Universität Zürich[UZH]

Seminar
Language Technology for Special Education
Fall semester 2021

Computer-assisted training software for psychomotor diagnostics

Author: Micaela Alexandra Ribeiro Vieira
ID number: 13-760-285

Lecturer: Dr. Sarah Rahel Ebling
Department of Computational Linguistics

Submission date: 23.01.2022

Abstract

Context: As in any other area of medicine, diagnoses play a fundamental role also in psychomotricity. In fact, only with a correct diagnosis it is possible to provide an adequate therapy to patients. Diagnostic is a skill that is refined with practice. Between 2014 and 2016, the Institute of Computational Linguistics at the University of Zurich and the Interkantonale Hochschule für Heilpädagogik carried out a project aimed at creating a software that would give psychomotor students the possibility to practice their diagnostic skills autonomously. This software, in German language, did not reach a development stage suitable for practical use.
Aims: Using the Python programming language, we resume the initial project by integrating new features and implementing state-of-the-art techniques.
Results: We created a code that uses semantic similarity to quantify how close the remarks of a student's diagnosis are to those of experts' diagnoses. Semantic similarity is built on top of sentence embeddings. We implemented three algorithms to embed remarks: doc2vec, SentenceBERT and InferSent. It is found that SentenceBERT performs best. We also made some preprocessing steps on the remarks. There are still caveats which are mainly due to the language of the project. In particular, a German model for contextual spelling correction is missing. Nevertheless, this work can surely be the starting point for the creation of a highly-performant real-world interface.

1 Introduction

Having a good diagnosis is crucial as it largely determines how effective a therapy will be. Expert health professionals who performed multiple deliberate practice sessions have a better and broader understanding about the relevant aspects of the medical issues they had to face than novices [Caspar et al., 2004]. Deliberate practice refers to the process of trying to solve a problem over and over and get feedback until the task is properly mastered [Ericsson et al., 1993]. Ideally, this learning methodology is based on systematic, fast feedback.

Traditionally, students in the medical field deepen their diagnostic skills by working side-by-side with experts. The combination of an expanding number of students and a relatively static population of patients and instructors leads to increasing difficulties in teaching clinical diagnoses this way. Consequently, students have fewer opportunity to practice their diagnostic skills, and the degree to which they are supervised and controlled whilst practicing these skills gets less and less [de Dombal et al., 1969]. One possibility to overcome this issue is to have a group of students that first remotely observe and judge a clinical situation and then discuss the findings with experts. Even if this methodology is a clear improvement of the individual side-by-side teaching, this approach suffers from limitations as well because there is still need to discuss one's observations with instructors, and this requires adequate time and supervision.

In 2014, the Institute of Computational Linguistics at the University of Zurich and the Interkantonale Hochschule für Heilpädagogik came up with the idea of creating a computer linguistics software to give psychomotor students a new tool to further practice diagnostic skills[1]. Since the final interface was intended for students of the Interkantonale Hochschule für Heilpädagogik, researchers chose to set the language of the software to German. Ideally, the software should have required no additional human resources. Therefore, this would have been a perfect training tool to allow psychomotor students to improve their diagnostic skills without any external help.

The functioning principle of the final software would have been the following (see Figure 1). The software provides a video of a psychomotor diagnosis evaluation to a student (1) who then writes their diagnosis (2). Then, an algorithm compares the student's diagnosis to the experts' diagnoses previously stored in the software (3). Based on the semantic similarity between the student's remarks and the

[1] See `www.hfh.ch/en/project/computer-aided-exercises-in-psychomotor-diagnosis` (retrieved on November 26, 2021).

experts' ones, the software gives feedback (4). The student has the possibility to restart the video and reassess the case. Deliberate practice will lead the student to enhance their diagnosis (5).

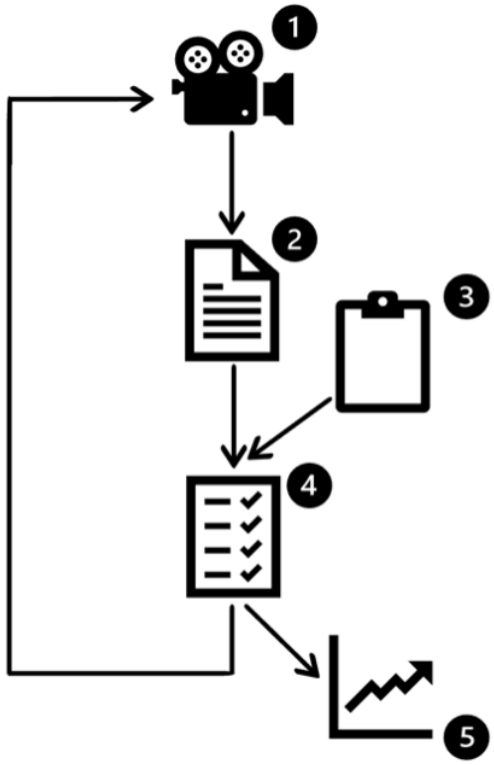

Figure 1: software functioning principle. The elements are: 1) video of a psychomotor diagnosis, 2) student's evaluation, 3) experts' evaluations, 4) feedback, 5) performance enhancement.

