Techniques for educational items projection

Bachelor’s Thesis

Daniel Homola

Brno, Spring 2018
This is where a copy of the official signed thesis assignment and a copy of the Statement of an Author is located in the printed version of the document.
Declaration

Hereby I declare that this paper is my original authorial work, which I have worked out on my own. All sources, references, and literature used or excerpted during elaboration of this work are properly cited and listed in complete reference to the due source.

Daniel Homola

Advisor: doc. Mgr. Radek Pelánek, Ph.D.
Acknowledgements

I wish to express my sincere gratitude to my advisor, doc. Mgr. Radek Pelánek, Ph.D., for providing me with such an interesting topic to research, and his guidance. I am also very grateful to my family for the support in my studies.
Abstract

This thesis studies the problematics of item similarities in educational systems and visualization of these items based on their similarities. First, we describe various measures that use learners’ performance data or text of the items for computation of the similarities. Second, we evaluate these measures mainly using results from clustering and visualization techniques, where the computed similarities are used as input. Then, we compare and study two different techniques used for data projection, which can provide us valuable insights into the data of the systems. The measures and the visualization techniques are evaluated on the data from the system Umíme česky. The results show that t-SNE is a good technique for projection of data and that the appropriate choice of similarity measure is crucial.
Keywords

educational data mining, similarity measures, natural language processing, data projection, t-SNE, PCA, Umíme česky
Contents

1 Introduction ................................................. 1

2 Educational data ........................................ 4
   2.1 Umíme česky ........................................... 4
   2.2 Problem description ................................... 5
   2.3 Data description ....................................... 6

3 Similarity measures ..................................... 7
   3.1 Measures based on learners’ performance data .... 8
   3.2 Measures based on educational items’ text ....... 11

4 Dimension reduction ................................... 16
   4.1 Principal component analysis ....................... 16
   4.2 t-distributed stochastic neighbor embedding .... 17

5 Evaluation .............................................. 21
   5.1 Data suitability and labeling ....................... 21
   5.2 Comparison of similarity measures ............... 24
   5.3 Visualizations ....................................... 33

6 Conclusion .............................................. 40

Bibliography ................................................ 41

A Contents of the electronic attachment ............. 44

B Manually labeled data .................................. 45
1 Introduction

With the rise of IT and availability of the Internet, the popularity of e-learning is increasing and people can learn something online in various ways. One of the ways is adaptive e-learning, where the educational system adapts to the user and users learn more efficiently.

Computer systems that customize the learning experience are called Intelligent tutoring systems (ITS) [1]. These systems do not use a one-size-fits-all strategy. They deliver the content to learners by using the data collected about them during the teaching session.

From the standpoint of education, there are many motivating factors why to customize the learning experience. Learners can save time by doing exercises that are not part of their preexisting knowledge; the system can offer hints to help the student understand the material; etc.

Moreover, it can also help teachers. The system saves the teachers’ time because they do not have to spend time evaluating students’ homework. In addition to that, it can provide data to teachers and developers looking to refine teaching methods or questions asked in the system.

Related work and aim of the research

Adaptive practice systems, whose priority is individualized learning, are also a subset of ITS. Systems like these are developed and maintained in the Adaptive Learning group1, a small research lab at the Faculty of Informatics, Masaryk University. One of these systems is the source of educational data for this thesis.

As mentioned above, the provision of data from systems to teachers and developers can be loosely coupled with the refinement of the teaching session. Thus it is beneficial to use data mining for getting some information out of the collected data, e.g., measuring the similarity of items. Measuring the item similarities can help to define or refine knowledge components, detect duplicates or outliers, or identify missing items. Adding to that, the similarities of items can serve as

the input to clustering or visualization techniques, which can simplify the process of the previously specified tasks.

This problem of educational items’ similarity was previously studied by Radek Pelánek and Jiří Řihák from the Adaptive Learning group [2, 3]. Their focus was on methods for computing item similarities based on learners’ performance data. The data have binary information about whether the questions were answered correctly or incorrectly. Based on these data they computed similarity for each pair of items and showed the best-performing similarity measures for our type of data.

Additionally, in the items from the system Umíme česky whose data we use in this thesis, we have an additional property – text of the item. Therefore, besides the best-performing measures based on learners’ data from previous research, we also utilize similarity measures that work with the text of the items.

Furthermore, our goal is to have similar items together in a cluster or close to each other in a visualization in two-dimensional space (projection of items onto a plane). Hence, the computed similarities are used as input into clustering or visualization techniques, and the results from these techniques are the basis for the evaluation part. Example of such a visualization is in Figure 1.1.

Since the quality of a visualization is difficult to measure and requires much effort to be estimated, the performance of similarity measures is also quantified by the results from clustering technique.

Even though the quality of a visualization is hard to quantify, the visualizations created by two different techniques are critically evaluated. The technique t-SNE which had the best outcomes in the previous research is studied comprehensively.

Structure of the thesis

The thesis is divided into six chapters:

- **Chapter 1 – Introduction**;
- **Chapter 2 – Educational data** is more about the problem description for particular data from educational system Umíme česky. Furthermore, the dataset used in this thesis is described more in-depth;
1. Introduction

- **Chapter 3 – Similarity measures** presents the previous research done in the Adaptive Learning group related to the educational items’ similarity. In addition to that, other techniques for measuring the similarity of data are introduced;
- **Chapter 4 – Dimension reduction** introduces and compares two main techniques for dimension reduction, which are used for data visualization in two-dimensional space;
- **Chapter 5 – Evaluation** contains discussion about issues related to data for this task, comparison of similarity measures and an application of dimension reduction techniques on educational item similarities. The evaluation of the results is reported;
- **Chapter 6 – Conclusion** summarizes the results of this thesis and gives an idea for future work.

Figure 1.1: Example of desired word projection, which was created manually. Similar items are in visually separated clusters. The items are grouped according to our opinion and are colored for easier interpretation. In a real educational system we would not know the target groups and the colors.
2 Educational data

This thesis uses educational data to improve educational systems. The systems collect data from learners, and based on gathered data about learners’ performance we can make assumptions about relations between items. These data are used to provide a better insight into item properties.

