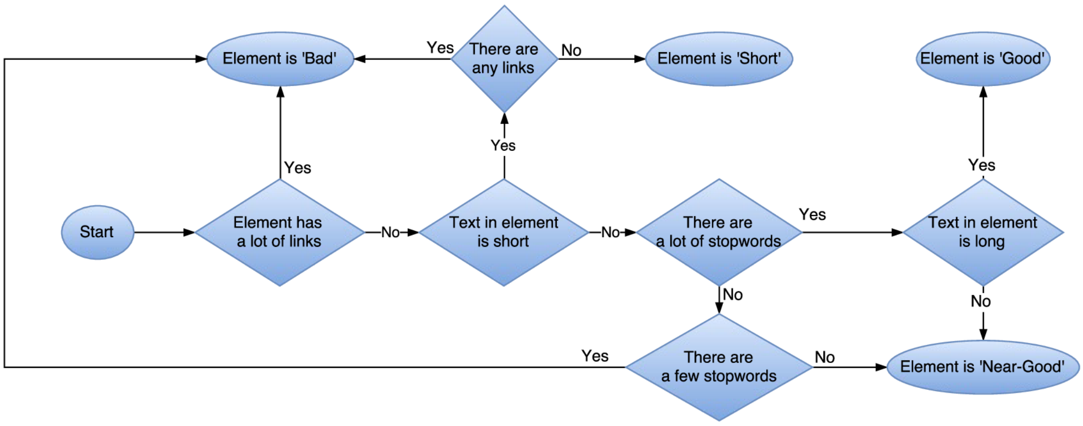
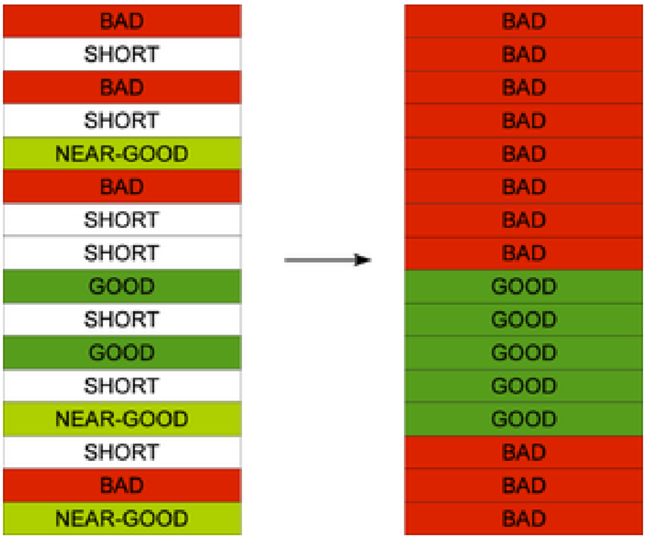


Figure 4.1: Flow diagram of jusText context-free classification.

Context-sensitive classification, where based on the classes of the surrounding blocks the re-classification of the short and near-good blocks for good and bad blocks takes place.



BAD	BAD
SHORT	BAD
BAD	BAD
SHORT	BAD
NEAR-GOOD	BAD
BAD	BAD
SHORT	BAD
SHORT	BAD
GOOD	GOOD
SHORT	GOOD
GOOD	GOOD
SHORT	GOOD
NEAR-GOOD	GOOD
SHORT	BAD
BAD	BAD
NEAR-GOOD	BAD

Figure 4.2: Output of jusText context-sensitive classification.

After last step, all boilerplate elements will have class “Bad” and all content elements will have class “Good”.

Chapter 5

Implementation

In order to solve the problem of developing the desired extension, which meets all our requirements, we should separate the functionality of the program into several parts where each part performs logically discrete functions.

5.1 Usage examples storage

The usage examples storage, which is implemented by Elasticsearch server, has the structure described below.

Mapping of documents

The data in Elalsticsearch are stored in documents which are JSON objects with unique ID.

Usage examples in my extension will be stored in form of sentences.

For my purpose the document of type Sentence was created. To store all needed for our logic information this document has several fields.