Researchers carried out this project between 2014 and 2016 and then shelved it. They developed a demo interface[2] based on a doc2vec algorithm to embed remarks (see Section 3.4.1) and calculated the semantic similarity between experts' and students' remarks with the cosine similarity. The sentence embedding algorithm was based on Gensim[3].

The goal of this study is to revive the original project by adding new functionalities and bringing it to the current state of the art. In particular, we develop functions to preprocess sentences and we implement new algorithms to embed the remarks. The creation of a suitable interface is instead left for a future work. This interface could be built on top of the demo interface that was ideated in the original project.

2 Psychomotor diagnostics

The term psychomotricity stems from the combination of "psycho" and "motricity". Psycho refers to the cognitive abilities of a person and motricity to their movement abilities. Therefore, we can consider psychomotricity as a function of the human being to synthesize their psychological and motoric components [Berdilă et al., 2019]. This function promotes the integral development of a person at a motoric, emotional, cognitive, and social level through body movement.

It is considered that psychomotor activity starts from the child's first years of life. Children manipulate objects and play because they need to learn the environment. Psychomotor development is unique

[2]See `https://pub.cl.uzh.ch/demo/hfh/`, retrieved on December 20, 2021.
[3]See `https://pypi.org/project/gensim/`, retrieved on December 20, 2021.

for each child and is the result of the accumulated experiences, of the reactions, and the answers given when different tasks are assigned to them. Each child has their own pattern of growth, development, and motor learning. Consequently, children build a subjective temperament and behavior [Barbosa, 2012].

Children experiencing psychomotor disorders benefit from attending dedicated therapeutic sessions. In these sessions, they play in a controlled environment and interact with the psychomotor teacher. Thanks to the situations experienced during these sessions, children learn how to express their emotions and how to control and manage time, space, and their movements.

Since each child is unique and has gone through an individual developmental path, any psychomotor therapy must necessarily be tailored to the kid's needs. Prior to therapy, the psychomotor therapist makes a diagnosis. In this field, the diagnosis is made using a standardized evaluation called Movement Assessment Battery for Children [M-ABC, Petermann, 2015]. This procedure is composed of gross and fine motor exercises, as well as graphomotor tasks. The psychomotor therapist evaluates the child not only during the tasks, but also between them. In fact, the therapist obtains a lot of information during breaks, such as how the child approaches the exercises or how they handle frustration.

Psychomotor therapists split their psychomotor diagnoses into two categories called anamnesis and game situation. The former contains medical remarks, whereas the latter points out remarks on how the child behaves during the tasks. Therapists further divide these two categories into three subcategories: observations, problems, and resources. Health professionals label the remarks that make up the diagnoses according to the aspect to which they relate. This labeling of remarks consists of two levels: the category – e.g., attention, restlessness, and mother's behavior – and the subcategory – e.g., school, rules, and duration of play. These labels are specific to the category-subcategory subdivision of the diagnoses and allow to build a fine-grained picture of the latter.

3 Project implementation

To test the software, we are provided with a single psychomotor diagnosis evaluation. For this evaluation, there are both expert's diagnoses (six files, one for each diagnosis category-subcategory combination, see Section 3.1) and some students' diagnoses (two files per student, one per category, see Section 3.2). This project consists of five Python codes. In addition to the main code (`psychomotor_diagnosis_main.py`), there are a code containing functions to preprocess remarks and to extract them from files depending on their categories and subcategories (`preprocessing.py`, see Section 3.3), and three codes containing the algorithms that can be used to calculate the semantic similarity between the students' remarks and those of the experts (`doc2vec.py`, `sentencebert.py`, and `infersent.py`, see Section 3.4).

Moreover, the SentenceBERT and InferSent algorithms need pretrained models stored in the same folder of the codes. These pretrained models can be downloaded with the `download_models` executable file.

All files are available on GitHub[4]. We designed the software in a modular way, so that it can be easily adaptable to real-world use.

3.1 Experts' remarks

In the original project, researchers divided the experts' remarks into six files, one for each diagnosis category-subcategory combination. The files are tab separated values (TSV) which have a name consisting of the category and the subcategory in German: `anamnese_beobachtungen`, `anamnese_her-`

[4]See `https://github.com/micaela-vieira/psychomotor_diagnosis`, retrieved on January 10, 2022.

`ausforderungen`, `anamnese_ressourcen`, `spielsituation_beobachtungen`, `spielsituation_herausforderungen`, and `spielsituation_ressourcen`.

These six files contain the remarks of three experts. In each file there are five columns which have:

1. the experts' acronyms which, for reasons of anonymity, replace the first and last names of the experts. The acronyms are AW, DJ, and JS;
2. the experts' remarks;
3. the IDs associated with the categories of the experts' remarks;
4. the categories of the experts' remarks;
5. the subcategories of the experts' remarks.

To show the format of an expert's file, in Figure 2 we display an excerpt from `anamnese_herausforderungen`. The researchers that worked on the original version of the project manually added the last three columns of experts' files.

In this study, we are interested only in the second and the fourth column. When the user chooses the desired diagnosis category and subcategory (see Section 4), the software selects and stores the experts' remarks from the corresponding file into a list.

