

FAMULUS: Interactive Annotation and Feedback Generation for Teaching Diagnostic Reasoning

Jonas Pfeiffer¹, Christian M. Meyer¹, Claudia Schulz¹,
Jan Kiesewetter², Jan Zottmann², Michael Sailer³, Elisabeth Bauer³,
Frank Fischer³, Martin R. Fischer², Iryna Gurevych¹

¹ Ubiquitous Knowledge Processing (UKP) Lab, Technische Universität Darmstadt, Germany

² Institute of Medical Education, University Hospital, LMU München, Germany

³ Chair of Education and Educational Psychology, LMU München, Germany

<http://famulus-project.de>

Abstract

Our proposed system FAMULUS helps students learn to diagnose based on automatic feedback in virtual patient simulations, and it supports instructors in labeling training data. Diagnosing is an exceptionally difficult skill to obtain but vital for many different professions (e.g., medical doctors, teachers). Previous case simulation systems are limited to multiple-choice questions and thus cannot give constructive individualized feedback on a student’s diagnostic reasoning process. Given initially only limited data, we leverage a (replaceable) NLP model to both support experts in their further data annotation with automatic suggestions, and we provide automatic feedback for students. We argue that because the central model consistently improves, our interactive approach encourages both students and instructors to recurrently use the tool, and thus accelerate the speed of data creation *and* annotation. We show results from two user studies on diagnostic reasoning in medicine and teacher education and outline how our system can be extended to further use cases.

1 Introduction

Motivation. Supporting students in learning has been the life purpose of many teachers throughout history. With the growing number of people who choose an academic path, it becomes increasingly important to leverage automatic methods to guide students and give them individualized feedback.

However, existing systems for technology-enhanced learning, mostly address skills on recalling, explaining, and applying knowledge, e.g., in automatically generated language learning exercises (Madnani et al., 2016) and math word problems (Koncel-Kedziorski et al., 2016). More complex cognitive tasks such as diagnostic reasoning require analytic and decision-making skills, for which there are yet only few solutions, even though

diagnostic skills are vital for many professions (e.g., medical doctors searching for a therapy, teachers identifying potential mental disorders at an early stage, engineers diagnosing a machine failure, etc.).

Training diagnostic skills is hard and typically relies on time-consuming and hard-to-control live role-plays. Online case simulations involving so-called *virtual patients* crystallized as an effective alternative to role-playing games (Berman et al., 2016; Jin et al., 2018). In case simulations, students collect information on a virtual patient across multiple screens, e.g., from patient–doctor dialogs, lab results, and medical imaging. To date, the students formulate their final diagnosis by means of multiple-choice questions, which are easy to assess, but prevent important analyses of the effectiveness and the efficiency of the diagnostic reasoning process. This is why we propose to complement multiple-choice questions with prompts asking for explanations of the students’ thought process. The open-form textual explanations enable good insight into the diagnostic reasoning process rather than only its result, leaving room for constructive methodological feedback. However, the text analysis and feedback generation components are highly complex and require advanced Natural Language Processing (NLP) techniques.

Contributions. To tackle this task, we propose our FAMULUS system to generate individual feedback on the students’ diagnostic skills. FAMULUS integrates (a) state-of-the-art neural sequence labeling models to generate individualized feedback, incorporated in our novel NeuralWeb service, and (b) a corpus construction tool enabling interactive model training with (c) an existing tool for conducting case simulations. The backbone of our system is a sequence labeling approach to identify fine-grained diagnostic entities (e.g., *liver values*, *blood pressure*) and epistemic activities (Fischer et al.,

☑ **Textaufgabe**

Die Körperliche Untersuchung war unauffällig. Allerdings waren im Labor die Entzündungswerte und Leberwerte auffällig. Der dicke Tropfen war negativ, daher war **Malaria** als Diagnose ausgeschlossen. Die Hepatitis Serologie war positiv und damit die Diagnose gesichert.

Vielen Dank für Ihre Antwort!