- *Text* field contains the text of sentence from news article.
- *Source* field contains the name of the article source. It has aesthetic necessity and also can assists in personalization of usage examples in case if learners prefer special resources.
- *Category* field contains the category of the article. It is needed to personalize the usage examples, because our system will find usage examples based on not only already learned words, but also on chosen category.
- *Link* field contains URL to the article. It is needed to give us the opportunity to show the user the original article.
- *Timestamp* field is needed to sort documents by date. This also will be used for personalization purpose. Extension will show user the sentences from the most recent articles, if he prefers.

To improve the search quality stemming is used.

Stemming is the process of reducing inflected (or sometimes derived) words to their word stem, base or root form—generally a written word form[2].

Snowball analyzer is responsible for stemming in the Elasticsearch server.

An analyzer is used at index Time and at search Time. It is used to create an index of terms.

Elasticsearch analyzer is a set of tokenizators and filters. Tokenizator is a tool used to separate text into tokens, such as individual words. Filters are used for manipulation with tokens.

Snowball analyzer includes:

- *Standard tokenizer* which separates text elements such as grapheme clusters, words and sentences.
- *Standard filter* which is used to normalize the token.
- *Lowercase filter* which converts the letters of the text into lowercase letters.
- *Snowball filter* which returns stemmed form of the word.

Unfortunately it is not possible to implement stemming for all languages. For such languages the step of stemmization will be skipped.

Here is final code of document mapping:

```
mappings = {
  "mappings": {
    "sentence": {
      "properties": {
        "text": {"type": "string"},
        "source": {"type": "string", "index": "not_analyzed"},
        "link": {"type": "string", "index": "not_analyzed"},
        "category": {"type": "string", "index": "not_analyzed"}
      },
      "_timestamp": {
        "enabled": True
      }
    }
  }
}
```

In case stemmization is supported by language the analyzer is added to mapping:

```
mappings["settings"] = {
  "analysis": {
    "analyzer": {
      "stemming": {
        "type": "snowball",
        "language": language,
        "stopwords": "_none_"
      }
    }
  }
}
```

Receiving usage examples from server

In Elasticsearch the results of search are sorted on the base of the *score* value, which depends on the weight of each term that appears in the document. Each query can also contains parameter *boost*, which changes the weight of the term.

For receiving usage examples from Elasticsearch we will increase *score*, if the result contains the words from user vocabulary or the article belongs to the area which user prefers.

The following sample represents the search request to Elasticsearch:

```
"query": {
  "bool": {
    "must": {"match_phrase": {"text": word}},
    "should": [
      {"term": {"category": category}},
      {"match": {"text": vocabulary}}
    ]
  }
}
```

Result must contain the text that user requests and should (but not must) have the same category which user prefer and contain the words from user vocabulary. *Should* cluses increase the *score* of resulting documents with match.

5.2 Relational database

Due to complex structure of data that we need to keep and the necessity to store big collection of data set relational SQL database was used, namely SQLite. The figure 5.1 describes the structure of our database.