AW	Kann ein Zeichen hoher innerer Anspannung sein. Kann darauf hindeuten, dass das, was sie erzählt, sie emotional stark berührt bzw. bewegt.	7	Verhalten der Mutter	Emotionen
AW	Die kurze Zeitspanne des Augenkontakts kann auf Unsicherheit oder emotionale Bewegtheit hindeuten.	7	Verhalten der Mutter	Emotionen
AW	Die Mutter wirkt aber auch besorgt und es scheint, als sei sie irgendwie unter Druck	7	Verhalten der Mutter	Emotionen
DJ	C. kann sich in Klassensituation weniger konzentrieren.	2	Aufmerksamkeit	Situativ
DJ	Die Lehrerin hat die Hypothese, dass er unterfordert ist	3	Motivation	Externe Anreize
DJ	Vielleicht gab es Konflikte im Zusammenhang mit der Autonomieentwicklung.	4	Entwicklung	Autonomie
DJ	Sein Entwicklungsstand und die schulischen Anforderungen differieren stark (Unterforderung).	4	Entwicklung	Diskrepanzen
DJ	Der Bereich des Sozio-Emotionalen steht als Indikation im Zentrum.	5	Indikation	Sozio-Emotional
JS	P: Aufschub der Bedürfnisbefriedigung scheint für C noch nicht lohnenswert zu sein.	1	Verhalten	Selbstregulation
JS	P: Das von der Lehrperson beobachtete unruhige Verhalten und das Stören der anderen Kinder weisen auf ein unkontrolliertes, impulsives Verhalten hin.	1	Verhalten	Selbstregulation
JS	P: Könnte ein Hinweis auf Problematik in der Selbststeuerung, Handlungskontrolle sowie in der Impulskontrolle sein.	1	Verhalten	Selbstregulation

Figure 2: Excerpt from `anamnese_herausforderungen`.

3.2 Student's remarks

In the original project, researchers split students' diagnoses into two tab separated values files (TSV), one for the anamnesis and one for the game situation. In both files there are four columns, one for each subcategory of the diagnosis and one for additional observations (in most cases, this last column is empty). To display the format of a student's file, in Figure 3 we show the full content of `Anamnese_1518_Ulmann.N`.

In this study we are interested only in the remarks belonging to the diagnosis subcategories, i.e., the first three columns of the files. When the user specifies a file to evaluate and selects the desired subcategory (see Section 4), the software stores the student's remarks from the sought column into a list.

The software accepts the student's diagnoses not only as TSV files, but also as a list of sentences written in the command line. This second option comes closest to the final interface where students will write their remarks in a dedicated field next to the video of the psychomotor diagnosis evaluation.

3.3 Preprocessing

The implemented software preprocesses the selected experts' and student's remarks. The preprocessing stage serves to build utterances that can be handled properly by the semantic similarity algorithms (see

Die Mutter sagt, dass er viele Fragen hat.		Er ist sehr interessiert und wissbegierig.	
Das Kind muss lernen, die Fragen zurückzuhalten.	Er ist ungeduldig.		
Zuhause wird darauf geachtet, dass er wartet.		Er wird zuhause auch gefördert.	
Er war schon als kleines Kind aktiv.		Er hat viel Energie.	
Er schlägt mit den Händen Rhythmen auf den Tisch.		Er ist musikalisch.	
Mutter sagt, dass es klappt, wenn er belohnt wird.	Er braucht eine Motivation und ein Ziel, um sich anzustrengen.		
Wenn er etwas haben will, gibt er sich Mühe.		Er ist zielstrebig.	
Er bastelt und klebt mit allem, was er findet.		Er ist kreativ und kann sich selbst beschäftigen.	
Beim Legospiel kann er der Anleitung folgen oder selbst etwas konstruieren.		Er kann Anweisungen befolgen, Pläne lesen und ist kreativ.	
Beim Legospiel kann er lange Zeit dranbleiben.		Er kann sich gut konzentrieren, wenn ihm eine Aufgabe gefällt.	
Er kann Velofahren und Skifahren.		Er ist motorisch gut.	
Er ist sehr motiviert und immer bereit etwas zu tun.		Er hat viel Motivation.	
Er möchte die Aufgaben so schnell wie möglich machen und dann wieder spielen gehen.		Er weiss, was er will.	
		Er spielt gerne.	

Figure 3: Example of student's file (`Anamnese_1518_Ulmann.N`).

Section 3.4). In particular, the software removes unnecessary tokens, replaces some abbreviations with the full forms, and corrects typos. The preprocessing stage consists of several steps:

- First, the software removes tokens written at the beginning of remarks that are not functional for their understanding. In fact, JS writes either *P.* or *P:* at the beginning of the remarks belonging to the problems subcategory (see Figure 2) and either *R.* or *R:* at the beginning of the remarks belonging to the resources subcategory;

- Then, the software replaces the widely used abbreviation *Th.* with the full form *Therapeut* (therapist). The choice of always using the masculine noun instead of the feminine one (*Therapeut* rather than *Therapeutin*) introduces a bias in the remarks. Moreover, this substitution gives rise to possible semantic misunderstandings when there are feminine possessive adjectives referring to the health professional (e.g., *ihr*, her). This is unavoidable unless we implement an algorithm to replace all gender-based words;

- Subsequently, the software substitutes the initial of the patient with a fictitious name. For reasons of anonymity, who wrote the remarks replaced the patient name with the initial. Here, we perform the reverse process;

- Afterwards, the software keeps intact common German abbreviations such as *d.h.* (*das heisst*, that is);

