DAVE: Differential Diagnostic Analysis Automation and Visualization from Clinical Notes

Hadi Hamoud and **Chadi Abou Chakra** and **Mira Dankar** and **Fadi A. Zaraket**
Electrical and Computer Engineering Department, American University of Beirut
hkh21,cha21,mkd07@mail.aub.edu, fz11@aub.edu.lb

Abstract

Electronic Medical Records are integral parts of modern healthcare. Part of the records are *clinical notes* that healthcare providers take during encounters with patients. Notes are key to *differential analysis* which is the reasoning process leading to diagnosis and treatment. This paper presents DAVE, a *differential analysis automation and visualization* to assist healthcare professionals through the differential analysis process. DAVE takes as input clinical notes as they are being written by professionals and suggests candidate diagnostic algorithms. We digitized textbook diagnostic algorithms into directed acyclic graphs. We trained a distributional semantics model using an annotated corpora of electronic medical records and text from diagnostic algorithm descriptions. The model, boosted with PUBMed-based semantic similarity metrics, ranks the diagnostic algorithm graphs and suggests the top three. The model achieved 74.3% success rate and was highly accepted by multiple medical professionals for usability.

1 Introduction

Information reported in *electronic medical records* (EMRs) revolutionized medical language research. Healthcare providers follow specific procedures in the process of caring for and managing a patient. The clinical assessment starts by noting the *chief complaints* of the patient and the purpose of the visit. This step is followed by a review of family history and symptoms. The healthcare providers document the aforementioned information in *clinical notes* embedded in an EMR. They proceed with *differential analysis* leading to a diagnosis of the case on hand and a declaration of the future actions to be taken.

The adequate decision comes from following a specific set of evidence-based *diagnostic algorithms* that medical professionals learn during their education and training. These algorithms are decision diagrams whose top nodes are labeled with markers and identifiers that occur typically in the chief complaints.

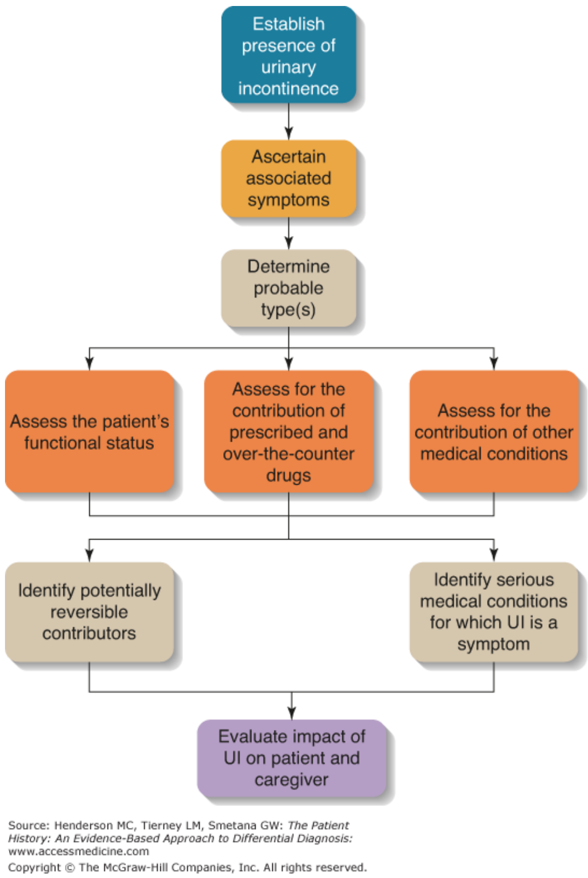

Figure 1: Diagnostic algorithm extracted from "Symptom to Diagnosis: An Evidence-based Guide"

The algorithms lead the healthcare providers with a series of choice nodes reaching to the leaves which typically dictate the diagnosis and the treatment plan. Intermediary nodes in the algorithms describe lab tests, medication prescriptions, special treatments, and life-style changes among other actions. The healthcare providers select the algorithms to follow after assessing the situation from discussions with the patient, initial clinical tests and documentation in the clinical notes and EMRs.