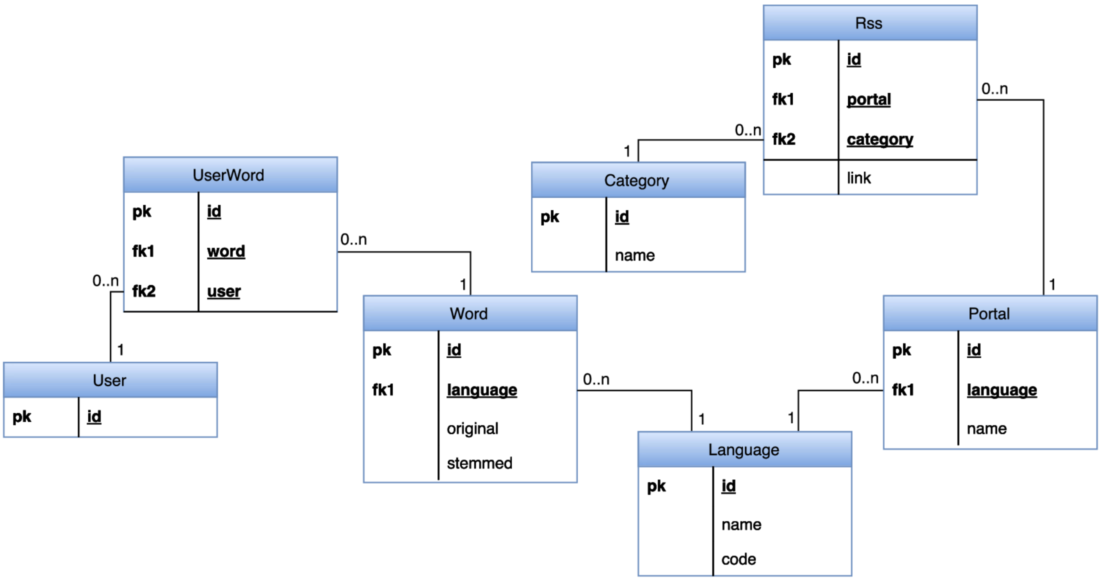


Figure 5.1: Structure of database.

For personalization purpose the user vocabulary was created. User vocabulary is implemented in our database by means of two tables: *UserWord* and *Word*.

Table *Word* contains all unique words that were requested by users. It stores language of the word, original and stemmed forms of the word. Stemmed form is needed to decrease Elasticsearch load by sending there only unique stemmed forms of words.

Table *UserWord* associates *Words* with *User*.

To store the information about RSS feeds, which are used as source of recent news articles of portal, the additional database table is used.

This table consists of three fields:

1. *Link* field contains the reference to RSS feed.

2. *Portal* field contains the description of the source portal, its name and language.

3. *Category* field contains the category of the news feed, for example sport, politics etc. It was separated from the table *Portal*, because the news portal may have several feeds with different categories.

Helper table *Language* stores language name and its ISO 639-1 code. ISO 639-1 code is used frequently as short definition of language. For example, it is used by translation service to indicate source and target languages.

5.3 Server-side logic

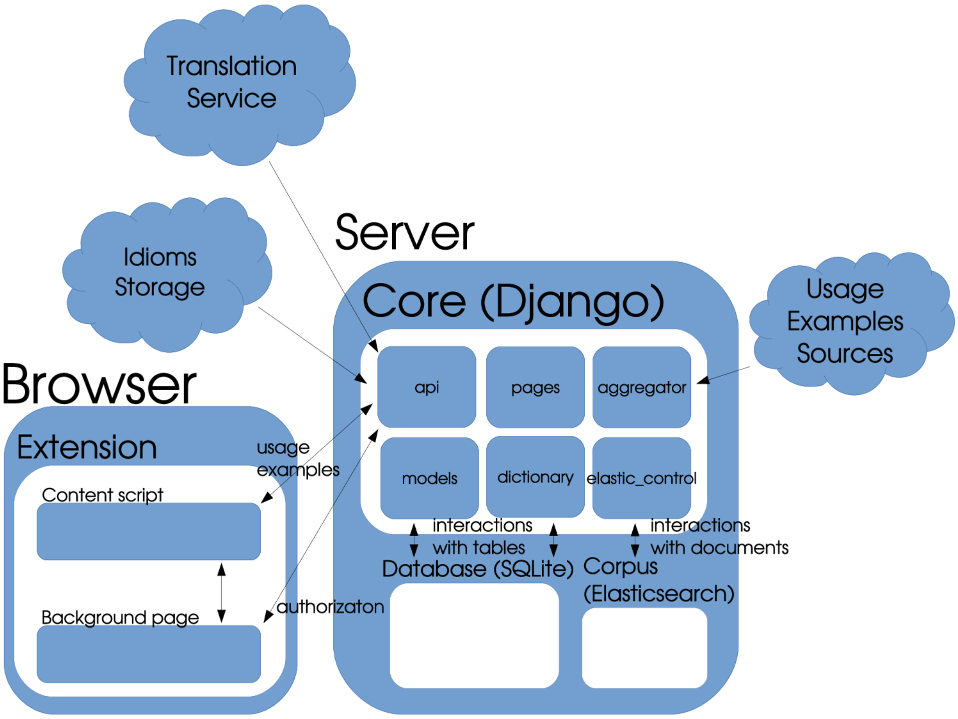


Figure 5.2: Server-side architecture.

As a backend of our system the framework Django was used. Django system is composed of different modules.

For my purpose I have created the following modules:

aggregator module

Module *aggregator* is responsible for aggregation of articles.

