L2 text recommendation system for Russian language

Nikolay Babakov[[1]](#footnote-0), Natalya Isupova[[2]](#footnote-1),Anastasiya A. Bonch-Osmolovskaya[[3]](#footnote-2)

Olga Eremina[[4]](#footnote-3)

**Abstract**

Language learning is a complicated process which includes numerous actions aimed at accumulation of a learner’s real-life language experience. Reading is one of the most important language-learning process, yet searching for texts which will be appropriate for the current language level of a learner is quite a time-consuming task. We propose a system for automatic text recommendation for Russian as L2 learners, which is based on evaluation of learners' language competence.

Though the system has been designed for Russian language, the general principles of the system can be transferred to any language.

**Keywords**: text recommendation, L2 texts reading.

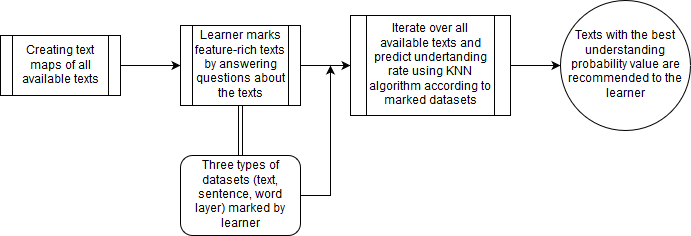
1. **Introduction**

Numerous L2 text recommendations approaches have been developed so far. The classical approach is based on a learner’s passing CEFR-like test and reading texts in accordance with the corresponding CEFR level. This approach is considerably straightforward, therefore it does not take enough text complexity properties into account, and most words positioning in a definite CEFR level raise quite an arguable question.

At the same time, we have found some interesting approaches which used tests for further recommendations (Kurdi, M., 2018), but in that case a hypothetical learner would be supposed to read text and to manually score such parameters as syntaxis, morphology and so on, which would actually be not so easy to evaluate.

We have tried to accumulate all experience of our colleagues and developed L2 text recommendation system which takes into consideration all domains of text reading complexity and provides the most accurate information about user knowledges in each domain which effects further recommendations (see Figure 1 ).

Figure 1. System outline



1. **Method**
   1. **Evaluation of learner’s language competence**
      1. *Textual annotation*

To collect the information about a learner's language competence, we use 10-15 texts with a number of questions related to them. The texts are ranged either within all possible CEFR levels or within the level the learner is supposed to have. There are several questions about each text learner should answer. The questions are designed in such a way that only one definite sentence in the text is enough for a correct answering to a given question.

The texts a test like the mentioned one are selected in accordance with the amount of high frequency specific words each text contains. The ideal test should contain minimum amount of texts with words which are the most typical for the topic inside which we will perform the recommendation.

Therethrough, when the user gives a correct or an incorrect answer to a question, he/she namely puts “+” or “-” mark to the corresponding group of words which shows whether these words and the sentence are understood or not. Such a marking is distributed within all layers of text features and then used for collection of the three datasets which thoroughly demonstrate the learner’s knowledge.

* + 1. *Learners competence datasets collection*

The next step is the collection dataset which will project the learner’s knowledges into three domains the text consists of: words, sentences and overall text features. All datasets’ vectors have target variable which shows whether the element of a text in a corresponding layer is understood correctly or not.

*Collocations and words layer*

We use the collocations and words layer to find out the words or group of words a learner has understood or not when answering the questions.

Each word or collocation inside is marked with 0 (for an incorrect answer) or 1 (for a correct answer) target variable.

*Sentences layer*

Each sentence is represented as a vector, which includes percentage of non-trivial morphological features such as specifical parts of speech (for Russian, such parts of speech as participle) and other complicated language elements(which actually can vary in different languages, as an example, in Russian there are such elements as reflexive verbs or numerous nouns connected with one another by means of different cases and syntax dependence) and mean syntax dependencies. The target variable for each sentence is 1 for a correct answer and 0 for an incorrect one.

*Text layer*

Each text is represented with a text complexity metrics such as LIX (Brown, J., Eskenazi, M., 2005) , type-token ratio and also averaged syntax ratios of all sentences vectors.Target variable is the value within 0 and 1 which represents the percentage of correctly answered questions. For example, if there are ten questions for a text and a lerner answered six of them correctly, the corresponding target variable for this text will be 0.6.

Table 1. Learner knowledge dataset representation

|  |  |  |  |
| --- | --- | --- | --- |
| **element id** | **X** | **target variable** | **comment** |
| WORDS DATASET | | | |
| word\_1 | word2vec(He) | 1 | words from a correctly answered sentence |
| word\_2 | word2vec(travels) | 1 |
| word\_3 | word2vec(alone) | 1 |
| word\_4 | word2vec(They) | 0 | words from an incorrectly answered sentence |
| word\_5 | word2vec(were) | 0 |
| word\_6 | word2vec(foolished) | 0 |
| SENTENCES DATASET | | | |
| sentence\_1 | percentage of difficulty of understanding objects, POS and mean syntax dependencies | 1 | correct answer |
| sentence\_2 | 0 | incorrect answer |
| TEXTS DATASETS | | | |
| text\_1 | LIX, TTR, average sentences properties | 8/10 | percentage of correctly answered questions |
| text\_2 | 6/10 |

* 1. **Text recommendation process**
     1. *Text preprocessing*

Text recommendation process requires availability of all potentially recommended texts parsed to json-file.