Fallübersicht

Die 36-jährige Frau Hoffmann stellt sich vor, mit einem seit einer Woche bestehenden grippalen Infekt. Als zusätzliche Symptome gibt sie Abgeschlagenheit, Appetitverlust, Übelkeit und Diarrhoe an. Sie war vor einem Monat ins Sansibar, vor der Reise wurde eine Gelbfieberimpfung durchgeführt.

Rückmeldung zu Differentialdiagnose

- ☐ Bei einem einwöchigen grippalen Infekt mit Gliederschmerzen und Abgeschlagenheit wäre zunächst eine Influenza-Infektion denkbar gewesen. Für einen grippalen Infekt ist die Symptomatik allerdings zu langanhaltend, da dieser meist nach 3 Tagen abklingt.
- ☐ Bei einer Diarrhoe hättest du auch eine Darmerkrankung, wie die Gastroenteritis, vermuten sollen.

👁 Nicht schlecht, dass du eine Tropenkrankheit differentialdiagnostisch in Betracht gezogen hast. Möglich wären zB. Malaria, Dengue Fieber, Cholera etc.

Figure 1: Excerpt of the generated feedback (bottom) to a student’s explanation of her/his diagnostic process (top). Blue responses are covered by the text and highlighted in yellow. Black responses are not covered by the diagnosis.

2014) such as *drawing conclusions* and *evidence evaluation*, in German text. We have previously published the scientific foundation of our system (Schulz et al., 2018b,a, 2019a), but introduce the technical aspects of the system for the first time. Additionally, for the first time, we evaluate its applicability to real-time use cases. Our evaluation results on the prediction quality and the annotation effectiveness are based on two user studies with German medicine (Med) and teacher education (TEd) students. We show how instructors are relieved from the burden of pre-annotating huge amounts of data by our interactive annotation workflow, and we discuss the generated individualized feedback that helps students acquire diagnostic skills. Finally, we explain how our system can be obtained, re-used, and extended to further use cases.

2 Case Simulation Example

Imagine a Med student training her/his skills with our system. She/he receives information about a virtual patient: The 36 years old Ms. Hoffmann reports about a common cold lasting for about one week. In an interview, she mentions abnormal fatigue, diminished appetite, nausea, and diarrhea. Further questioning reveals that she stayed in Sansibar about a month ago. Prior to her travel, she was vaccinated against yellow fever.

Based on such inputs and further lab results, the student explains her/his diagnosis (see Figure 1): “Physical examination was clinically unremarkable. But the lab results show noticeable inflammation markers and liver values. Thick blood film was negative, therefore Malaria was excluded as a diagnosis. Hepatitis serology was positive, which assures the diagnosis.”

In order to automatically provide feedback, we define a set of *diagnostic classes* covering fine-grained diagnostic entities related to the case (e.g.,

names of diseases, medical examinations, therapies) manually defined by domain experts, and epistemic activities (Fischer et al., 2014) that characterize the reasoning process. As epistemic activity classes, we use *hypothesis generation* (HG; the derivation of possible answers to the problem), *evidence generation* (EG; the derivation of evidence, e.g., through deductive reasoning or observing phenomena), *evidence evaluation* (EE; the assessment of whether and to which degree evidence supports an answer to the problem), and *drawing conclusions* (DC; the aggregation and weighing of evidence and knowledge to derive a final answer to the problem) discussed by Schulz et al. (2019a).

FAMULUS analyzes the previously mentioned diagnostic text and returns feedback on multiple important aspects related to the case. It successfully detects all aspects verbalized in the text (e.g., the discussion of tropical diseases; marked in blue in Figure 1). Aspects that are not addressed in the text are discussed and provide additional input to what the student has missed (e.g., that the differential diagnosis should consider a potential bowel disease due to the diarrhea). For the present example, the student correctly diagnoses a Hepatitis variant (correct would be Hepatitis A), which is positively acknowledged in the generated feedback. In the supplementary video material, we show two original German diagnostic texts and the corresponding feedback generated by our system.

3 System Architecture

FAMULUS consists of three intercommunicating components introduced in this section.