They select and follow the algorithm that best

Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics System Demonstrations, pages 307–314
May 2-4, 2023 ©2023 Association for Computational Linguistics

matches the encountered case. They traverse the decision diagram of the algorithm, come up with an adequate diagnosis, and declare further actions to be taken.

The healthcare provider documents the information collecting during the encounter with the patient in a clinical note summarizing the process from the initial complaint to the actual outcomes. Our aim is to assist the healthcare professionals in the process of coming up with adequate diagnosis by filtering out the top three diagnostic algorithms matching their notes and visualizing them.

In this paper, we present DAVE (Differential Analysis Visualizer for EMRs), a system that automates the selection and visualization process of the diagnostic algorithms. DAVE alleviates healthcare providers from the tedious tasks that require remembering and visualizing the graphs. It also simplifies the decision-making process involved in the *diagnostic differential analysis* (DDA) phase. DDA consists of analyzing in real time the input notes and selecting the most suitable algorithms.

We developed and fine-tuned DAVE via training on pre-existing curated datasets. We leveraged an annotated corpora of electronic medical records collected from AUBMC and the Hariri Medical Center in Beirut, Lebanon. The corpora consist of 151,930 total medical notes focused on family medicine. All the notes are annotated with diagnostic codes and treatment plans that annotate the whole record. A subset of 3,616 of the notes are richly annotated with textual annotations referring to textual elements in the note itself.

DAVE also leverages digitized clinical diagnostic algorithms extracted from medical books: "The Patient History: Evidence-Based Approach" (Henderson et al., 2012a) and "Symptom to Diagnosis: An Evidence-based Guide" (Stern et al., 2019). These textbook algorithms represent the steps that need to be taken to determine and treat a specific healthcare condition. Our method involves *cross-document* analysis based on cross-referencing electronic medical record entities. This aims at extracting diagnosis indicators from the notes using natural language processing (NLP) and computational linguistics (CL).

We presented DAVE to a number of medical professionals, physicians, and medical IT experts and received overwhelmingly positive feedback. We demonstrated the project to three physicians and interviewed them about usability and functionality. After using DAVE, they provided positive feedback and emphasized its usefulness, particularly for students, nurses, and young physicians. We also interviewed the head of IT at AUBMC, a well-established medical center, and her team. She also praised the software and encouraged its integration within the medical center for professional use.

2 Related Work

Information extraction (IE) from EMR has become a crucial tool to progress medical and clinical practice research since the emergence of digital records. The understanding gained from this data has a significant positive impact on current medical research. For this reason , a number of studies and initiatives have looked into the best ways of extracting data from medical records. Rule-based algorithms, machine learning (ML) models and keyword-based search are the major methodologies used in IE from EMRs, with rule-based models being the most accurate but most time consuming to build. This is due to the fact that the rules express accurately the direct knowledge and experience of healthcare professionals (Ford et al., 2016; Wang et al., 2018).

Several reviews discussed the different approaches used for IE in the medical field (Ford et al., 2016; Wang et al., 2018; Deléger et al., 2010). DAVE will be built using ML as its core. It will mimic the behavior of rule-based methods by capturing insight from clinical diagnostic books. It avoids the cost of rule-based approaches since healthcare professionals will not have to manually provide rules.

The work (B.Sharafeddin, 2020) introduced a model for the automation of the process of diagnostic extraction from clinical notes. It matches unstructured de-identified medical notes to medical diagnostic algorithms using a cross-document analysis method. The model uses the diagnostic algorithms (Henderson et al., 2012b) from medical books to build Bayesian Networks corresponding to every diagnosis case. In these networks, nodes are interrelated by informational dependencies.In other words, each node is given a conditional probability depending on the probability of its parent (Pearl, 2011).

Afterwards, it calculates the distributional similarity for the words in the electronic medical notes and United States Medical Licensing Examination (USMLE) questions using DISCO (extracting DIStributionally related words using CO-occurrences)

(Kolb, 2008).

This method allows the extraction of diagnosis indicators by retrieving the semantic similarity between words and phrases in large text fields to create sets of similar words for every word (B.Sharafeddin, 2020).