Such task is a part of a discipline called educational data mining, and its objectives are wide [4]. One of the objectives is the adaptability of a system to teach a particular student, and another one is the management of questions asked in exercises. The second mentioned is partially related to this work.

2.1 Umíme česky

Educational data that were used in this work come from the system Umíme česky. The system has many different exercises, which focus on the practice of Czech orthography and grammar. The practice of Czech orthography and grammar is most useful for Czech students attending elementary school but some exercises are also suitable for high school students and adults. They can entertain themselves and improve their knowledge of the Czech language.

Czech schools use this system as a supportive part of teaching. For example, the system shows the level of topic mastery for a learner based on the correctness of answers, and the level of mastery helps teachers with assigning homework. Then, homework for students can be formulated as – “Reach mastery in selected topics.” [5, 3].

This educational system contains lots of items (questions, problems), which belong to a certain topic. With a large number of items in the system, there are suitable items for different types of students. But along with the suitability of items, it is hard to manage all the items.

2. Mastery learning is a strategy of learning that demands the students to achieve a level of mastery in prerequisite knowledge before proceeding to more advanced topics.
2. Educational data

2.2 Problem description

The problem lies, as suggested in the introduction, in concept definition or refinement, detection of duplicates or outliers, or identification of missing items.

A concept (also known as knowledge component) in an educational system is formed by a set of questions and trains a specific part of knowledge. Related to the system Umíme česky, a concept practices specific grammatical rules.

Concepts are arranged in a tree-like structure, and thus some concepts are coarser and contain more sub-concepts. For example, concept I/Y Vyjmenovaná slova consists of many sub-concepts, which are specified by filling letters i and y after a certain letter.

Questions are mapped to concepts, and exercises are composed of questions. Example of such an exercise within the concept Vyjmenovaná slova B, which is included in I/Y Vyjmenovaná slova, is shown in Figure 2.1.

![Figure 2.1: Practice of filling in i/y.](image)
The bar on top is the mastery.

With many concepts and large dataset, the management of items (questions) is difficult. Therefore, it is useful to analyze item similarities, which can be computed in various ways. Afterward, the similarities can be used as input into clustering or visualization techniques. From visualizations, teachers or system developers gain knowledge,
which can help with the management of items. If too many items are similar within a concept, some of them can be moved or deleted; if an item is not in a cluster with other items, there might be a lack of items practicing the grammatical phenomenon of that item, or the item should be moved to another concept; etc. Hence, item similarities and reasonable (to some extent) visualizations are the main problem studied in this work. The resulting visualizations are projections of items onto a plane, which provide valuable insights into the data of the system Umíme česky.

2.3 Data description

In this work, the data from fill-in exercises of Umíme česky were used, in which the student has two options how to fill in the missing letter or letters in a word or a phrase.

Here, important attributes of answers and questions are described:

- **Answers** contain information about the correctness of the answer, which is binary information; and response time value.

<table>
<thead>
<tr>
<th>id</th>
<th>user</th>
<th>question</th>
<th>correct</th>
<th>responseTime</th>
</tr>
</thead>
<tbody>
<tr>
<td>282</td>
<td>1338213</td>
<td>97</td>
<td>1</td>
<td>450</td>
</tr>
</tbody>
</table>

- **Questions** contain the phrase, in which learners are asked to fill in the missing letter or letters; the correct missing value; and distractor value, which is an incorrect answer option.

<table>
<thead>
<tr>
<th>id</th>
<th>question</th>
<th>correct</th>
<th>distractor</th>
</tr>
</thead>
<tbody>
<tr>
<td>97</td>
<td>b_linkář</td>
<td>y</td>
<td>i</td>
</tr>
</tbody>
</table>
3 Similarity measures

*Similarity measure* in this thesis is a numerical measure of how alike two items are (higher value means higher similarity).

It was already explained before why it is beneficial to compute similarities between items in our educational systems. This chapter presents techniques used for this task, and in Chapter 5 the techniques are evaluated and compared.

The process of how similarities are computed and used is as shown in the following Figure 3.1. At first, we measure the similarity for each pair of items, and the computed similarities construct a symmetric *item similarity matrix*. Subsequently, we compute clusters of items or project items onto a plane using the matrix.

![Diagram of the process of computing and using item similarities.](image)

Figure 3.1: Process of computing and using item similarities. Adapted from [2].

For computing item similarity matrix we can use learner data but also other information such as the text of questions, or input from domain experts. Here, we present measures that compute the similarities based on learners’ performance data or educational items’ text.
3. Similarity measures

3.1 Measures based on learners’ performance data

This section briefly describes the previous research on computing item similarities, where the focus was on using learners’ performance data [2].

The item similarities are computed using a matrix L x I, where L is the number of learners and I is the number of items. This matrix often has many missing values.

Learning of a student is ignored because the learning is slow and items are presented in a randomized order. Therefore, the learning does not matter as much as it should in other cases – e.g., when items are presented in a fixed order or learning is fast [2].

Although we have available the correctness of the answer and the response time, only the correctness is used for computing the item similarity matrix in this approach. The response time is dependent on the text length and difficulty of an item. Moreover, all answers have a 50% guess factor. Thus response times are noisy and their utilization is difficult.

Overview

There are many similarity measures that can be used for computing similarity on binary data (correct/incorrect) of learners’ answers on items [6]. In the previous research of Radek Pelánek and Jiří Řihák, they chose 6 of the measures and compared them [2]. They covered the measures that had good results in the previous work and also measures from the related work.

With dichotomous data, the binary similarity can be defined by a 2 x 2 contingency table (see Table 3.1). In the contingency table, there are two items i and j, which are represented by a binary vector based on the learners’ performance data. As shown in the table: a is the number of items where the values of i and j are both 0 (incorrect answer); b is the number of items where the values of i and j are 0 and 1; c is the number of items where the values of i and j are 1 and 0; and d is the number of items where both i and j are 1 (correct answer). $a+b+c+d$ equals to $n$, which is a total number of values in the binary vector [6].
3. Similarity measures

Some similarity measures are asymmetric in regard to 0 and 1 values. In educational systems, incorrect answers have higher importance because they are less frequent. Therefore, in the previous research the focus in asymmetric similarity measures was on incorrect answers (value $a$) [2].