3.1 NeuralWeb

NeuralWeb¹ is a Python-based web service that communicates with all other components and thus

¹github.com/UKPLab/emnlp2019-NeuralWeb

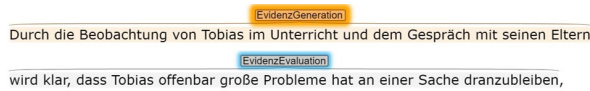


Figure 2: Annotation suggestion (grey) and accepted suggestion (orange) in the INCEpTION platform.

resembles the core of our system. It is responsible for interactive training and prediction of the diagnostic classes and for the generation of individualized feedback. We divide its functionality in a *model* and a *feedback DB* part, encompassed by a *wrapper class* that can be easily adapted for new machine learning methods and case studies.

Model. The wrapper class includes a loading function which leverages the downstream model architecture and copies the respective weights into memory. The supported neural architectures are written in Keras², and PyTorch³ and are therefore easy to adapt. NeuralWeb currently provides a recent BiLSTM architecture (Reimers and Gurevych, 2017) implemented in Keras and Flair (Akbik et al., 2018) implemented in PyTorch, which holds the current state of the art on many sequence-labeling tasks. A prediction function of the wrapper preprocesses a text (sentence splitting and tokenization using NLTK) and leverages the pre-trained model to predict and return the diagnostic classes.

NeuralWeb additionally enables automatic re-training of the model within the framework which is useful when new data has been generated and annotated, improving the model automatically. This functionality is currently implemented for the Keras-based model.

Feedback DB. The output of the model is a set of discrete diagnostic class labels, which hardly yield valuable feedback for users. It is thus essential to provide an additional description, indicating whether or not the diagnosis is correct, what is missing, and if the diagnostic process is sound. We thus introduce a Feedback DB, which includes descriptive text snippets written by experts. These descriptions are associated with diagnostic classes predicted by the model and a specific case study. For example the patient in case 1 *has* Hepatitis A, whereas the patient in case 2 *does not*. The feedback for a student who diagnoses Hepatitis A thus needs to be different with respect to the case she/he currently works on. The Feedback DB is

²keras.io

³pytorch.org

an independent resource queried by the wrapper class. With respect to the predicted labels, the corresponding feedback text will be generated. FAMULUS finally returns the labeled text spans of the diagnostic text together with the feedback text indicating the reasons for the prediction.

3.2 INCEpTION

Expert annotation by instructors is required due to the complexity of labeling diagnostic texts. For this reason, we leverage the INCEpTION text annotation platform (Klie et al., 2018) which enables interactive semantic annotation. The *recommender* system which provides instructors with automatically generated annotation suggestions is one of the key functionalities of the platform. Suggestions can be obtained from various integrated classifiers as well as from external sources such as NeuralWeb. The platform uses the user feedback (accepted/rejected annotations) as well as user-created annotations to continually improve the classifiers. We leverage this functionality to create an efficient interactive annotation process for our diagnostic classes and thus to create training data for our NLP models. Figure 2 shows an example of the labeling process with suggestions by our pre-trained model.

3.3 CASUS

CASUS⁴ is an interactive system designed for case simulations with virtual patients. It incorporates all aspects necessary for conducting diagnostic case simulations (i.e., videos, images, text, audio integration). Students receive information relevant for solving the case. They are subsequently required to formulate their diagnosis in multiple-choice questions and our new, free-text prompts, directly integrated in CASUS. After submission, CASUS presents the feedback received from NeuralWeb.

While CASUS is a sophisticated proprietary simulation tool, we would like to stress that this is not a requirement. FAMULUS can be used with any open-source front-end tool providing a text box and communicating with NeuralWeb to print out the individualized feedback. We provide a simple version of such a tool in our GitHub repository together with NeuralWeb.