In order to compute *semantic similarity*, we also used DISCO, also known as the KOLB (after Peter Kolb) vector similarity. DISCO retrieves the most semantically similar words for an input word (Kolb, 2008). It is accompanied with the DISCO builder tool (DBuilder). The latter creates a database of contextually similar words, given a text corpus. Prior to the actual build, DBuilder recommends configuring the corpus to the given format to ensure the best possible outcome. DBuilder accepts lemmatized or tokenized input in single or multiple files. Larger contexts (or paragraphs as referred to by the documentation) should be specified using tags. DBuilder takes care of excluding stop-words from the context as they can contribute to noise. It is also important to specify the size of the context each word should be taken in when configuring the builder instance. (Kolb, 2008). Giving a ± 3 to the context window will check the surrounding three words from the right and left of each target term. Subsequently, DBuilder creates a matrix of co-occurrences in which each row describes a specific word.

The size of the resulting matrix would be $n \times m \times r$ where n denotes the number of words in the corpus, m denotes the number of words used as features and r is the window size.

Equation 1 is used to provide meaningful weights for the features in which w, and w' stand for words, r stands for the window size, $-$ within functional parameters stands for dependency relation and f is the frequency of occurrence.

$$\frac{\log(((f(w,r,w') - 0.95) * f(-,r,-))}{f(w,r,-) * f(-,r,w'))} \tag{1}$$

The information theory based Lin's measure is then used for the comparison of every word vector with all other word vectors to create a distributional similarity scheme between all the words (Kolb, 2008).

Negated words present in the medical notes lead to faulty results when matching them to the corresponding diagnostic algorithm in our work. The ConText algorithm provides an approach to deal with negated words by employing a specific scope of its trigger terms. Once medical conditions are indexed, the algorithm assigns three contextual properties to each condition being (i) Negation, (ii) Temporality and (iii) Experiencer. Negation can be either negated or affirmed. Temporality can be recent, historical, or hypothetical. Finally, experiencer can be either patient or other. A set of trigger words is assigned to each non-default status (Harkema et al., 2009).

The status of the contextual properties is updated when the indexed medical condition term falls within the scope of one of the trigger terms. Moreover, the algorithm contains pseudo-trigger words for each non-default status of the contextual properties. The scope of the trigger or pseudo-trigger terms includes all the clinical conditions following them till the termination word or till the end of the sentence if no termination word was present. The termination words are assembled in the algorithm following conceptual groups. In other words, the algorithm takes as input the sentence, indexes the clinical conditions, and assigns each of the contextual properties a default value. Then, it marks all the trigger, pseudo-trigger and termination words and iterates through them to determine the scope of each trigger term and update the corresponding contextual property of the indexed medical case.

3 Methods

This section discusses the preprocessing of EMR notes, building and utilising DISCO models, aggregating scores from different models, capturing user input, and visualizing the graphs.

3.1 Preprocessing

Since medical text contains a large amount of noise that affects the results of an NLP model we executed a series of preprocessing and data cleaning before inputting the data in our model.

3.1.1 Abbreviations

Medical doctors tend to use numerous abbreviations when documenting their clinical notes given the overloaded schedule, heavy workload, and overall workflow efficiency. In the aim of simulating a real medical note, a medical data set for abbreviations had to be included. This dataset was used to replace all the abbreviations in the medical corpora by their actual meaning. When processing the user's input, tokens are then matched with their full word if applicable.

3.1.2 Autocorrection

Apart from using abbreviations, potential spelling mistakes were accounted for in both the training and the user data. DAVE used Levenshtein's method of distance metrics for auto-correction through the PySpellChecker implementation. Theoretically, the Levenshtein distance is a metric for measuring the differences between two strings treated as two separate sequences (Yujian and Bo, 2007). This metric creates a matrix by looping over all the letters of the two input strings it takes according to the rule in figure 2 shown below.

$$\operatorname{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j)=0, \\ \min \begin{cases} \operatorname{lev}_{a,b}(i-1,j)+1 \\ \operatorname{lev}_{a,b}(i,j-1)+1 \\ \operatorname{lev}_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

Figure 2: Levenshtein Distance Rule for Matrix Creation

The size of the matrix would be $(m+1)\times(n+1)$ where m and n are the sizes of the tested words, respectively. The matrix is filled from the upper left cell initialized to zero to the lower right entry with the actual distance between both words (Yujian and Bo, 2007).