- Finally, the software inspects for their spelling all tokens containing only alphabetic characters that do not undergo the aforementioned preprocessing steps. First, an algorithm checks if the tokens are either part of the German Enchant dictionary[5], or correct German compound words[6]. If not, we assume that the tokens contain typos. The initial goal was to correct these tokens using contextualSpellCheck[7], a contextual spelling algorithm. However, contextualSpellCheck is not implemented for German but only for English. Since we have not found a contextual spelling algorithm suited for German (and being its creation outside the scope of this study), we opt for a more elementary algorithm based on the Levenshtein distance [Levenshtein, 1966]. Given two strings X and Y, the Levenshtein distance is the minimum number of elementary changes – i.e., deletion of a character, substitution of a character by another, and insertion of a new character – necessary to transform string X into string Y. The algorithm searches which German words have the shortest Levenshtein distance from a token containing typos and replaces the original

[5]See `https://pyenchant.github.io/pyenchant/index.html`, retrieved on December 20, 2021.
[6]See `https://pypi.org/project/compound-word-splitter/`, retrieved on December 20, 2021.
[7]See `https://pypi.org/project/contextualSpellCheck/`, retrieved on December 20, 2021.

token with one of them. The implemented algorithm is easy to use, but is prone to errors, because the selected word is not necessarily the one intended by the person who wrote the sentence. In fact, multiple words may have the same Levenshtein distance from the target token. Take for example the remark *Die Th. fagt ob er wütend sei* (The therapist *fagt* if he is angry) taken from `Spielsit_1417_Simon.V`. The token *fagt* is replaced with *fast* (almost). However, it is clear from the context that the appropriate correction should have been *fragt* (asks). We could remove this source of errors only with the adoption of a contextual spelling algorithm.

3.4 Semantic similarity algorithms

The software compares preprocessed student's remarks with preprocessed experts' remarks belonging to the same diagnosis category and subcategory. For each student's remark, the software searches the most semantically similar expert's remark. The software returns the most semantically similar expert's remark together with its category and the similarity score. If the similarity score is higher than a preset threshold (we choose a value of 0.9), the software displays the expert's remark most similar to that of the student. Otherwise, the software shows only the score. The purpose of the threshold is to facilitate deliberate practice. In fact, if the expert's remark were provided when the similarity score is too low, the student would be inclined to copy the correct remark and not to think why their remark is wrong. At the end of the diagnosis, the software displays a final feedback with indications on the remarks' categories (see Section 3.5).

The remarks' comparison is possible thanks to sentence embedding, a method to express the meaning of a sentence with a multi-dimensional vector [see Arora et al., 2017, and reference therein]. The strength of sentence embedding is that sentences which have a similar meaning have a similar numerical representation. We can appreciate the similarity between embeddings thanks to cosine similarity. Cosine similarity is a measure of similarity between two non-zero vectors. Given two vectors $\vec{u}$ and $\vec{v}$, the cosine similarity score is

$$S_c(\vec{u}, \vec{v}) = \frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\| \, \|\vec{v}\|}. \tag{1}$$

If two sentences have congruent meaning, the similarity score tends to one. If two sentences have different meanings, the similarity score is close to zero. If two sentences have opposite meanings, the similarity score approaches minus one. In Figure 4 there are 2-dimensional examples to visualize cosine similarity applied to semantic similarity on sentences. Note that state of the art embeddings have up to hundreds dimensions.

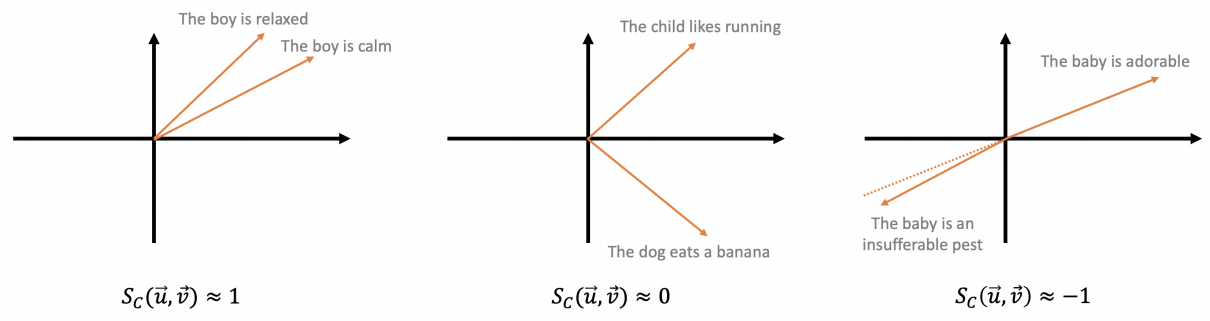

Figure 4: Cosine similarity examples.

Here, we implement three sentence embedding techniques: doc2vec, SentenceBERT and InferSent.

3.4.1 doc2vec

word2vec takes advantage of a neural network – either the Skip-gram model, or the Continuous Bag of Words model (CBOW) – to learn word embeddings [Mikolov et al., 2013]. A properly encoded text is divided into "windows" of a given size and then these windows are fed to the neural network. The neural network splits the window into a "target" (a single word) and a "context". CBOW predicts the target given its context, whereas Skip-gram works the other way around.