For collecting of news articles in corpus the RSS feeds functionality of news portals is used. It is easy and convenient way to update our source of examples with relevant data continually.

aggregator periodically processes the RSS feeds and collects unprocessed articles from the given resources and stores them in Elasticsearch.

To implement periodic tasks a Celery library is used.

aggregator takes all feed sources from database and starts aggregating the sentences from news articles.

Process of aggregation for each news feed is described in figure 5.3.

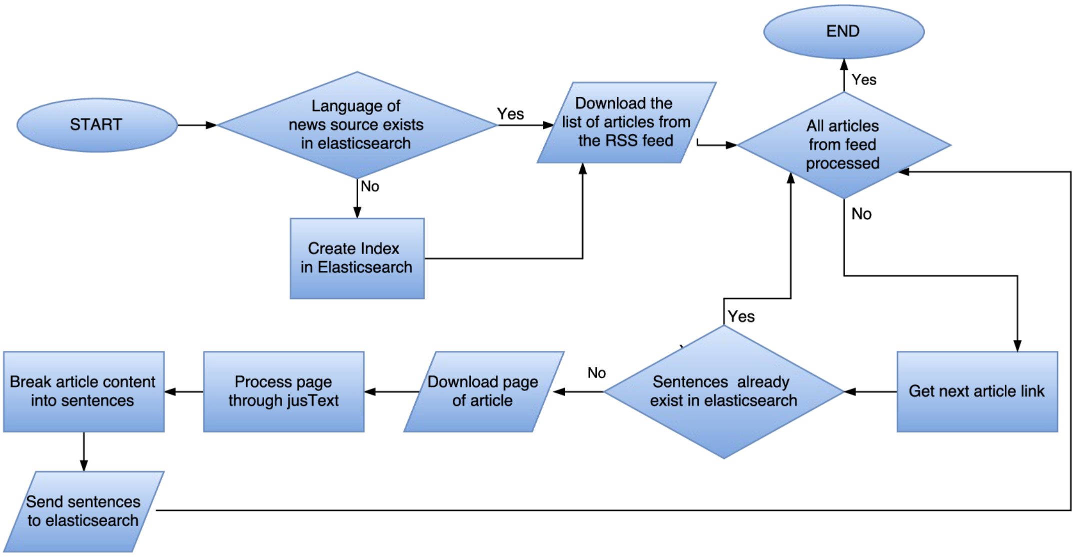

END
Yes
START
Language of news source exists in elasticsearch
Yes
Download the list of articles from the RSS feed
All articles from feed processed
No
No
Create Index in Elasticsearch
Yes
No
Break article content into sentences
Process page through jusText
Download page of article
Sentences already exist in elasticsearch
Get next article link
Send sentences to elasticsearch

Figure 5.3: Usage examples aggregation flow diagram.

api module

Communication between server and client is made by API which consists of two callable methods.

- Method *getinfo* supports two types of request: GET and POST.
  - The request of GET type checks if user is authorized on the server. Boolean value (true or false) is returned to the client, based on authorization result.
  - The request of POST type is used to get from the client usage examples and position of the selection cursor. Based on the received data the sentence containing the requested word, the word itself, word's sequence number in the sentence and the language are defined. The word sequence number is needed for defining

siblings of requested word which is needed for idioms detection. After this the founded set of text units (words and idioms) with translations and examples of use are returned to the client.

- Method *sentences* supports only GET method. It is used to receive a word or phrase and offset and return the usage examples of this word or phrase from corpus. Offset is needed in case if user asks for next set of usage examples.

dictionary module

dictionary module contains the object-relational mapping models of database tables and logic of user vocabulary. It contains the model *Word* to store words and model *UserWord* for the association of the word with the user.

elastic_control module

This module works with our Elasticsearch server. For communication with the server python library *elasticsearch* is used.

models module

Module *models* contains additional object-relational mapping models such as *Language*, *Category* and *Portal*, which provide mapping to corresponding database tables.

pages module

Because our extension is not distributed over Chrome Web Store but using our own pages, this module is responsible for these web pages.