This method proved to be the fastest for offline and dynamic performance. It uses the permutations within a 2 edit distance radius from the original word and returns the most likely correct result. However, in the case of medical autocorrection, many medical terms and abbreviations were treated as spelling mistakes and corrected accordingly. Therefore, a set of medical terms was included to be interpreted by PySpellChecker before making the decision to correct any given token. The process goes as follows:

1. The spell checker goes through the tokenized words and flags the included unknown words.

2. The unknown words are then filtered by the set of medical terms.

3. If not found to be medical terms, another pass is done to determine if the word is an abbreviation of a medical or an English term.

4. If the word is still not found, auto-correction is then applied.

For the training medical notes corpora, all of the tokenized words were checked in the described manner. However, for the dynamic user input, if the word is pre-computed and available in the model, it is automatically used for scoring and not passed for spell checking.

3.1.3 Negation Extraction

DAVE handled negated sentences featured in training medical notes by extracting and analyzing them on a context level rather than a token level by virtue of DISCO. However, negation can contribute to noise when analyzing the user's input. Therefore, DAVE adopted a conservative negation extraction technique. If an input token symbolizes negation, the following word is excluded from matching. For instance, in the sentence: "No Fever", the word "Fever" would be removed as it might lead to a faulty diagnosis conclusion if included(Mehrabi et al., 2015). The negation operators were identified within context from a precomputed NegSpaCy negation list.

Furthermore, we experimented with the ConText algorithm approach (Harkema et al., 2009), implemented as part of the MedSpacy library (Eyre et al., 2021) in the negation extraction scheme. The scheme returned good results in terms of accuracy and relevance. However, it is not yet implemented in the application and requires additional work to ensure optimization. The algorithm is in consideration for future work.

3.2 DISCO, PubMed word spaces

The DISCO (Kolb, 2008) linguistic tools compute the distributional similarity between given words in a left-right context array, given the start and the end of the context. DISCO takes input in tokenized or lemmatized format. The clinical notes corpora gathered from the Medical Institutes, is processed through DISCO Builder to create a word space out of the given lemmatized/tokenized documents. DBuilder relies on tags to determine the full corpora, and the seperate contexts at hand. We configured the medical corpora by lemmatizing the notes and seperating them into "contexts", where each note detail was considered to be one context. From there, every given word is analyzed based off a specific context window (i.e. words to its left and right). Word vectors are created based off contextual similarity. The weighting methods and the similarity measures are following Lin's measure and the KOLB methods respectively. The output of DBuilder is a word space, packaged through an indexing schema, ready to be loaded through the

DISCO java library to output scores. With a corpus size of $7,060,230$ feature words, $2,122,706$ after filtering, the output word space included a $136,145$ target feature words.
In order to provide better accuracy metrics, we also utilized a pre-computed word space from PubMed, with over 181 million tokens and feature words. This word space is built using approximately $100,000$ medical articles from the PubMed Open Access database.

3.3 Score Aggregation

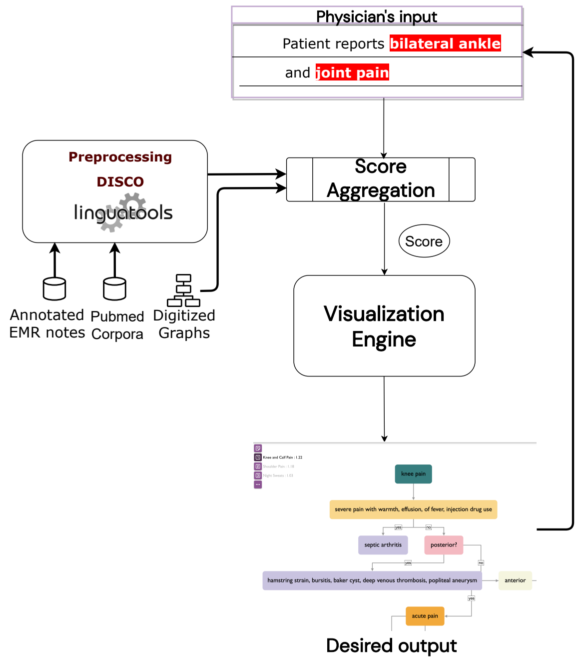

Figure 3: Illustration of DAVE's Pipeline

To generate diagnostic predictions, DAVE utilizes a map from the diagnostic algorithm nodes to the words present in these nodes featured. DAVE tokenizes the feature words and assigns the peak word score at 1. Every instance of a given term is stored accordingly, including repeated occurrences within the same graph.

