L2 text recommendation system for Russian language

Nikolay Babakov

Department of Humanities, National Research University Higher School of Economics, Moscow, Russian Federation
International online language-learning service "Lingualeo"
bbkhse@gmail.com

1. Overview

Language learning is a complicated process which includes many actions aimed to develop learner's real-life language experience. Reading is one of the most important language-learning process, but searching texts which are appropriate for the current level of the learner is a very time consuming task. We propose a system for automatic text recommendation for Russian as L2 learners, 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.

2. System outline

1. Collect texts corpora related to some special topic and generate json text-map describing three text domains (overall text features, sentences features and words features)
2. Learner answers the questions about 10-15 texts
3. Algorithm generates user knowledge model based on three domains in accordance with the answers to the questions
4. KNN algorithm uses generated knowledge model and text maps (designed in the same dimensionality) and looks for top similar texts which are recommended to a learner

3. Text prepocessing

Text recommendation process requires having all potentially recommended texts parsed to json-file. We extract three types of features and create JSON text map which includes all of these features

```
{
    #text features
    "LIX":value,
    "Type_token_ratio":value,
    "sentence_map":
            #sentence features
            [{"special_sentence_features":
            {"complicated_language_objects":value,
            "complicated_POS":value,
            "mean_syntax_dependencies":value}},
            "sentence_words":
                    #word features
                    [{"original_word":word,
                    "lemma":lemma,"importance"tf_idf,
                    "grammar_prop":POS, "lex_vector":w2v},
                    {...}],
        {...}]
}
```

4. Generation of learners knowledge model

4.1 Collecting datasets

The learner is provided with numerous texts and he/she is supposed to answer questions related to these texts. Each question corresponds to some sentence in each text. So answering these questions namely provides us with the manually marked datasets indicating learner's knowledge.

For example we can have the following text to be indexed by the learner:

He travels alone. He was foolished

So we can provide the learner with the questions about each sentence, for example

1. ***How did he travel?***
2. ***What happened to him?***

4.2 Indexation of datasets

Let's say that first question has been answered correctly and second one has been answered incorrectly. According to this information we perform indexing of the words, sentences and the whole text the questions were targeted to. Sample dataset is as follows:

Element id	Initial data	Target variable	Comment
WORDS DATASET			
word_1	word2vec(He)	1	words from correctly answered sentence
word_2	word2vec(travels)	1	
word_3	word2vec(alone)	1	
word_4	word2vec(He)	0	words from incorrectly answered sentence
word_5	word2vec(was)	0	
word_6	word2vec(foolished)	0	
SENTENCES DATASET			
sentence_1	complicated language objects vector	1	correct answer
sentence_2		0	incorrect answer
TEXTS DATASETS			
text_1	LIX, TTR, averaged sentences properties	1/2	percentage of correctly answered questions

Tabela 1: *Learner knowledges dataset*

5. Text recommendation process

On this stage we namely have three datasets representing learner knowledge in three domains and 3*N vectors (where N is the quantity of texts in our corpora) which describe each text in same three domains. This lets us apply KNN algorithm and find top similar to learner's knowledge texts

Dataset type	***Prediction object***	***Result***	***Recommendation reference***
words dataset	word	W, [0...1]	% of correctly understood words values
sentences dataset	sentence	S, [0...1]	% of correctly understood sentences
text dataset	text	T, [0...1]	% of correctly answered questions

Tabela 2: *Understanding predictions interpretation*

After applying KNN each text gets three predictions which illustrate understanding probability for each domain as shown in a table below

	Words understanding prediction, W	**Sentence understanding prediction, S**	**Text understanding prediction, T**
text 1	0.7	0.8	0.6
text 2	0.7	0.4	0.3

Tabela 3: *Model predictions example*

It is assumed that for good understanding of the text it is necessary to be familiar with 80 percent of the material or grammar rules. So the closer each calculated value to 0.8 the more likely the text is recommended. We call this value Understanding Deviation and mark it as UnDev.

$$UnDev = \sqrt{\frac{(W-0.8)^2 + (S-0.8)^2 + (T-0.8)^2}{3}} \quad (1)$$

When all texts are analyzed there is UnDev corresponding to each text in a text database. The texts are sorted by value and three texts with the least standard deviation are recommended to the learner.

6. Evaluation and current results

There are several approaches for evaluating the model results, One of them is asking the learner to mark sentences from recommended texts within some scale like "Don't understand the sentence at all . . . Understand general sense . . . Fully understand"
The model is assumed to work well if the learner marks 80% of sentences from the text as "Fully understand"
The model itself can be tested by providing learners with the the questions to recommended texts, collecting their answers and calculating standard deviation from 0.8.

$$UnDev* = \sqrt{\frac{(W*-0.8)^2 + (S*-0.8)^2 + (T*-0.8)^2}{3}} \quad (2)$$

In this formula we refer to the same values as in the formula 1 but here the values are calculated from real results after the learner gets the recommended texts and mark them according to his/her understanding.
Obviously, the less UnDev* value is the better this system has worked. That is why the final metric is as follows

$$UR^2 = 1 - \sqrt{\frac{(W*-0.8)^2 + (S*-0.8)^2 + (T*-0.8)^2}{3}} \quad (3)$$

For now we have performed tests with 20 learners (each of them got three texts recommended for reading) and the UR^2 was 0.75