5.4 Client-side logic

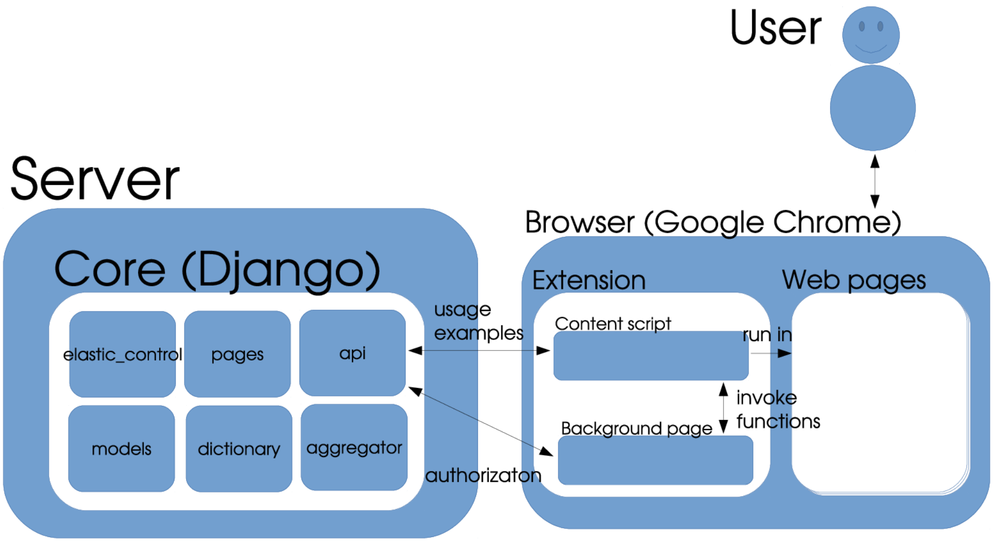


Figure 5.4: Client-side architecture.

In this section the logic of our browser extension and problematic of writing extensions for Google Chrome will be described.

Figure 5.4 contains scheme of our extension, its communication with server and responsibilities of each part.

Chrome Extension consists of several basic parts:

1. Manifest contains description, settings and the rights of our extension.
2. Background page is responsible for communication between the extension parts and it contains general extension logic. In our project this part is responsible for the user authorization, working with a local storage and calling content script functions by means of messages.
3. UI pages are extensions pages, which the user can interact with, such as a pop-up windows in the toolbar.
4. Content script is a script that runs on the selected pages visited by user. It is a core part of our extension and will be described more detailed.

Content script

Content script is used to display server output to the user.

Server output is displayed by using new page element which is served as a window of our extension. All manipulations with extension window is made using javascript framework AngularJS.

After user request the client sends request to the server for obtaining usage example. Request contains the text of clicked DOM element, position of click and user preferable way to order usage examples.

Flow diagram of Content script is presented on the figure 5.5.

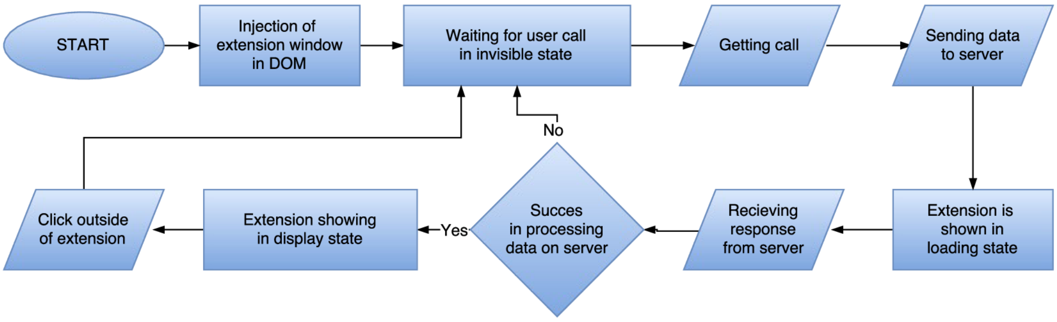

START
Injection of extension window in DOM
Waiting for user call in invisible state
Getting call
Sending data to server
No
Click outside of extension
Extension showing in display state
Yes
Succes in processing data on server
Recieving response from server
Extension is shown in loading state

Figure 5.5: Content script flow diagram.