Table 3.1: Contingency table for two items.

<table>
<thead>
<tr>
<th></th>
<th>item i</th>
<th>incorrect</th>
<th>correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>item j</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>incorrect</td>
<td></td>
<td>$a$</td>
<td>$b$</td>
</tr>
<tr>
<td>correct</td>
<td>$c$</td>
<td>$d$</td>
<td></td>
</tr>
</tbody>
</table>

Similarity measures used in previous research

The 6 chosen measures were (measures are called by names of researchers who proposed them): Yule, Pearson, Cohen, Sokal, Jaccard, and Ochiai. Table 3.2 lists definitions of these 6 binary similarity measures.

Table 3.2: Similarity measures definitions based on the contingency table (Table 3.1).

<table>
<thead>
<tr>
<th>Measure</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yule</td>
<td>$S_y = (ad - bc)/(ad + bc)$</td>
</tr>
<tr>
<td>Pearson</td>
<td>$S_p = (ad - bc)/\sqrt{(a + b)(a + c)(b + d)(c + d)}$</td>
</tr>
<tr>
<td>Cohen</td>
<td>$S_c = (P_o - P_e)/(1 - P_e)$</td>
</tr>
<tr>
<td></td>
<td>$P_o = (a + d)/n$</td>
</tr>
<tr>
<td></td>
<td>$P_e = ((a + b)(a + c) + (b + d)(c + d))/n^2$</td>
</tr>
<tr>
<td>Sokal</td>
<td>$S_s = (a + d)/(a + b + c + d)$</td>
</tr>
<tr>
<td>Jaccard</td>
<td>$S_j = a/(a + b + c)$</td>
</tr>
<tr>
<td>Ochiai</td>
<td>$S_o = a/\sqrt{(a + b)(a + c)}$</td>
</tr>
</tbody>
</table>
3. Similarity measures

Second level of similarity measure

Another useful idea from the previous research is "second level of item similarity", which causes the reduction of noise in the data by using more information.

Firstly, item similarities are computed using measures described in the previous section. Then the "second level of item similarity" takes the computed similarity matrix and uses Euclidean distance or Pearson correlation on each pair of item similarities from the similarity matrix. Similarities computed this way use information on all items, not just the pairwise information on items whose similarity is computed (Figure 3.2).

![Figure 3.2: Schema of how the “second level similarity” is applied in the process of similarity matrix computation. Taken from [3].](image)

In the first approach, items are similar when learners’ performance on these items is similar. In the second approach – with the second step of item similarity, we consider two items similar when they are similar in regard to other items. This reduces the noise in the data [2].

From a more technical perspective, from the second step (in particular using Euclidean distance) we can acquire a measure that is a distance metric satisfying triangle inequality. Therefore, it can be then used in some clustering algorithms, which might require the previously stated property [2, 3].
Essential results from previous research

Comparison and evaluation of similarity measures performance were done by computing correlations among similarity measures and by assessing the quality of clusterings because the quality of a visualization is hard to quantify (keep Figure 3.1 in mind). The most important results are listed below:

- Pearson, Yule, and Cohen measures have better results than Ochiai, Sokal, and Jaccard measures.
- Using the second step of item similarity is advantageous.
- Pearson correlation coefficient is a good “default choice”.
- The amount of available data is crucial.

The reasoning is comprehensively described in [2]. Some of the techniques used for the reasoning are also used for comparison of newly proposed techniques to the “best” similarity measure from the previous research later in Chapter 5.

Best-performing similarity measure from previous research

There is not much of a difference in choosing among Pearson, Yule, or Cohen in the first step, and choosing between Euclidean distance and Pearson in the second step [2]. Thus the best practical choice is using Pearson correlation coefficient. One of the reasons why to choose Pearson is also the availability of Pearson correlation coefficient in almost all computational libraries. Therefore, Pearson is considered to be the best similarity measure based on the learners’ data throughout this thesis, which is later compared to newly suggested similarity measures.

3.2 Measures based on educational items’ text

In this section, we propose another approach for computing item similarities. The basis of the approach is the utilization of items’ text. Suggested measures compute the similarity for each pair of items and create the similarity matrix.
3. Similarity measures

The main difference between this approach and the first approach – which is based on learners’ data – is the fact that the only requirement for this approach is the availability of item’s text. For this reason, there is no problem of having not enough collected data from learners of the educational system. Additionally, in some systems, we should consider the learning of the student, which also does not influence this approach. Hence, the item similarity matrix is probably less noisy than the item similarity matrices from the first approach.

When using this approach, it is important to note that although some questions in the system contain more than just one word, only the word with missing letter or letters is taken into consideration when computing the item similarities. The word into which the correct missing value is inserted is the most important one. This word has the main impact on practicing the specific grammatical rule and the major similarity we are looking for.

Overview

Words can be similar in a lexical way and in a semantical way. *Lexical similarity* means that words have a similar character sequence, *semantical similarity* means the likeness of their meaning and that words have common “characteristics”.

In the fields of text mining and natural language processing, various text similarity approaches can compute the similarity between words. Based on a survey of text similarity approaches [7] we distinguish algorithms for lexical and semantical similarity:

- lexical similarity
  - *String-Based algorithms* – operating on string sequences and a composition of characters,

- semantical similarity
  - *Corpus-Based algorithms* – measuring the similarity using information exclusively derived from large corpora,
  - *Knowledge-Based algorithms* – determining the similarity using information drawn from semantic networks – e.g., WordNet [8], which can be thought of as a large electronic
3. Similarity measures

dictionary. It groups English words into sets of synonyms called synsets.

In this work, measures from both areas of text similarity are used: one measure related to lexical similarity – *Edit distance*, and another related to semantical similarity – *Word2vec*. In the following sections, the measures are explained.