4 FAMULUS Process

The FAMULUS system consists of an interactive learning cycle connecting the three components

⁴www.instruct.eu

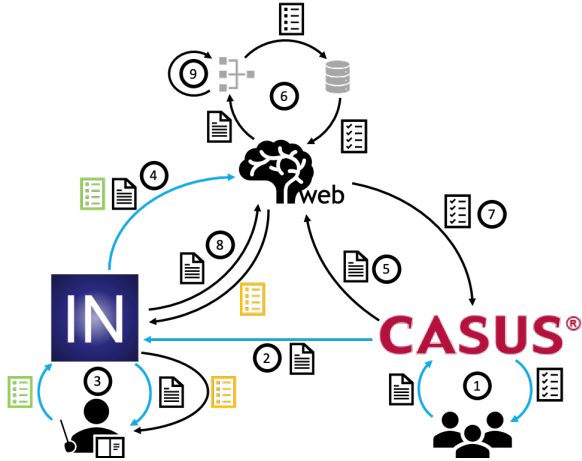


Figure 3: FAMULUS process for annotating data, training models, and generating individualized feedback. During the cold-start phase, only the blue edges are used, until enough labeled data exists to train a model.

introduced in §3 and illustrated in Figure 3.

Cold-Start. Because a small initial set of annotated data is necessary to train a preliminary model in NeuralWeb, ① few pilot users first submit their diagnoses to the CASUS system. In this cold-start phase, the users either do not receive *any* feedback or a *default* feedback text. For our experiments, all users receive a default gold diagnosis written by experts, for the users to compare their results manually. ② The students’ diagnostic texts are sent to INCEption, where ③ instructors label the data according to the predefined annotation schema. ④ The gold labels (visualized in green) are stored and sent to NeuralWeb. Using this labeled training data, we can train our models to automatically predict the diagnostic classes found in a given text.

Warm-Run. After an initial model has been trained, a new set of users can benefit from the trained model to receive individualized feedback. ① Similarly to the cold-start phase, users work through the case study and submit their diagnosis to the CASUS system. ⑤ Instead of receiving a default feedback, the diagnosis is sent to NeuralWeb. ⑥ NeuralWeb processes the text through the trained model and generates individualized feedback with regards to the Feedback DB. ⑦ The individualized feedback is sent back to CASUS which visualizes it for the user.

Like in the cold-start phase, ② the diagnostic text is also sent to INCEption. ⑧ But instead of relying solely on the instructor, the trained model in NeuralWeb predicts preliminary annotations (de-

		EG	EE	HG	DC
Med	BiLSTM	71.60	80.20	69.28	65.32
	UB	85.61	90.25	86.37	85.58
TEd	BiLSTM	78.53	78.87	57.16	61.77
	UB	93.29	90.71	81.77	82.11

Table 1: Individual macro-F1 scores following Schulz et al. (2019a) for each of the epistemic activities. The BiLSTM uses FastText embeddings (Bojanowski et al., 2017). This architecture is equal to Flair when only using FastText embeddings. UB reports the human upper bound (inter-annotator agreement) indicating room for improvement.

noted in yellow) which are additionally presented to the instructor (see Figure 2). ③ These predictions should increase the labeling speed, as in many cases, the instructor simply has to accept the suggestions the model has predicted. ④ The validated (green) labels are sent back to NeuralWeb and ⑨ the model can be interactively retrained for each additional data point which has been labeled.

5 Evaluation

We employ our proposed FAMULUS architecture in two studies yielding 1,107 Med and 944 TEd diagnostic texts written for eight distinct cases per domain. While a full analysis of the two studies is beyond the scope of this paper, we focus on three research questions highly relevant to the systemic aspects of FAMULUS: (1) the quality of the predicted diagnostic classes, (2) the computation time of the prediction and feedback generation system to assess the applicability of our system in real-time applications, and (3) the benefits of providing annotation suggestions to the instructors.

Prediction quality. In Table 1, we report the performance of the BiLSTM implementation for predicting epistemic activities in the Med and TEd data. As we can see, the difficulty of predicting the classes varies between different activities. Despite some room for improvement with respect to the human upper bound (UB) based on inter-rater agreement, the interactive nature of FAMULUS helps in succeeding in this attempt by continually improving the model when new data is available.

We conduct similar experiments for the prediction of fine-grained diagnostic entities, but omit a comprehensive discussion due to space limitations.