In this study we are interested in the CBOW neural network. By training such a model over large corpora, it is possible to predict which word is more likely to appear either before or after a set of words. As can be seen from the example in Figure 5, the CBOW architecture has four layers:

1. the input layer contains the context words of the window;
2. the first hidden layer is made of the vector representations of the context words;
3. the second hidden layer has a single vector enclosing all context words. This is achieved by averaging/concatenating the vector representations of the context words;
4. the output layer accommodates the prediction, i.e., the target word of the window.

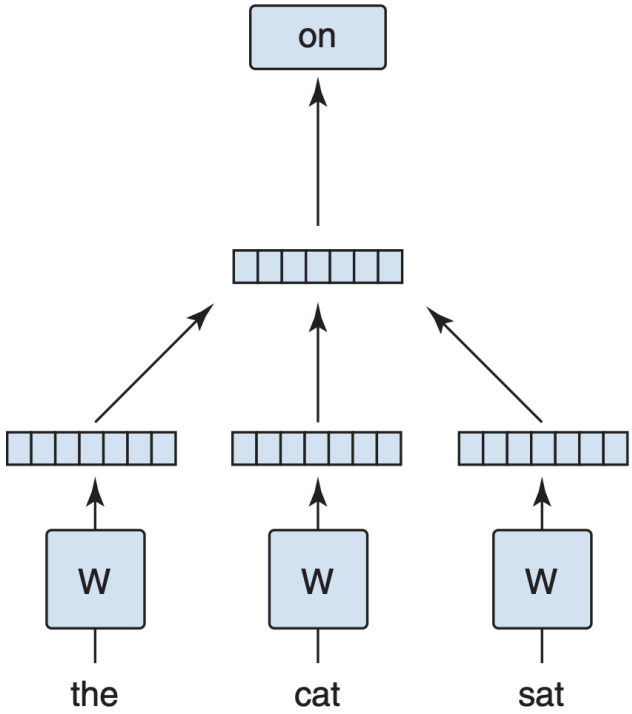

Figure 5: CBOW architecture. The context words *the*, *cat*, and *sat* are used to predict the target word *on* [image retrieved from Teofili, 2019, p. 172].

doc2vec is an extension of word2vec [Le and Mikolov, 2014]. For the CBOW neural network, the intuition is to add a unique paragraph ID to the context (see Figure 6). Therefore, the neural network relates the target word not only to other words, but also to a label. Despite the name, the doc2vec algorithm associates paragraph IDs also to single sentences and full documents. Since paragraph IDs act as a memory that keep track of what is missing from the context to complete the window, this model is called Distributed memory model of paragraph vectors (PV-DM).

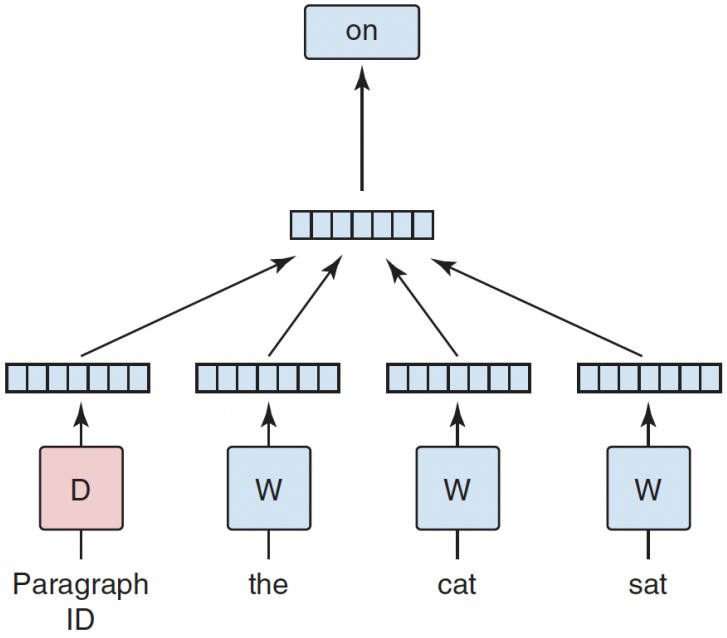

Figure 6: PV-DM architecture for the same example of Figure 5 [image retrieved from Teofili, 2019, p. 172].

doc2vec tags sentences with a unique paragraph ID and generates a vector representation containing the label. This allows us to find sentences having a similar meaning using cosine similarity.

3.4.2 SentenceBERT

BERT stands for Bidirectional Encoder Representations from Transformers and is a very valuable method for natural language processing [Devlin et al., 2019]. BERT owes its success to three aspects. First, it uses transformers that learn contextual relations between words in a text. One example of this attention mechanism is the ability to relate a first name with pronouns referring to the same person occurring over a whole paragraph. Second, BERT is pretrained on a large corpus of unlabeled text including the entire Wikipedia and the Book Corpus. This gives the model the capability to grasp a very deep understanding of how a language works. Third, the model is bidirectional: during the training phase, it learns information on a target word from context on both the left and the right side. This is a remarkable upgrade compared to the doc2vec method where only a direction is considered.

Due to the large complexity of the model, computing similarity between two sentences using BERT is feasible, but very slow. The reason for that is that BERT needs to process sentences one after the other. The speed bottleneck can be removed with SentenceBERT [SBERT, Reimers and Gurevych, 2019].