**Edit distance**

Many techniques for computing lexical similarity deal with the problem of string matching allowing errors, also called *approximate string matching*. Lots of these techniques are discussed in [9], and we have chosen one of the most commonly used measures for this task – *Levenshtein distance* [10] (also called “edit distance”). Edit distance allows operations of insertion, deletion, and replacement, and in its simplified definition, all the operations cost 1. Then, our problem is the minimal number of edits to transform one word into the other (see Figure 3.3).

![Figure 3.3: Example of edit distance between words “hřejivý” and “křivý”. “d” represents deletion, “s” represents substitution, and the total cost with our definition of the distance is 3.](image)

Afterward, the normalized Levenshtein similarity can be calculated from the distance as:

\[
S_{lev} = 1 - \frac{\text{distance}(a, b)}{\max(a.length, b.length)}
\]

Furthermore, there might be some cases where the value of Edit distance is too high and it has no significance for us – that value already tells us that they are not similar, and hence the similarity should
be 0. Thus we can add a threshold $M$ into the formula, if we want to constrain the result of the distance between words to a certain value:

$$S_{lev\_threshold} = 1 - \min(distance(a,b), M) / M$$

**Word2vec**

Word2vec\(^1\) is a group of models that attracts a lot of attention in the recent years. It was first introduced in 2013 \([11]\) and falls into the category of *word embeddings*. Word embedding is the name for natural language processing techniques where the words are represented in vector space.

Vector space models embed words in a continuous vector space where semantically similar words are mapped near to each other. To these ends, the *distributional hypothesis* states that words that are used and occur in the same contexts indicate similar meanings \([12]\). When the usage of the word defines its meaning, it can be summarized by the underlying idea of John Firth: “You shall know a word by the company it keeps!” \([13, p. 11]\).

The idea of this vector representation is the retention of semantical relations between words. Adding to that, as the algorithm learns “features” about each word – these features also capture syntactic relations. That is the advantage of taking a large corpus of texts, looking at each word, embedding it and predicting its neighbors.

Word2vec incorporates two models – *Continuous Bag-of-Words* and *Continuous Skip-gram*. You can choose whichever you want. More about the models can be found in publications \([11, 14]\).

Since word2vec is one of the state-of-the-art tools for measuring syntactic and semantic word similarities, it is the second technique used for the computation of items’ text similarity in this work.

**Application of word2vec**

Although we do not need data from learners for computing the item similarities, we need a word2vec model trained on Czech corpus because our text data are in the Czech language. Some factors influence the quality of word vectors that are based on a particular model \([11]\):

\(^1\) [https://code.google.com/archive/p/word2vec/]
3. Similarity measures

- amount and quality of the training data
- size of the vectors
- training algorithm

Even though we have not found publicly available and well-performing word2vec model for the Czech language, we created our own model using latest Czech Wikipedia dump available online\(^2\). It is not a huge dataset but for our purposes it is enough to test whether it is a good approach for computing similarity between educational items. Its disadvantage is that some words from the educational system Umíme česky are not in the resulting word vectors of the model. Examples of words, which aren’t in the model – “přežvykoval”, “podkrkonošští”, “nepřebývaly”, etc.

Python library Gensim\(^1\), which contains word2vec, is used in this thesis. The library calculates the similarity between the vectors by Cosine similarity (also used in original papers \([11, 14]\)) and uses Continuous Bag-of-Words model for the training by default. Considering that tuning the hyperparameters is a complex task and it is not the main problem of this thesis, mostly default parameters are used for the word2vec model.

Similarity matrix can be created by computing the similarity for each pair of vectors (a vector is now the representation of an item). Then, the projections can be built on that similarity matrix but they can also be constructed directly from the word vectors (see Figure 3.4).

![Figure 3.4: Projection construction from word vectors.](https://dumps.wikimedia.org/cswiki/latest/cswiki-latest-pages-articles.xml.bz2)
4 Dimension reduction

One of our goals is to visualize data onto a plane based on the computed similarities from the previous chapter. But first, we need to lower the dimension of the item similarity matrix (Figure 4.1).

Here, the dimension of the data is the number of items that are compared to other items. Since there are as many dimensions as there are items, in our case it means that, e.g., in a dataset of a concept containing 100 items, the space (similarity matrix) is 100-dimensional.

To get to two-dimensional space, dimension reduction techniques can be utilized. Reducing the number of the dimensions positively influence many activities – performance improvement of other machine learning algorithms, removal of noisy and redundant features, a decrease in computational complexity, data compression or data visualization. In our work we use dimension reduction techniques for data visualization (projections into two dimensions).

There are many dimension reduction techniques [16]. We describe two main approaches related to our task.

4.1 Principal component analysis

Principal component analysis (PCA) [17] is one the most frequently used method for tasks of dimension reduction. It is an unsupervised statistical technique used to examine the interrelations among a set of variables in order to identify the underlying structure of those variables.

PCA finds directions in the data that maximizes the variance along the data points. These directions are called principal components. Each principal component, or dimension – found by PCA – is a linear combination of the variables in the data. Therefore, we always use 2 principal components.
components for our task because we want projections – 2 dimensions. Adding to that, in this process the original set of possibly correlated variables is converted into a set of linearly uncorrelated variables.

First principal component has the largest possible variance. Then, each succeeding principal component has the greatest possible variance while being orthogonal to preceding principal components. The fact that they are orthogonal is also the reason why the components are uncorrelated.

The process of finding the principal components is further described in [18, p. 375-377].

### 4.2 t-distributed stochastic neighbor embedding

T-distributed stochastic neighbor embedding (t-SNE) [19] is still quite a modern method, even though it was introduced in 2008. It is used mainly for the visualization of high-dimensional datasets and its popularity is rising (Figure 4.2).

The idea behind t-SNE is to preserve the distances from the original space in the lower dimensional space. Put simply, if two data points are close together (similar) in high-dimensional space, we want them to be close in the 2D map too.

For that reason, t-SNE works as follows: Firstly, the distances (similarities) are transformed into a probability distribution over pairs of high-dimensional data points. Secondly, it defines a similar probability distribution over low-dimensional data points from the constructed map, and optimizes the result so that the two distributions are as similar as possible. Note that t-SNE does not retain distances but probabilities.