User has a possibility to visit source page from where usage example was obtained. On this page usage example is highlighted.

5.5 Integration with language learning system

Because our extension is browser extension there is no need for special integration with portal. It only has to be tested on portal pages

As additional feature extension gives an opportunity to evaluate the knowledge level of students which are using extension based on their vocabulary

Also using built corpus portal courses was evaluated for relevancy of used word and for connection to specific area. Based on result some courses were modified

5.6 Additional features

In order to make our extension more universal and improve learning process as much as possible, we have expended the functionality of our extension. The additional features, such as Translation, Detection of idioms and Quality interface were added. These features are described below.

5.6.1 Translation

Our extension will provide user with the translation of the word itself as well as the whole sentence containing requested word and translation of idioms.

There are several services that provide multilingual translating, for example Bing Translator, Google Translate etc.

For translation of text Yandex Translate was chosen due to good quality of translation and clear and easy usage of their API.

Additionally Yandex provides the service Yandex Dictionary which is used for obtaining dictionary translation of the word with several possible translations.

Sentence is translated using Yandex Translate.

Requested word is translated using Yandex Dictionary and in case of failure Yandex Translate is used.

The translation of idioms will be described in the next part.

5.6.2 Detection of idioms

As source of idioms the portal Wiktionary was used. Wiktionary is collaborative project for creating a free lexical database in every language, containing meanings, etymologies and pronunciations. Among other things Wiktionary has a big collection of idioms for different languages with their meanings translation.

Work with Wiktionary is accomplished by using a MediaWiki API. MediaWiki API is a web service that provides convenient access to wiki features, data, and meta-data over HTTP.

The process of idiom detection is shown on the figure 5.6.

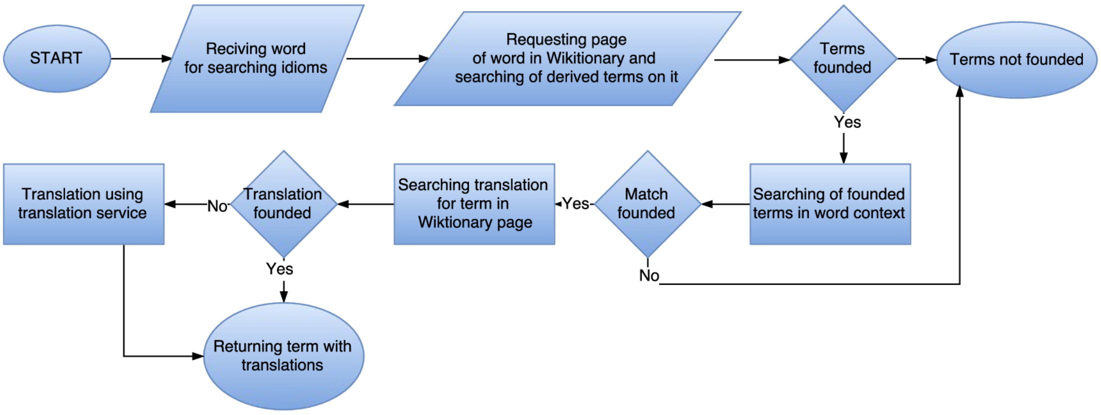

START
Reciving word for searching idioms
Requesting page of word in Wikitionary and searching of derived terms on it
Terms founded
Terms not founded
Yes
Searching of founded terms in word context
Match founded
Yes
No
Searching translation for term in Wiktionary page
Translation founded
No
Yes
Translation using translation service
Returning term with translations

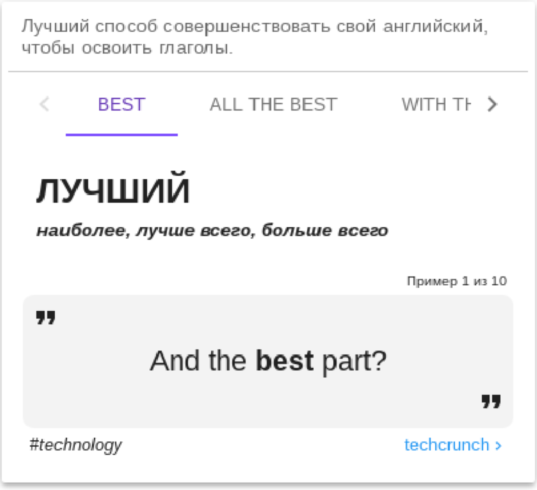
Figure 5.6: Idiom detection flow diagram.