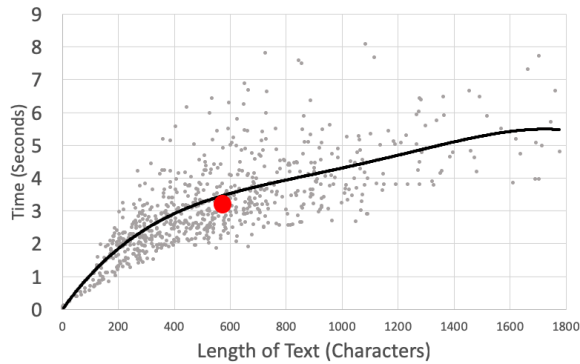


Figure 4: Prediction time for the submitted diagnostic texts of TED students. The illustrated model is Flair with character and Flair embeddings, hidden size 256, 2 layer BiLSTM with CRF-head. The red dot indicates the mean time and length of 3.15 seconds and 562 characters respectively. The trendline is illustrated in black. The experiment was conducted on a 13-inch 2017 MacBook Pro with i7 Processor and 16GB RAM.

Computation time. In order to present the feasibility of deploying FAMULUS in a real-time scenario, we plot the inference times of the submitted diagnostic texts in Figure 4. The inference time includes sentence splitting, tokenization, model prediction, and feedback generation using the Feedback DB. We find that on average the submitted texts have a length of 562 characters with an average inference time of 3.15 seconds on a common desktop machine. The different inference times for similar text lengths are due to variable sentence lengths, as longer sentences require more inference time. We batch all sentences of one diagnostic text and pass them through the model simultaneously. As we can see in the graph, the automatic feedback generation does not surpass 9 seconds. This is intuitively faster than any human is able to read, process, and output feedback text, even by leveraging prewritten descriptions. This demonstrates the effectiveness and scalability of FAMULUS in a real-time scenario.

Annotation suggestions. To evaluate the effects of providing annotation suggestions, we have conducted an extensive study (Schulz et al., 2019b) considering annotation time, annotation quality, potential biases, and the ease of use. To this end, we asked five Med and four TED instructors to annotate diagnostic texts. Two of the instructors per domain did not receive any suggestions. For the remaining instructors, we provided suggestions in multiple phases using different models and setups. Overall,

we find positive effects yielding a speed-up of 34 to 42 seconds per annotated text. The instructors accept 56 % (Med) and 54 % (TED) of the annotations. While we observe a slightly higher inter-annotator agreement if instructors receive suggestions, we also study whether the instructors’ decisions are influenced by the suggestions, but only observe a negligible effect.

6 Dissemination

In this section, we introduce how the components of our system can be obtained and linked with each other. Additionally, we discuss multiple use cases that can benefit from our architecture.

Availability. The NeuralWeb component is the heart of our system and has been newly developed for our purposes. We make NeuralWeb available as open-source software in our GitHub repository under the Apache License 2.0. We integrate the annotation suggestions generated by our system into the INCEpTION annotation tool, which is available as open-source software under the Apache License 2.0. To conduct the case simulations, we use the CASUS system which can be obtained from its publisher Instruct. We provide a simple but free alternative to CASUS which includes only the necessary functionality for the FAMULUS system, which is to write the diagnostic text and visualize feedback. This system, together with connection functionalities to INCEpTION and CASUS, can be found in our NeuralWeb repository.

For using FAMULUS, a server or virtual machine is needed on which the system is deployed. A thorough description can be found in our GitHub repository, including the respective URLs and ports that need to be adapted.

Use cases. Our proposed architecture is primarily useful to prepare and conduct case simulations that train diagnostic skills based on text analysis and automated feedback generation methods. Besides developing new cases for the Med and TED domains which is the subject of our research, case simulations can be useful for students in engineering (e.g., diagnosing a machine failure), law (investigating evidence in a lawsuit), economy (optimizing business processes), and many more.

In order to leverage the FAMULUS system, three prior steps need to be made, independent of our system: (1) Expert instructors develop a set of case studies, for which they provide all necessary in-

formation. The case study can be integrated into a simulation tool such as CASUS or provided in printed form. (2) The instructors define an annotation schema, i.e. what kinds of diagnostic classes should be annotated (e.g., observations of teachers in a classroom). (3) As the individualized feedback can vary from case to case, corresponding descriptions need to be defined by the instructors.