![Figure 4.2: Rising tendency of t-SNE citations. Taken from scholar.google.com.](image-url)
4. Dimension reduction

This method has been chosen as a cornerstone in most of the resulting projections in Chapter 5 because it has significantly better results than other existing methods in the task of creating two-dimensional maps [19]. The maps (visualizations) using t-SNE are often much more readable than the maps using other techniques. Furthermore, it had better results in the previous research of Jiří Řihák from the Adaptive Learning group on the same data [3, p. 63]. Another reason is also that it works on a different principle than PCA (see PCA vs. t-SNE below).

Some of the drawbacks of t-SNE are the specific settings of parameters needed to get good results, and the non-determinism. The non-determinism means that you do not get the same output each time you run it, although the results are expected to be similar. How to determine the t-SNE parameters is discussed on the next page.

PCA vs. t-SNE

The approach to handling item similarities from the dataset is very different in PCA and t-SNE. They work on distinct fundamentals and that is why we discuss the differences between them.

Our objective can be simply explained. We want distances in the low dimensional map to reflect the similarities in the original high-dimensional data. As it was already stated, PCA maximizes variance and thus is mainly concerned with preserving large pairwise distances in the map. In other words, it wants the dissimilar items to end up far apart. But the question is whether it is what we want from a visualization because large pairwise distances in the data are probably unreliable.

Some techniques focus on preserving large pairwise distances as well as PCA, for example, classical multidimensional scaling [20]. But many techniques aim to preserve smaller pairwise distances in the map as well as t-SNE, such as Isomap [21] or Locally Linear Embedding [22] (some of them are described and compared to t-SNE in [19]).

Ultimately, t-SNE captures not only the local similarities of the high-dimensional data points but also the global point of view, e.g., the presence of clusters.
4. Dimension reduction

**t-SNE parameters and hints**

To make t-SNE perform well, many hyperparameters have to be set. Setting of those parameters is specific for each dataset, and therefore it has to be explored first. After successful exploration, it can give us the results that make sense. To achieve good results – it is vital to learn how to set the parameters of t-SNE, and also interpret and analyze plots from t-SNE [23].

Three most important parameters are:

- **perplexity**
  - The most crucial parameter, which can also be defined as a number of effective nearest neighbors for each item.
  - For a good performance of t-SNE, the perplexity requires tuning.
  - Balancing perplexity value is also loosely coupled with global aspects of the data in the visualization.

- **learning rate (also known as “epsilon”)**
  - If the learning rate is too high, all points are in a rounded shape and relatively equidistant from its nearest neighbors.
  - If it is too low, the result can be a dense cloud of points.

- **number of iterations (for the optimization)**
  - We should always iterate until the plot is stabilized.

General pieces of advice for t-SNE are:

- Typical values for the perplexity range between 5 and 50 [19] but with larger or denser dataset it can be higher.

- Distances between clusters should generally not be taken into account.

- T-SNE tends to expand denser regions of data, thus the cluster sizes (the space they bound, not the number of points) are not important [23].
4. Dimension reduction

- If you use the same data and perplexity, you can run t-SNE many times and select the solution with the lowest divergence of the two probability distributions.

- Training for different datasets demands different numbers of iterations to stabilize the plot.

Python machine learning library *scikit-learn* [24] offers implementation of *TSNE*[^1] and is used in this thesis.

5 Evaluation

Similarity measures from Chapter 3 and techniques for projection of data from Chapter 4 are evaluated in this chapter on the Umíme česky dataset. At first, we describe the choice of concepts used for evaluation, then we move on to a comparison of similarity measures and examine the performance of t-SNE and PCA on these concepts.

5.1 Data suitability and labeling

Firstly, we choose some concepts from Umíme česky, and then we label them for further analysis and comparisons. The choice of concepts is influenced by many issues related to the computation of item similarities.

Issues related to computation of similarities

When computing item similarities based on learners’ data, we should ask ourselves if we have enough data so that the similarities are stable. This issue was already studied before [2], and one of the key results is that with more data we get better stability. The size of the collected data from students is shown in Figure 5.1.

However, when computing item similarities based on items’ text, there is no problem with the size of the data because similarities are computed only using the text of items. But there is a problem when using word2vec because we create a model that probably does not comprise all words from our dataset. Although we used a large corpus from Wikipedia for creating the word2vec model, many words are in fill-in exercises of Umíme česky but are not in our word2vec model. For example in the concept Vyjmenovaná slova B, 29 words (out of 273) are missing in the word2vec model (e.g., “ledabyly”, “odrobinka”, “bicepsy”, etc.). But note that the number of missing words can be decreased by editing the words or using an even larger corpus.

Moreover, there are knowledge components, where the items are not similar text-wise but are similar grammatically. In these knowledge components, measures based on the students’ data could easily outperform measures based on the items’ text. Adding to that, in some
knowledge components we are not able to compute similarity based on items’ text. E.g., the concept Předložky s/z practices grammatical rule of placing the right preposition before a word – preposition “s” or preposition “z”. Thus with our approach to measuring the similarity between fill-in words, we are not able to measure the similarity of questions from this concept based on their text.

Choice of concepts and manual labeling

When choosing concepts for comparison of similarity measures and analysis, we only select from the pool of concepts in which we can use both our approaches for computing item similarities. Another important aspect is the previously mentioned size of the collected data from learners.

From Figure 5.1 we see that concept Vyjmenovaná slova B has the most answers. Thus it is the first concept chosen for further work. Another chosen concept is Koncovky přídavných jmen because the similarity principle is different as in Vyjmenovaná slova B, and it also has a relatively good amount of data from learners available.

From our point of view, the similarity principle for Vyjmenovaná slova B is in the base word or similar word structure. For Koncovky přídavných jmen, word similarity is based on the grammatical phenomenon in Czech (“mladý”, “jarní”) and inflection. Examining two different themes of real-world data can help us to have a better idea of how our similarity measures perform.