After receiving request for word translation from the client, the request to the Wiktionary containing this word is sent. If page with the word exists, the searching process of derived terms containing the word occurs. If derived terms are found the same terms are searched in word context. If the match is found the translation of the term from Wiktionary page is sent to client. If translation is not found the translation by Yandex Translate is used.

5.6.3 Interface

Interface designs also play an important role in engendering students perceptions of technology-enhanced learning[9]. Interface should be intuitive, minimalist and respond to the latest trends in the design of interfaces.

Interface is represented by the additional page element which is served as a window of our extension. The example of the window is shown on the figure 5.7.



Лучший способ совершенствовать свой английский, чтобы освоить глаголы.

ЛУЧШИЙ

наиболее, лучше всего, больше всего

Пример 1 из 10

And the **best** part?

#technology

techcrunch ›

Figure 5.7: Extension graphic interface.

Our interface includes the following building blocks:

1. Block of translation of a whole sentence.
2. Tabs of possible text units for word in processed context.
3. Block of one or more translations of the word and phrases.
4. Blocks of usage examples for the words or phrases. Also, includes category and link to source article of usage example.

For graphical interface of extension the Material Design is used. This design language was developed by Google and it is used in most of their new products. Using this style our extension will be native for Chrome users. Also our extension will be better combined with SpeakASAP pages, because the portal is made in this style.

In order to implement the graphic interface elements of AngularJS library the Angular Material has been used.

Chapter 6

Evaluation

In this part we will make a comparison between our extension and browser extensions of other language learning portals. The advantages and disadvantages of extension will be mentioned and also the possible future development will be considered.

6.1 Comparison

My extension is created with the cooperation of portal SpeakASAP. In order to understand the benefits and drawbacks of the extension the comparative analysis with browser extensions of other language learning portals is performed.

There are two portals, which have their browser extensions – Lingualeo and Lingua.ly.

	speakasap	lingualeo	lingua.ly
Translation of word and sentence	+	+	+
Pronunciation	-	+	+
Transcription	-	+	-
Usage examples	+	-	-
Idiom detection	+	-	-
Image association	-	+	-
Divide on types of word	-	-	+
Base form of word	-	+	-
Languages	13	1	10

Table 6.1: Browser extensions comparison.

6.2 Survey

In order to analyze the extension deeper the questionnaire was sent to the users of extension. Twenty six users took part in the survey. All questions in questionnaire were divided into several blocks. According to received reviews the following conclusion of evaluation for different features of extension is made.

Graphical interface

All users sent the positive feedback regarding the graphical interface. The following characteristics of interface were assessed: design, intuitivity and convenience of using.

The quality of translation

The questionnaire included the questions about the quality of the translation. The feedback about the translation of the word itself was positive, but there were some claims about the translation quality for the whole sentence.

The usage examples

The special attention in questionnaire was paid to the function of usage examples.

1. 92,3% of users used the function of usage example and mentioned that this function is very useful.
2. 53,8% of users went to the source page of usage example. The users noticed that this function makes them read the full article, which was considered as advantage.
3. Only 84.6% of users understood all examples clearly. Other users remarked that not all examples were clear.
4. 100% of users pointed out that the usage examples were from their preferred area.

General questions

Generally the extension was evaluated positively. All users noticed the usefulness of the extension in learning process of foreign languages, but only 76.9% will continue to use this extension in the future.

6.3 Pros and Cons

On the base of the performed comparative analysis and users survey the pros and cons of our solution have been found out.

Pros of extension:

- Personalization – the extension shows user the examples based on user vocabulary and preferable area.
- Usage example – this function gives user the opportunity for better understanding and memorizing of the new learned word.
- Multilingualism - the extension supports 13 languages and it can be further extended.
- Simplicity - the extension does not require a special knowledge from user and it can be used directly after installation.