The feature words are then augmented using the DISCO word space model. Each word featured originally in the map is passed through the indexed word space to retrieve the most semantically similar words within two collocation contexts.

1. The given word with the clinical note corpora word space.

2. The given word with the PubMed corpora word space.

The top three most semantically similar terms and their respective similarity scores are fed back into the model, and are matched with the same graphs as the original input term, but with the scores provided by DISCO.

To further refine the scores and reduce the noise caused by frequently occurring terms, DAVE applies a score modification scheme. The modified score considers the frequency of the term. This modified score assigns a lower weight to frequently occurring terms and a higher weight to less frequent terms, thus improving the accuracy of DAVE by giving more weight to terms specific to a particular diagnosis.

Finally, according to the user's input the score of each term is aggregated to provide a final score for each potential diagnosis. DAVE determines the score of each diagnostic graph by summing the scores of its nodes. This scoring aggregation scheme helps DAVE to accurately identify the most relevant diagnoses for a given clinical note.

3.4 Capturing User Input

DAVE takes the clinical notes written by the user as input, tokenizes them, and matches them with the feature words inside the computed DISCO models. With each tokenized input word w, if present, DAVE proceeds to score each diagnostic algorithm a based on the occurrences of w inside a. This leads to incrementing the scores of the involved algorithms, progressively forming a leaderboard of the top matching algorithms to the user's live input. The top matching algorithms are then transmitted to the user's interface graphing engine for display.

3.5 Graph Visualization

DAVE implemented the visualization of the graphs using Cytoscape.js(Franz et al., 2016), a powerful graph engine, and presented through a web interface. Cytoscape.js enabled us to achieve a fast and interactive experience for physicians.

It allowed us to customize our visualization schema by adding multiple extra features to enhance the user's experience. The user is prompted with the top three graphs along with their corresponding score and can issue a request for three more graphs. Furthermore, we implemented the displaying of the top matching node within a graph. The user is then free to accept or reject the given node. If rejected, the user is taken to the next top scoring node within the given graph. The Accept-Reject feature serves as a potential reinforcement learning model, to provide better results in the future.

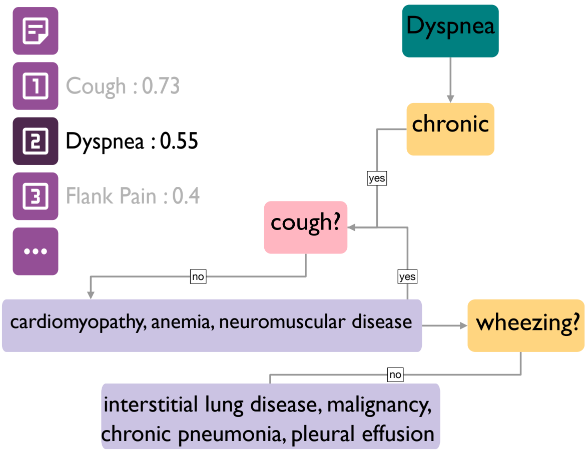

Figure 4: DAVE demo illustration

4 Testing

We tested the NLP model we built for DAVE on real medical notes from AUBMC and the Hariri Medical Center. The notes, in XML format, have the following standardized layout.

- The <text> tag represents the input of the medical doctor,
- The <DxDesc> represents the Diagnosis Description, which is the final diagnosis concluded by the MD.
- These two fields were the target for experimental testing.
- The remaining fields were anonymized if out of context.

We constructed a unit testing program that traverses the given notes and randomly picks out N notes for testing. One constraint forced on picking the target test note was the length of the text field after tokenization. We imposed a minimum length of 10 words, as notes shorter than that would prove inconclusive even to the medical doctor himself since they should be considered as poorly documented. After picking the N notes, the program pairs each text field with its diagnosis description field. Having gathered the test notes, the program stores the test notes for reference in future testing runs and requests the hash map model in question. The test program then compiles the given notes and passes them through the pipeline and outputs the top three results along with their scores.