After the choice of concepts, we label some items from these concepts into groups manually so that we can evaluate similarity measures by clustering these labeled items and compare predicted labels from clustering to targeted labels. Additionally, it is easier to determine the quality of a visualization when we color our labeled groups and check if the words in the same group are close to each other. The manual labeling for these concepts is in Appendix B.
Figure 5.1: Counts of answers for top 30 knowledge components. Knowledge component I/Y Vyjmenovaná slova is excluded from this bar plot because its granularity is too high – it contains too many concepts, which means we do not consider using this knowledge component for further work. If a student answers multiple times to one item, we only take the first answer into account.
5.2 Comparison of similarity measures

Regarding the comparison of similarity measures, quality of a visualization is hard to quantify. On the other hand, quality of clusters can be assessed without difficulty. Therefore, clustering technique that determines clusters is utilized. Afterward, we can measure how the predicted values agree with the targeted values from manual labeling using adjusted Rand index (ARI) \([25, 26]\) (a value of ARI is close to 0 for random labeling – ARI is corrected for chance). Furthermore, we evaluate correlations among similarity measures.

As already mentioned in Chapter 3, Pearson correlation coefficient is the default choice from the perspective of computing similarities between items based on learners’ data. Additionally, we can use Pearson again or Euclidean distance to get better results (second level similarity). Thus, the measures used for comparison are Pearson, Pearson with the second step, Edit distance, and word2vec.

It is also worth mentioning that Euclidean distance might be used implicitly in some other algorithms, e.g., PCA (forces usage of Euclidean distance), t-SNE and hierarchical clustering (all of them are used in this thesis). Hence, we should be aware of this before we decide to use second level similarity. However, the second step can be employed repeatedly, e.g., we can use Pearson, Pearson again in the second step, and the result is used as input into t-SNE, which would use another step of Euclidean distance. We can also feed the result based on our measures to the algorithms such as t-SNE stating that we have precomputed distances – another step of Euclidean distance is not applied in this manner.

Keep in mind that for the similarity measures that work with the text of the items – Edit distance and word2vec, we use only the word with missing fill-in letter. Note also that for comparison and correlation of similarity measures we used only the words contained in the word2vec model.

**Correlation of similarity measures**

First, we study how the similarity measures are related. That is done by computing a correlation coefficient, which is used to measure statistical relationship between two variables. In our case, we analyze
These relationships for each pair of similarity measures. We flatten the similarity matrices into vectors and compute the correlation coefficient between each of two vectors. Figure 5.2 shows the results.

For correlations we also used the two selected datasets (Vyjmenovaná slova B and Koncovky přídavných jmen) because pairwise similarity matrices would be too huge to compute for the whole dataset of fill-in exercises from Umíme česky (similarity matrix of 4773 items – 4773x4773). Correlations are built on the following similarity measures: Pearson, Pearson→Pearson, Edit distance, word2vec and Edit distances with a threshold.

Information extracted from computed correlation coefficients is summarized below:

- The correlations between Pearson and Pearson→Pearson are high (around 0.9) in our datasets. Using second level similarity in other datasets (e.g., with a smaller amount of data) can lead to larger differences [2].

- Large differences are between similarity measures based on learners’ data and similarity measures based on items’ text (usually < 0.4).

- The correlations of Edit distance with threshold 4 and Pearson are higher than with other measures of Edit distance and word2vec. Therefore, setting a threshold in Edit distance might lead to more similar results to Pearson.

- Reasonable result is also that the correlations of techniques that work with the text of the items and Pearson are higher than with Pearson→Pearson, because Pearson→Pearson uses information on all items, not only the pairwise information.

- Low correlations of word2vec with all other measures (usually < 0.5) indicate that it works differently and that the approach of word2vec and Edit distance is distinct.
Figure 5.2: Correlations between measures in concepts *Vyjmenovaná slova B* and *Koncovky přídavných jmen*. “ED with M” is abbreviation for “Edit distance with threshold” and “dpearson” represents “Pearson→Pearson” (second level similarity).
Clustering of labeled items

As already stated in this chapter, we partly labeled two datasets from *Umíme česky* (*Vyjmenovaná slova B* and *Koncovky přídavných jmen*). That is the basis for the following evaluation of similarity measures according to the performance of clustering of these labeled items.

We cluster the labeled items from the dataset and compare predicted labels from clustering to manually provided labels using adjusted Rand index. Although there are many clustering techniques [27], we chose hierarchical clustering for the evaluation. We used the agglomerative approach where each point is initially a cluster, and the clusters are successively merged. One of the reasons why to use hierarchical clustering is the fact that we get the same and stable clustering results. Moreover, the approach and idea of hierarchical clustering are intuitive for our task. To visualize the composition of clusters, we plot the results from hierarchical clustering as a dendrogram – which is a tree diagram often used for this task. Example of such a dendrogram is in Figure 5.3.

Table 5.1 shows the results on two partly labeled concepts:

- Default Edit distance performs best in the clustering of these labeled data.

- A threshold for Edit distance does not improve the clustering results according to our manual labels.

- Word2vec performs quite well in regard to other measures.

- Pearson→Pearson and Pearson→Euclid have almost the same results.

- Any similarity measure used with Euclidean distance as the second step always brings better results than without the second step. Therefore, in later visualizations we always did the second step with Euclidean distance.

The hierarchical clustering is computed based on the specific similarity measure. When we do not use Euclidean distance in the algorithm of hierarchical clustering, we transform the similarity matrix (constructed by specific similarity measure) to distance matrix, and
then we compute the clusters. Note that the manual labeling, specific dataset, and clustering algorithm influence the results.

Based on these clustering results we decided to explore how the similarities of labeled data would look like in visualizations. We project the data onto a plane using PCA on the similarity matrices of labeled items from dataset Koncovky přídavných jmen. Figure 5.4 is based on item similarities computed by Edit distance. Item similarities for Figure 5.5 were computed by Pearson. The results of the visualizations somewhat agree with the clustering results: visualization based on the similarity matrix computed by Pearson looks worse than visualization based on the similarity matrix computed by Edit distance. In the visualization based on similarities computed by Edit distance there are some groups together; in the visualization based on similarities computed by Pearson it looks like a random placing of items.
Table 5.1: Comparison of similarity measures by hierarchical clustering of labeled items using adjusted Rand index. The top results based on specific measures are highlighted. “ED with M” is an abbreviation for “Edit distance with a threshold”.