Cons of extension:

- The corpus does not contain enough examples and it is needed to be extended in order to cover as many existing words and phrases as possible.
- Currently it does not support the word tagging which is also can be helpful.
- Poor quality of sentence translation
- Some parts of interface not fully clear for users

6.4 Future development

Learning of foreign languages is constantly growing area. That gives us a big opportunity for further improvement. New functionalities of the extension which will improve the learning process of foreign languages can be developed and implemented.

To improve quality of personalization the corpus of usage examples can be further expanded. Thus it will include as much words of particular language as possible including rarely used words. New types of text sources can be added such as books, blogs, etc.

Also the ways to predict readability of usage examples for user can be improved.

In the future the corpus of video or audio data can be added. This will give an opportunity to present the usage examples in video and audio form which will also improve the efficiency of learning process.

We can also use the parallel corpus and give user an opportunity to learn not separated words, but learn equivalent text constructions from native language and learning language.

And last but not least on the base of user survey the following additional features can be added to the extension:

1. Definition of the word
2. Transcription of the word
3. Audio pronunciation of the word
4. Usage frequency of the word

Chapter 7

Conclusion

In our thesis we have studied the foreign language learning problematic including different approaches for using corpora in learning and existing DDL web tools that facilitate learning process.

Based on the results of the analysis and requirements we have designed and implemented the distributed client-server application in form of Google Chrome extension. The main advantage of this extension is personalized usage example functionality, which gives the powerful tool for learning foreign language, understanding foreign texts and expending user vocabulary.

To find optimal solution we have made a study and analysis of different technologies such as searching servers, ORM, relational databases, MVC pattern, web frameworks, material design, etc.

We have paid particular attention to corpus problematic. We have studied its types and ways of usage.

Corpus that is used in our extension already contains 247133 sentences in different languages. This gives us the ability to examine language items in actual language and use it for the improvement of courses quality. Existence of API for communication with corpus can be used in other projects and for other purposes.

We have made the evaluation of our extension, including the comparison with existing solutions and considering pros and cons.

We have also integrated our extension into the existing learning language portal, which brought an additional help to the students of this portal with the learning process of foreign languages.

As a result, our extension proves its usefulness. It has good potential to become wide spread tool for anyone who learns foreign language as well as for different learning language portals.

Bibliography

[1] jusText. [Retrieved 9 May. 2016] http://corpus.tools/wiki/Justext.

[2] Stemming. [Retrieved 9 May. 2016] https://en.wikipedia.org/wiki/Stemming.

[3] You Know, for Search. [Retrieved 9 May. 2016] https://www.elastic.co/guide/en/elasticsearch/guide/current/intro.html.

[4] Lingualeo rolls out new system of online learning of english language, December 2014. [Online; posted 16-May-2016] http://www.rbc.ru/own_business/22/12/2014/549291e39a79476302c47844.

[5] Ken Beatty. *Teaching & researching: computer-assisted language learning*. Routledge, 2015.

[6] Silvia Bernardini. Exploring new directions for discovery learning. *Language and Computers*, pages 165–182, 2002.

[7] Michael Levy. *Computer-assisted language learning: context and conceptualization*. Clarendon Press, 1997.

[8] Ute Romer. *Corpus linguistics*, volume 1. De Gruyter, 2009.

[9] Chris Trevitt. Interactive multimedia in university teaching and learning: Some pointers to help promote discussion of design criteria. Technical report, January 1995. Paper presented at the session on package development at the Computers in University Biological Education virtual conference CITI Liverpool, England, United Kingdom (January 30-February 10, 1995).

[10] Roumen Vesselinov and John Grego. Duolingo effectiveness study. Technical report, Queens College, City University of New York, December 2012. http://static.duolingo.com/s3/DuolingoReport_Final.pdf.

[11] Xiaohui Xu. Research on the application of context theory in vocabulary study. *Theory and Practice in Language Studies*, 3(6):1059–1064, June 2013.

[12] Enes Yilmaz and Adem Soruç. The use of concordance for teaching vocabulary: A data-driven learning approach. *Procedia - Social and Behavioral Sciences*, 191:2626–2630, June 2015.

Appendix A

CD Content

- **server/** – server-side sources, manual and license
- **client/** – client-side sources, manual and license
- **extension/** – packaged extension and user guide
- **thesis/** – tech. report LaTeX sources
- **thesis.pdf** – tech. report