The output is then dumped into a text file, where we check and calculate the given model and pipeline precision and accuracy. Even though the final diagnosis and considerations are included in the test notes, performance testing is reviewed manually since final note diagnoses are not standardized and show countless expressions for the same diagnosis across the dataset.

5 User Study and Feedback

During its development phases, Dr. Lama Sharafeddine and Dr. Rabiaa Algeboury provided valuable feedback on DAVE, praising its usability and effectiveness in its final version. As medical professionals themselves, they recommended the tool as a learning aid for up-and-coming physicians and a support tool for note completion during patient visits, as well as for the visualization of more complex cases. Additionally, they expressed interest in training DAVE with notes and algorithms in specialty medicine so that medical doctor residents may also benefit from its use.

We conducted two sets of interviews with medical IT experts, specifically the head of IT of AUBMC, Ms. Rola Antoun, and her team.

In the first round, we presented the idea of DAVE and our preliminary implementation plans. They commended our design and helped us brainstorm potential use cases among physicians. They also provided us with insight on what to avoid so that physicians don't refrain from using the software. For example, because it appears complex or requires any additional load like navigating through multiple pages and pressing multiple buttons. Finally, they showcased a few of the more popular software tools and applications among physicians, in hopes of providing further inspiration for DAVE. After completing the project, we met for a second round to demonstrate our work and get usability feedback. They praised the project, were fascinated by the results, and thought it was mature enough to be deployed in the EHR of the medical center for professional use. Furthermore, we had three physicians try out the software and discuss its usability and practicality. We first asked them if this software would benefit them. Two of three said they find the visualization very helpful and beneficiary and would assist the complex task of differential analysis visualization in memory. They also thought that expanding beyond the top three matches is helpful. When asked about user-friendliness, all physicians found the user experience to be simple, fast, and non-tasking. Finally, we received some ideas to make DAVE better. For example, adding specialty diagnostic books to target specific areas of medicine would make the software more useful for more experienced physicians.

6 Results

DAVE's model presented successful and promising results. Considering the difficulty and challenges of NLP in the medical field and case detection algorithms in general, DAVE's model achieved a fairly high accuracy while maintaining a small computation overhead since the results are required to be displayed instantaneously. It also has no problem supporting large text and does not require GPU resources. The best version of the model with all the processing and word augmentations achieved a 74.3% rate of displaying a related diagnosis in the top three scored algorithms. We opted for precision at the best three results to eliminate any chance of bias from headache and fatigue diagnoses which are very general making their symptoms usually present in all diagnoses. This accuracy increased significantly with the addition of PubMed medical corpora as shown in Table 1. The table presents different accuracy results after each step of the pipeline. The prediction rate improved significantly from the processing and the elimination of misleading scores and unnecessary words.

Text Corpus	**Score**
Notes + PubMed	57%
Notes + PubMed + stop words removed (SWR)	68%
Notes + PubMed + SWR + modification of scores (MS)	70%
Notes + PubMed + SWR + MS + correction (CORR)	73%
Notes + PubMed + SWR + MS + CORR + negation	74.3%

Table 1: Summary of the experimental results throughout multiple stages of the DAVE's pipeline

To further understand the performance of DAVE we also conducted a precision@k test where the evaluation metric is predicting a correct diagnosis in one of the top k-matched graphs, with k ranging from 1 to 5. Table 2 presents the results. The results reinforce that displaying the *top three* scored algorithms for the medical professionals is the best practical choice.

To conclude, DAVE achieved accurate results and was deemed user-friendly and very convenient for professional use. It offloads a tedious task from physicians and supports their decision-making during patient visits. It was also regarded as ready for professional use by medical professionals and medical IT experts. Several medical information outlets such as hospitals, and medical insurance companies provide text to diagnosis portals. However, they do not attempt to fully automate the differential diagnosis process. Up to our knowledge no DAVE alternative systems exist so that we can perform a one to one comparison.