<table>
<thead>
<tr>
<th></th>
<th>Koncovky příd. jmen</th>
<th>Slova B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pearson</td>
<td>0.05</td>
<td>0.29</td>
</tr>
<tr>
<td>Pearson → Pearson</td>
<td>0.13</td>
<td>0.35</td>
</tr>
<tr>
<td>Pearson → Euclid</td>
<td>0.12</td>
<td>0.34</td>
</tr>
<tr>
<td>Word2vec</td>
<td>0.34</td>
<td>0.25</td>
</tr>
<tr>
<td>Word2vec → Euclid</td>
<td>0.5</td>
<td>0.38</td>
</tr>
<tr>
<td>Edit distance</td>
<td>0.48</td>
<td>0.67</td>
</tr>
<tr>
<td>Edit distance → Euclid</td>
<td>0.49</td>
<td>0.73</td>
</tr>
<tr>
<td>ED with M=4</td>
<td>-0.03</td>
<td>0.19</td>
</tr>
<tr>
<td>ED with M=5</td>
<td>0.01</td>
<td>0.23</td>
</tr>
<tr>
<td>ED with M=6</td>
<td>0.09</td>
<td>0.22</td>
</tr>
<tr>
<td>ED with M=7</td>
<td>0.3</td>
<td>0.38</td>
</tr>
<tr>
<td>ED with M=8</td>
<td>0.31</td>
<td>0.38</td>
</tr>
<tr>
<td>ED with M=4 → Euclid</td>
<td>0.14</td>
<td>0.44</td>
</tr>
<tr>
<td>ED with M=5 → Euclid</td>
<td>0.2</td>
<td>0.52</td>
</tr>
<tr>
<td>ED with M=6 → Euclid</td>
<td>0.32</td>
<td>0.63</td>
</tr>
<tr>
<td>ED with M=7 → Euclid</td>
<td>0.26</td>
<td>0.64</td>
</tr>
<tr>
<td>ED with M=8 → Euclid</td>
<td>0.36</td>
<td>0.48</td>
</tr>
</tbody>
</table>
Figure 5.3: Hierarchical clustering applied on the similarity matrix computed by Edit distance. Labeled items of concept Vyjmenovaná slova B are used, and Euclidean distance is applied implicitly in the hierarchical clustering on the similarity matrix. Y-axis measures the closeness of clusters. The links below cluster nodes are colored according to predicted clusters. We defined the number of clusters that we want to find as 8 because we want to compare it to 8 manually labeled groups. The coloring of groups is different from the coloring in projections and in Appendix B.
Figure 5.4: Projection of labeled items onto a plane using PCA on the similarity matrix computed by Edit distance. Some items were deleted or slightly moved so that all the items are properly visible. PCA forces usage of Euclidean distance on the similarity matrix.
Figure 5.5: Projection of labeled items onto a plane using PCA on the similarity matrix computed by Pearson. Some items were deleted or slightly moved so that all the items are properly visible. PCA forces usage of Euclidean distance on the similarity matrix.
5.3 Visualizations

In this section, we describe the quality of visualizations based on PCA compared to the visualizations based on t-SNE, and study suitable parameter settings of t-SNE for our dataset. Eventually, we show some examples of word projections by t-SNE and various measures. We discuss the quality of visualizations by looking at them.

Some of the visualizations are included only in the electronic attachment because they have similar aspects to the visualizations illustrated here. Furthermore, the projections of words are not only visualized by matplotlib\(^1\) – Python plotting library, but also by Plotly\(^2\) – an interactive, browser-based graphing library. In visualizations by Plotly we can easily manipulate the axes or filter items by groups, which gives us the ability to interact with the visualization.

All visualizations are based on concepts Vyjmenovaná slova B and Koncovky přídavných jmen from Umíme česky. Some items from these concepts are manually labeled and divided into groups, which are listed in Appendix B. The color for each group is the same in all visualizations, and the respective coloring of groups is also included in the appendix.

**t-SNE vs. PCA**

First, we compare t-SNE to PCA using our dataset. Example of such comparison is in Figure 5.6. We can see what the advantage of t-SNE is: clusters are better visually separated.

Therefore, if we tune the hyperparameters of t-SNE right, we can get better results than from other data visualization techniques. From PCA we often get a result that looks like a cloud of points, where we can not easily see the separated clusters.

From this point, we only explore the performance of t-SNE with various parameter settings and similarity measures.

---

1. https://matplotlib.org/
5. Evaluation

Parameter settings of t-SNE

For a higher quality visualization of t-SNE we have to analyze multiple plots based on various perplexity values. Analysis of the plots is an essential part of exploring item similarities in our data with t-SNE. This step is specific for each dataset, each domain and also each similarity measure used for computing item similarities. Projection based on one similarity measure can produce more clusters than projection with the same parameters based on another similarity measure.

Other t-SNE parameters such as the learning rate or the number of iterations are also important but not as important as the perplexity value.

From our experience, the perplexity between values 5 to 15 gives us good results (visually separated clusters of similar items) for both of our concepts Vyjmenovaná slova B and Koncovky přídavných jmen. In Figure 5.7, we show an example of how the perplexity value influences clustering of the items.

As we can see, with perplexity set to 2, t-SNE mostly deals with local similarities in the data. The subplots with the perplexity 5, 10, and 20 isolate the labeled clusters quite well, whereas the subplot with
the perplexity set to 50 stacks more items together and we slightly lose information about the clusters. Furthermore, the projection created by t-SNE with the perplexity set to 100 is strange because the perplexity value should be smaller than the number of items – the similarity matrix contains 93 items.

Figure 5.7: Influence of the perplexity value on the clusters of items in the projection using t-SNE. Dataset of Koncovky přídavných jmen is used and items with manually provided labels are colored according to their groups (black color is for unlabeled data). The input to t-SNE is the similarity matrix that is computed based on word vectors (word2vec). Additionally, t-SNE uses Euclidean distance on the similarity matrix.