SBERT simultaneously processes two sentences in the same way (see Figure 7). The algorithm first passes these sentences to BERT and then it feeds the output into a pooling layer to create fixed-size sentence embeddings. The pooling layer serves to address input sentences of varying lengths into numerical representations of identical size. Then, the algorithm fine-tunes the output of this layer on semantic textual similarity data in order to create sentence embeddings which are semantically meaningful. In the original work, researchers trained SBERT on both the SNLI – a dataset with 570000 sentences pairs tagged with the labels “neutral”, “entailment”, or “contradiction” [Bowman et al., 2015] – and the Multi-

Genre NLI – a dataset composed of 430000 sentences pairs [Williams et al., 2018]. Finally, the algorithm compares the semantically meaningful sentence embeddings with cosine similarity.

The semantic textual similarity data needed to fine-tune SBERT is language dependent. Since the project of this study is in German, we must employ semantic textual similarity data different from the one of the original project. For this purpose, we use German SentenceBERT [Chan et al., 2020].

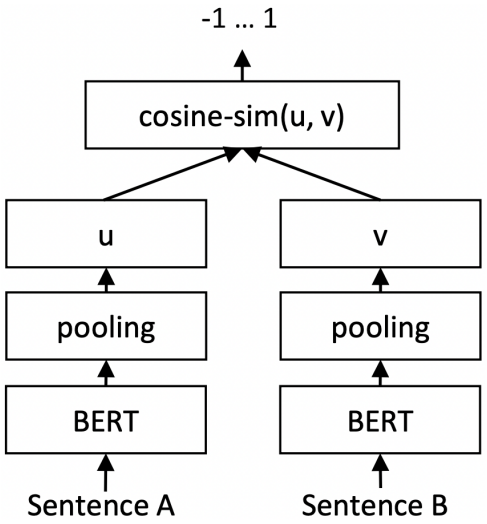

Figure 7: SBERT architecture to compute similarity between two sentences [image retrieved from Reimers and Gurevych, 2019].

3.4.3 InferSent

InferSent is a supervised sentence embedding method which have an architecture which is very similar to the one of SentenceBERT (cfr. Figure 7 and Figure 8). SentenceBERT and InferSent differ primarily in the training data used. SentenceBERT has BERT at its core and is only fine-tuned with SNLI and Multi-Genre NLI. InferSent is trained directly on SNLI [Conneau et al., 2017].

Similarly to SentenceBERT, InferSent takes two sentences and encodes them to generate the corresponding embeddings u and v. Then, the algorithm extracts relations between the two sentence embeddings thanks to concatenation (u, v), absolute element-wise difference $(|u - v|)$, and element-wise product $(u * v)$. Finally, the algorithm feeds the output vector resulting from these three operations into a classifier composed of multiple fully-connected layers and a softmax layer. The classifier classifies the sentences pair into one of the three SNLI categories ("neutral", "entailment", and "contradiction").

Conneau et al. [2017] experimented with various architectures to encode sentences into fixed-size numerical representations. It was found that a Bi-directional LSTM [BiLSTM, Graves and Schmidhuber, 2005] with max pooling performs best.

Up to now, InferSent is developed only for English. Since our project contains remarks in German, it is decided to translate utterances into English to perform semantic similarity comparisons[8]. This is clearly a limitation, in particular if typos were not properly handled (see Section 3.3).

[8] See `https://pypi.org/project/deep-translator/`, retrieved on December 20, 2021.

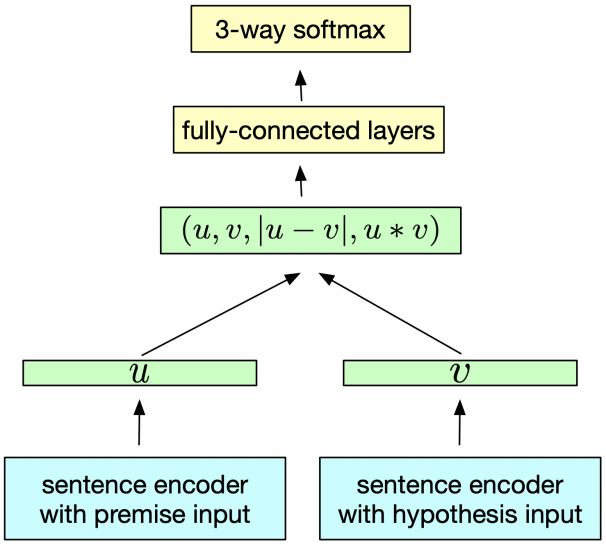

Figure 8: SNLI training architecture [image retrieved from Conneau et al., 2017].

3.5 Final feedback

At the end of the diagnosis, students receive feedback to appraise their performances (see Section 4). As described in Section 3.4, the software associates each remark written by the students with an expert's remark and from this association it stores the similarity score and the category of the expert's remark. To build the final feedback, the software first groups the similarity scores according to the categories of the expert's remarks, and then it averages them. These average scores are output together with the corresponding categories. The software outputs also categories which are not met by the student but are present in the experts' diagnoses. This way, students understand for which categories they did a good job, for which they still have to improve, and which ones they omitted.