The text is preprocessed by means of UDpipe. We need to extract three types of features and create JSON text map which will include all mentioned above features.

* + 1. *Recommendation algorithm*

The recommendation algorithm matches the vectros collected during competence evaluation with the vectors of text analysis. K-Nearest-Neighbors approach is applied to predict the understanding rate for each potentially recommended text.

The recommendation algorithm automatically iterates over each text, extracts same feature levels (words layer, sentences layer, text layer) and searches for each word’s, each sentence’s and text’s features vector’s top-5 most similar (according to the cosine similarity) vectors in the evaluation dataset. The target variables of these top-5 most similar vectors are averaged proportionally to the similarity value and are applied to predict understanding of each word, each sentence of the text, and the potentially recommended text as a whole. The process of matching the variables between two datasets is illustrated in the Table 2.

Table 2. Text understanding prediction overview for all three layers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Dataset type** | **Prediction object** | **Result** | **Prediction result meaning** | **Recommendation reference** |
| words dataset | word | W, +-val [0...1] | the value corresponds to the normalized tf-idf which is positive for correct understanding and negative for incorrect | percentage of correctly understood words values |
| sentences dataset | sentence | S, [0...1] | 0 for non-understanding  1 for understanding | percentage of correctly understood sentences |
| text dataset | text | T, [0...1] | percentage of correctly understood sentences | percentage of correctly answered questions |

As a result, for each analyzed text we get a vector with three numbers ranging from 0 to 1. The result can be illustrated as follows [0...1,0...1,0...1]

It is assumed that for good understanding of the text it is necessary to be familiar with 80% of the information or grammar rules. Therefore, the closer each calculated value to 0.8, the more likely the texts are recommended. We call this value Understanding Deviation and mark it as UD. The formula for calculating UD is shown below:

(1)

where

W - percentage of words in an examined text with correct understanding prediction

S - percentage of sentences in an examined text with correct understanding prediction

T - percentage of abstract text-related questions with correct answer prediction

FInally, when the analysis of al the text is finished we get the UD corresponding to each text in a text database. The texts are sorted by value, and the three texts with the least standard deviation are recommended for a potential the learner.

1. **Model performance evaluation**

We employ two different approaches to evaluate of the recommended text understanding. The first one provides text related questions, referring to several definite sentences in the text given. The second approach suggests asking the user to mark the sentences which include the biggest amount of any predefined features within some scale like “Not understand this sentence at all … Understand the general sense … Fully understand”. Choosing an approach depends on time availability of test designers. The first approach which implies questions to the recommended texts is more time consuming because the questions are supposed to be prepared manually.

After applying any of the named feedback model, we get the same knowledge datasets but they have different application. They are used for model performance evaluation.

The model is supposed to work well if the examinee answers 80% of questions related to the recommended text or marks 80% of sentences from the text as “Fully understand”.

The model itself can be tested by providing learners with the recommended texts, collecting their answers and calculating standard deviation from 0.8.

Thus, the formula is the same with the previous one but we handle not predicted but real values.

(2)

W\* - percentage of correctly understood words’ tf-idf normalized value in recommended text

S\* - percentage of correctly understood sentences in recommended text

T\* - percentage of correctly answered text related questions

Obviously, the less UD\* value we get, the better this system has worked. That is why the final metric of this is R-square-like (let’s name it UR-square which stands for Understanding Rate) and is calculated, using the following formula:

(3)

For the time being, we have performed tests with 20 learners (each of them got three texts recommended for reading), and the UR-square achieved turned out to be 0.75.

1. **Conclusions**

In the current paper we have presented the new approach for L2 text recommendation according to learners skills which are checked using most natural way - texts questions answering.

The next steps will be applying more tests with learners of different backgrounds, applying the approach to other languages and tuning the test parameters to reach higher accuracy.

**References**

Brown, J., Eskenazi, M. (2005) *Student, text and curriculum modeling for reader-specific document retrieval.* Proceedings of the IASTED International Conference on Human-Computer Interaction, Phoenix

Kurdi, M. (2018). *A reading recommendation system for ESL learners based on linguistic features*. Proceedings of the 31th International Florida Artificial Intelligence Society Conference, Melbourne, Florida

1. Nikolay Babakov, Moscow, Russian Federation, bbkhse@gmail.com [↑](#footnote-ref-0)
2. Natalya Isupova, Moscow, Russian Federation, nata\_isupova@inbox.ru [↑](#footnote-ref-1)
3. Anastasiya A. Bonch-Osmolovskaya, Moscow, Russian Federation, abonch@hse.ru [↑](#footnote-ref-2)
4. Olga Eremina, Moscow, Russian Federation, oeremina@hse.ru [↑](#footnote-ref-3)