Our approach to setting the number of iterations was simple – after setting perplexity and learning rate, we run t-SNE with different values of iterations. If the plots with higher values of iterations are relatively similar to the plots with lower values of iterations, we have probably reached a stable configuration. Learning rate is tricky to set; we need to try many values to set it. Based on experience, if learning rate’s value is too high or too low, it can give strange results. We set
100 for learning rate by default and optimize the value if the result looks odd.

However, the perplexity always makes the biggest difference, and thus it is the most crucial parameter for setting if we want to improve the results.

**Visualizations by t-SNE**

At last, we present some visualizations by t-SNE, which from our point of view have better quality: Figure 5.8 (based on similarity matrix built on Edit distance), Figure 5.9 (based on similarity matrix built on word2vec) and Figure 5.10 (based on similarity matrix built on Pearson). The labeled items from the same group are often close to each other, and the groups of similar items are often separated.

We can also see some patterns and clusters created from the unlabeled items in the visualization that is built on *Vyjmenovaná slova B* dataset (Figure 5.10).

The two visualizations that work with the concept *Koncovky přídných jmen* are constructed by using measures that computed the similarity of items based on the items’ text (Figure 5.8, Figure 5.9). We can see how well the measures perform and separate the clusters from each other.

It is important to note that with the second level similarity used on the similarity matrix computed by the measures, the visualizations look better – similar items are grouped and the clusters are further apart from each other.

The choice of the parameters for these visualizations was influenced by exploring the quality of visualizations. We created many visualizations with various parameter settings, and then we chose the best-looking ones of these visualizations.

The three projections of items were edited so that all the words are visible: some of the items were deleted or slightly moved. The particular dataset, similarity measure, and perplexity value are in the title of the figures.
Figure 5.8: Projection of items using t-SNE on the similarity matrix computed by Edit distance. Euclidean distance is used on the similarity matrix in the t-SNE implicitly (second level similarity).
Figure 5.9: Projection of items using t-SNE on the similarity matrix computed by word2vec. Euclidean distance is used on the similarity matrix in the t-SNE implicitly (second level similarity).
Figure 5.10: Projection of items using \( t\text{-SNE} \) on the similarity matrix computed by Pearson. Euclidean distance is used on the similarity matrix in the \( t\text{-SNE} \) implicitly (second level similarity).
6 Conclusion

In this thesis, we focused on the application of dimension reduction techniques for data projection with the goal of having similar items close to each other in the visualization.

We used item similarities as input to dimension reduction techniques. Hence, similarity measures were the first topic, which had to be studied for this task. We continued in the previous work of Radek Pelánek and Jiří Řihák [2], in which they examined computation of item similarities based on learners’ performance data. We proposed two new similarity measures for computing the item similarities based on the items’ text.

We compared two different approaches of dimension reduction techniques (PCA, t-SNE), which are used for projection of data onto a plane. Additionally, we described the usage of t-SNE and setting of its parameters.

We partly labeled our dataset, and then we clustered and visualized the items using their computed pairwise similarities. We compared the performance of similarity measures according to the manual labeling. Based on the results of clusterings and visualizations, we can say that the similarity measures based on the items’ text are also useful and can give us outcomes that are different than the outcomes of measures that use the learners’ performance data. Based on the visualizations, we can find duplicated items, similar grammatical phenomenon tested in many questions, or outliers, which can lead us to the idea that the item should be moved to another concept or items similar to that item should be added.

The limitations of the evaluation are in the manual labeling part because it could very much influence our results.

The future work can combine the similarity measures that work with learners’ data with the measures that work with the text. Moreover, the quality of a visualization is difficult to measure, and therefore evaluation of the projections can be studied further. The effectiveness of different visualization techniques in visual data mining [28] can be inspected in-depth and most likely would need a user study.
Bibliography


A Contents of the electronic attachment

Here we describe the contents of the electronic attachment. There are the following files and directories in the archive:

- *.ipynb (IPython notebook files) – contains analyses, where the functions from other scripts are used,
- *.html (IPython notebooks converted to HTML),
- requirements.txt – contains a list of python libraries that were used in this thesis,
- data (directory) – contains scripts related to data processing,
- projections (directory) – contains scripts related to projection,
- utils (directory) – contains scripts related to creating the word2vec model,
- similarities (directory) – contains scripts related to similarity measures,
- manual_labeling (directory) – contains manual labeling of 2 concepts (also in Appendix B),
- visualizations (directory).
B Manually labeled data

Vyjmenovaná slova B and Koncovky přídavných jmen are the knowledge components used for evaluating the similarity measures and visualizations. We have manually determined groups of items corresponding to their similarity based on Czech grammar rules and their base words.

For the measures that compute the similarity based on the text of the items, only the word in italics is used. The color of groups in all the visualizations in the thesis is in the parentheses. Unlabeled items are black.

Vyjmenovaná slova B

In this concept, the similarity of words has been determined mostly by their base words.

1. abych, abychom, abys, kdybyste, kdyby, aby (blue)
2. bylina, bylinkář, bylinkový, bylinka, ruské pověsti se nazývají byliny (green)
3. biologie, biograf, biotop, biosféra, biofyzika, biografie, biorytmus, biocemik (red)
4. zbytečný, zbytek koláče, zbytek, zbylý, zbylé látky odložila do krabice, zbyl mi kousek dortu, zbyly po něm dluhy, měla zbytečné starosti (cyan)
5. kobyla, kobylka (magenta)
6. obyvatelka, obyvatelstvo, výzva obyvatelstva k pořádku (orange)
7. biblický příběh, bible (gray)
8. dobytkářství, zabýval se dobytkářstvím, obchodoval s dobytkem, dobytče, dobytek (purple)
Koncovky přídavných jmen (jarní/mladý)

In this concept, the similarity of words has been determined mostly by Czech grammar rules.


2. s *kozím* sýrem, skrábnutí *orlím* pařátem, s *pštrosím* perem, Příchodí byli ohlášeni veselým *psím* štěkotem. (green)

3. z *žabích* jíker, o *pavích* perech (red)