4 Demo example

The main code (`psychomotor_diagnosis_main.py`) contains several parameters the user can choose from. Several of them are preset by default, but the user always has the possibility to manually select other options. The parameters are:

- -A, i.e., the algorithm. The user may choose among doc2vec, SentenceBERT, and InferSent. By default, we set SentenceBERT because we rated it to be the algorithm that performs best. Our evaluation is based on the comparison of the similarity scores and most similar sentences that the three algorithms generate for the remarks contained in `students/Anamnese1417Eggel.f.tsv`;
- -c, i.e., the diagnosis category. The user may choose between *anamnese* (anamnesis), and *spielsituation* (game situation). By default, we set *anamnese*;
- -s, i.e., the diagnosis subcategory. The user may choose among *beobachtungen* (observations), *herausforderungen* (problems), and *ressourcen* (resources). By default, we set *ressourcen*;

- -a, i.e., the list of abbreviations not to preprocess. By default, this list contains the German abbreviations *d.h.* (*das heisst*, that is), *s.a.* (*siehe auch*, see also), *u.a.* (*unter anderem*, among others), and *z.B.* (*zum Beispiel*, for example);
- -p, i.e., the fictitious name of the patient. By default, we choose *Andreas*;
- -f *filename*, where *filename* is the name of the file containing the student's diagnosis. If the student wants to write the diagnosis directly from the command line, they should select another parameter (i.e., -w);
- -o, i.e., the possibility to save the output of the diagnosis evaluation to a text file. In this text file, the software writes experts' remarks even if the similarity score is lower than the preset threshold of 0.9.

According to the chosen parameters, the program selects the desired remarks, preprocesses them and performs semantic similarity comparisons. Here, we propose an example with the command `python psychomotor_diagnosis_main.py -s ressourcen -f students/Anamnese_1417_-Eggel.f.tsv`. With this command we decide to:

1. use SentenceBERT. -A is not specified, therefore the default algorithm is chosen;
2. observe the category *anamnese*. -c is not specified, therefore the default category is analyzed;
3. look at the subcategory *ressourcen*;
4. employ the preset list of abbreviations and patient name. -a and -p are not specified, so the default parameters are utilized;
5. evaluate the remarks contained in `students/Anamnese_1417_Eggel.f.tsv`;
6. do not save an output text file. This is because -o is not specified.

In Figure 9 we display what the program prints on the command line. In the upper part there are the student's remarks, the similarity scores, and the most similar expert's remarks. In the lower part there is the final feedback.

Nine remarks belonging to the category-subcategory anamnesis-resources are present in the chosen file. For each student's remark, the similarity score is greater than the threshold of 0.9. This means that every most similar expert's remark is also written. Analyzing the several remarks' pairs, we can see that some of them have a very similar meaning. For example, the remarks *Die Mutter möchte dem Kind helfen* (The mother wants to help the child) and *Die Mutter sieht klar die Stärken ihres Sohnes und will ihn unterstützen* (The mother clearly sees the strengths of her son and wants to support him) are almost identical. Nevertheless, there are also pairs where the meaning is different although the similarity score is very high. An example is the pair composed of *Die Ausdauer des Jungen scheint gut zu sein* (The endurance of the boy seems to be good) and *Die Mutter sieht klar die Stärken ihres Sohnes und will ihn unterstützen* (The mother clearly sees the strengths of her son and wants to support him). This may be caused by a faulty preprocessing of the set of remarks.

From the final report, we note that the nine remarks are divided into three categories: *Begabungen* (inclinations), *Interessen/Motivation* (interest/motivation), and *Eltern* (parents). In the former category there is one remark, while the latter categories have four remarks each. The average similarity score is very high in all these three categories (0.920, 0.952, and 0.952, respectively). Nevertheless, the student's diagnosis is not exhaustive, as there are four categories which are not met. These categories are *Aufmerksamkeit* (attention), *Schule* (school), *Externe Anreize* (external incentives), and *Indikation* (indication).

```
Die Ausdauer des Jungen scheint gut zu sein.     0.94    Die Mutter sieht klar die Stärken ihres Sohnes
 und will ihn unterstützen.

Der Junge kann sich in eine Aufgabe vertiefen, wenn ihn etwas interessiert.      0.95    In diesem Gebi
et scheint Neugier und Interesse ganz aus ihm selbst zu kommen. Die Motivation im Sport wird nicht von
aussen erzeugt, und er erzielt dadurch wie auch durch seine Fähigkeiten im Bereich der Motorik schnelle
 Fortschritte.

Es scheint, als habe sich der Junge gerne bewegt.        0.95    Die Mutter sieht klar die Stärken ihre
s Sohnes und will ihn unterstützen.

Es scheint, als habe der Junge Freude an Rhythmen.       0.95    Scheint Stärken im Bereich von Rhythmu
s /Musik zu haben und will dies üben.

Die Mutter möchte dem Kind helfen.       0.96    Die Mutter sieht klar die Stärken ihres Sohnes und wil
l ihn unterstützen.

Der Junge ist zu begeistern, wenn es darum geht etwas zu gewinnen.       0.96    In diesem Gebiet schei
nt Neugier und Interesse ganz aus ihm selbst zu kommen. Die Motivation im Sport wird nicht von aussen e
rzeugt, und er erzielt dadurch wie auch durch seine Fähigkeiten im Bereich der Motorik schnelle Fortsch
ritte.

Die Handlungsplanung fällt ihm einfach.          0.95    Wenn er etwas will, kann er rasch Fortschritte
 erzielen.

Der Junge ist kreativ.   0.92    Andreas ist ein geschicktes Kind.

Die Eltern kümmern sich um den Jungen und können ihn gut wahrnehmen.     0.96    Die Mutter sieht klar
die Stärken ihres Sohnes und will ihn unterstützen.



********************************************************************************
FINAL REPORT
********************************************************************************
          CATEGORY                    NR. ELEMENTS                    AVERAGE
        Aufmerksamkeit                       0                             0
          Begabungen                         1                          0.92
            Schule                           0                             0
   Interessen / Motivation                   4                         0.952
       Externe Anreize                       0                             0
          Indikation                         0                             0
            Eltern                           4                         0.952
********************************************************************************
```

Figure 9: Command line output of `python psychomotor_diagnosis_main.py -s ressourcen -f students/Anamnese_1417_Eggel.f.tsv`.