7 Conclusion

In this paper, we have introduced FAMULUS, a case simulation system integrating interactive data acquisition and model training, and individualized feedback generation for students' explanations of diagnostic reasoning processes. Our analysis shows how FAMULUS helps experts in annotating data fast and reliable while successfully predicting entities and activities occurring in diagnostic texts. FAMULUS is applicable in real-time scenarios and generates feedback much faster than humans. While we focus specifically on diagnostic case simulations in medicine and teacher education, we outline the steps necessary to adapt our approach to many other disciplines requiring the training of diagnostic skills. We open-source all components necessary to employ FAMULUS in new case studies, hoping to encourage more research in this area.

Acknowledgments

This work has been supported by the German Federal Ministry of Education and Research (BMBF) under the reference 16DHL1040 (FAMULUS).

References

- Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. [Contextual string embeddings for sequence labeling](#). In *Proceedings of the 27th COLING*, pages 1638–1649.
- Norman B. Berman, Steven J. Durning, Martin R. Fischer, Soren Huwendiek, and Marc M. Triola. 2016. [The Role for Virtual Patients in the Future of Medical Education](#). *Academic Medicine: Journal of the Association of American Medical Colleges*, 91(9):1217–1222.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. [Enriching Word Vectors with Subword Information](#). *Transactions of the Association for Computational Linguistics*, 5:135–146.
- Frank Fischer, Ingo Kollar, Stefan Ufer, Beate Sodian, Heinrich Hussmann, Reinhard Pekrun, Birgit Neuhaus, Birgit Dorner, Sabine Pankofer, Martin R. Fischer, Jan-Willem Strijbos, Moritz Heene, and Julia Eberle. 2014. [Scientific Reasoning and Argumentation: Advancing an Interdisciplinary Research Agenda in Education](#). *Frontline Learning Research*, 4:28–45.
- Lifeng Jin, David King, Amad Hussein, Michael White, and Douglas Danforth. 2018. [Using Paraphrasing and Memory-Augmented Models to Combat Data Sparsity in Question Interpretation with a Virtual Patient Dialogue System](#). In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*, pages 13–23.
- Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. [The INCEPtion platform: Machine-assisted and knowledge-oriented interactive annotation](#). In *Proceedings of the 27th COLING: System Demonstrations*, pages 5–9.
- Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. 2016. [MAWPS: A Math Word Problem Repository](#). In *Proceedings of the 2016 Conference of the NAACL*, pages 1152–1157.
- Nitin Madnani, Jill Burstein, John Sabatini, Kietha Biggers, and Slava Andreyev. 2016. [Language Muse: Automated Linguistic Activity Generation for English Language Learners](#). In *Proceedings of the 54th Annual Meeting of the ACL: System Demonstrations*, pages 79–84.
- Nils Reimers and Iryna Gurevych. 2017. [Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging](#). In *Proceedings of the 2017 Conference on EMNLP*, pages 338–348.
- Claudia Schulz, Christian M. Meyer, and Iryna Gurevych. 2019a. [Challenges in the Automatic Analysis of Students' Diagnostic Reasoning](#). In *Proceedings of the 33rd AAAI*, pages 6974–6981.
- Claudia Schulz, Christian M. Meyer, Jan Kiesewetter, Michael Sailer, Elisabeth Bauer, Martin R. Fischer, Frank Fischer, and Iryna Gurevych. 2019b. [Analysis of Automatic Annotation Suggestions for Hard Discourse-Level Tasks in Expert Domains](#). In *Proceedings of the 57th Annual Meeting of the ACL*, pages 2761–2772.
- Claudia Schulz, Christian M. Meyer, Michael Sailer, Jan Kiesewetter, Elisabeth Bauer, Frank Fischer, Martin R. Fischer, and Iryna Gurevych. 2018a. [Challenges in the Automatic Analysis of Students' Diagnostic Reasoning](#). arXiv:1811.10550.
- Claudia Schulz, Michael Sailer, Jan Kiesewetter, Elisabeth Bauer, Fischer Fischer, Martin R. Fischer, and Iryna Gurevych. 2018b. [Automatic recommendations for data coding: A use case from medical and teacher education](#). In *Proceedings of the 14th IEEE International Conference on e-Science*, pages 364–365.