K	1	2	3	4	5
Precision (%)	31.25	53.125	74.3	78.125	78.125

Table 2: Precision obtained when a correct diagnosis is in the top k-matched algorithms.

7 Limitations

The main limitation of DAVE is testing it in real time situations. Since the involvement of physicians is required to test DAVE in different note-taking stages, exhaustive realtime testing proved to be difficult. DAVE's objective is to guide and support the physician during the diagnosis process, hence we are interested in the accuracy of the program during different stages of completion of the clinical note. We tested DAVE with volunteer physicians, however, we consider the sample tests as initial and we think that full deployment requires more systematic testing.

8 Conclusion

This paper details a novel model that can suggest diagnostic algorithms to medical professionals based on their clinical notes in real-time. The model uses directed acyclic graphs and semantic similarity metrics to rank diagnostic algorithm graphs taken from digitized medical textbooks and suggests the top three for consideration. The model achieves 74.3% success rate and high acceptance for usability. This model is a significant step forward in improving the accuracy and efficiency of the differential analysis process, which is crucial in making timely and accurate diagnoses and developing effective treatment plans for patients. You can find DAVE online: www.davemr.com. The following is a DAVE system demonstration video. You could also find DAVE's source code on Github

References

B.Sharafeddin. 2020. Cross-document analysis for diagnosis extraction from electronic medical records.

Louise Deléger, Cyril Grouin, and Pierre Zweigenbaum. 2010. Extracting medical information from narrative patient records: the case of medication-related information. *Journal of the American Medical Informatics Association*, 17(5):555–558.

H. Eyre, A. B. Chapman, K. S. Peterson, J. Shi, P. R. Alba, M. M. Jones, T. L. Box, S. L. DuVall, and O. V. Patterson. 2021. Launching into clinical space with medspaCy: a new clinical text processing toolkit in Python. *AMIA Annu Symp Proc*, 2021:438–447.

Elizabeth Ford, John A Carroll, Helen E Smith, Donia Scott, and Jackie A Cassell. 2016. Extracting information from the text of electronic medical records to improve case detection: a systematic review. *Journal of the American Medical Informatics Association*, 23(5):1007–1015.

Max Franz, Christian T Lopes, Gerardo Huck, Yue Dong, Onur Sumer, and Gary D Bader. 2016. Cytoscape. js: a graph theory library for visualisation and analysis. *Bioinformatics*, 32(2):309–311.

Henk Harkema, John N Dowling, Tyler Thornblade, and Wendy W Chapman. 2009. Context: an algorithm for determining negation, experiencer, and temporal status from clinical reports. *Journal of biomedical informatics*, 42(5):839–851.

Mark Henderson, Lawrence M Tierney, and Gerald W Smetana. 2012a. *The patient history: Evidence-based approach*. McGraw Hill Professional.

Mark Richard Henderson, Lawrence M Tierney, and Gerald W Smetana. 2012b. *The patient history: an evidence-based approach to differential diagnosis*. Ph.D. thesis, Univerza v Ljubljani, Medicinska fakulteta.

Peter Kolb. 2008. Disco: A multilingual database of distributionally similar words. *Proceedings of KONVENS-2008, Berlin*, 156.

Saeed Mehrabi, Anand Krishnan, Sunghwan Sohn, Alexandra M Roch, Heidi Schmidt, Joe Kesterson, Chris Beesley, Paul Dexter, C Max Schmidt, Hongfang Liu, et al. 2015. Deepen: A negation detection system for clinical text incorporating dependency relation into negex. *Journal of biomedical informatics*, 54:213–219.

Judea Pearl. 2011. Bayesian networks.

Scott DC Stern, Adam S Cifu, and Diane Altkorn. 2019. *Symptom to Diagnosis An Evidence Based Guide*. McGraw Hill Professional.

Yanshan Wang, Liwei Wang, Majid Rastegar-Mojarad, Sungrim Moon, Feichen Shen, Naveed Afzal, Sijia Liu, Yuqun Zeng, Saeed Mehrabi, Sunghwan Sohn, et al. 2018. Clinical information extraction applications: a literature review. *Journal of biomedical informatics*, 77:34–49.

Li Yujian and Liu Bo. 2007. A normalized levenshtein distance metric. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1091–1095.