5 Suggestions for future improvements

The software described in this study may be improved in a number of ways. Possible suggestions are:

- establish a standard on how experts have to write remarks. This way, too specific preprocessing steps – such as removing the *R.* and *P.* tokens at the beginning of remarks – would be avoided;
- provide experts with an interface where to write their remarks. This interface should also contain lists of categories and subcategories that can be associated to remarks. Such interface would add consistency to the anamneses, as the manual labeling of remarks made by those who worked on the original version of the project would be bypassed. This implies that some preprocessing steps we have implemented could be deleted to make the code more efficient (think for example at the removal of leading and trailing whitespaces around remarks categories), but also that the experts require less time to produce their remarks;

- perform spelling correction in a contextual way and not only with the Levenshtein distance;
- pay attention to the gender of the health professional and replace all terms referring to them in a consistent manner;
- develop a language-specific model for InferSent. This would avoid translating remarks from German into English;
- fine-tune SentenceBERT on in-domain data to have a better model. On the GitHub repository there is a first try of this fine-tuning (`sentencebert_finetuning_try.py`) starting from 34 remarks pairs manually labeled for their semantic similarity. These remarks are contained in a file created by the researchers of the original project (`AnamBeob_Ratings.tsv`);
- further develop all algorithms to obtain scores more representative of the real semantic similarity between student-expert remarks pairs;
- implement alternative algorithms such as Universal Sentence Encoder [Cer et al., 2018], Skip-thought [Kiros et al., 2015], and FastSent [Hill et al., 2016]. These algorithms may in fact perform better than those used in this study;
- give suggestions to students if their diagnosis is not complete, i.e., when not all categories associated with the expert's remarks are met;
- extend the final feedback to include also the subcategories of the experts' remarks;
- build a user-friendly interface that everyone can utilize. The algorithm, the full name of the patient, the list of abbreviations, and the threshold should be preset. The user must instead choose the category, the subcategory, and whether to save the output with scores and most similar remarks.

6 Conclusions

Resuming a project abandoned a few years ago, we implemented a computer-assisted training software for psychomotor diagnostics. We upgraded to 2021 standards the initial design and we included several preprocessing steps.

Psychomotor students cannot use the created software yet because it lacks a user-friendly interface. A contextual spelling correction is also missing. However, what has been done can be a good starting point to create an interface for real use.

This study focused on the field of psychomotricity and addressed diagnoses in German. Nevertheless, this work can be easily tailored to other fields of medicine and to all languages for which semantic similarity models are available. This software could therefore become a valuable aid for any medical therapist student in the world.

References

Arora, S., Liang, Y., and Ma, T. (2017). A simple but tough-to-beat baseline for sentence embeddings. In *ICLR*.

Barbosa, R. O. M. (2012). Between the psychomotricity and human development: the importance of physical education in early childhood education. *Efdeportes*, 169.

Berdilă, A., Talaghir, L. G., Iconomescu, T. M., and Rus, C. M. (2019). Values and interferences of psychomotricity in education - a study of the domain-specific literature. *Revista Romaneasca pentru Educatie Multidimensionala*.

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Caspar, F., Berger, T., and Hautle, I. (2004). The right view of your patient: A computer-assisted, individualized module for psychotherapy training. *Psychotherapy*, 41:125–135.

Cer, D., Yang, Y., Kong, S.-Y., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., Sung, Y.-H., Strope, B., and Kurzweil, R. (2018). Universal sentence encoder.

Chan, B., Schweter, S., and Möller, T. (2020). German's next language model.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., and Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680. Association for Computational Linguistics.

de Dombal, F. J., Hartley, J. R., and Sleeman, D. H. (1969). A computer assisted system for learning clinical diagnosis. *Lancet*, 1(7586):145–148.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding.

Ericsson, K. A., Krampe, R. T., and Tesch-Römer, C. (1993). The role of deliberate practice in the acquisition of expert performance. *Psychological Review*, 100(3):363–406.

Graves, A. and Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional lstm networks. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 4, pages 2047–2052 vol. 4.

Hill, F., Cho, K., and Korhonen, A. (2016). Learning distributed representations of sentences from unlabelled data.

Kiros, R., Zhu, Y., Salakhutdinov, R., Zemel, R. S., Torralba, A., Urtasun, R., and Fidler, S. (2015). Skip-thought vectors.

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1188–1196. PMLR.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space.

Petermann, F. (2015). *M-ABC-2, Movement Assessment Battery for Children*. Pearson.

Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks.

Teofili, T. (2019). *Deep Learning for Search*. Manning Publications.

Williams, A., Nangia, N., and Bowman, S. (2018). A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.