Enhancing Diabetes Care through AI-Driven Lie Detection in a Diabetes Support System

Testing the validity of lie detection using an SVM model trained on linguistic cues

Renee van Westerlaak[1]

Supervisor(s): Prof. Catholijn Jonker[1], J.D. Top, MSc[2]

[1]EEMCS, Delft University of Technology, The Netherlands

[2]Bernoulli Institute, University of Groningen, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 22, 2025

Name of the student: Renee van Westerlaak
Final project course: CSE3000 Research Project
Thesis committee: Prof. Catholijn Jonker, J.D. Top, MSc, Dr. Avishek Anand

Abstract

This paper presents a deception-detection module for a diabetes support system, addressing the challenge of unreliable patient self-reporting and ultimately attempting to improve diabetes care. The research is for a system called *CHIP* developed by the Hybrid Intelligence project group and TNO. Linguistic cues, such as motion verbs, negation terms, and exclusive terms were identified through a literature study and encoded using custom dictionaries. Cue detection was implemented using the SpaCy NLP library, which identifies and counts cue occurrences. A stylometric machine learning approach was favored over LLMs for explainability and scientific substantiation. In this research, an SVM model, selected for its alignment with prior research (the Mafiascum experiment), was trained on annotated Mafia game data, using normalized cue frequencies as features for the model. Although the SVM achieved high accuracy on truthful messages (F1 between 0.78–0.84), it performed poorly in detecting deception (F1 between 0.21–0.22), likely because of the high frequency of truthful input compared to deceptive input. The low accuracy, along with the model's domain transferability and performance limitations, suggest further work is needed, particularly with context-specific data and possible integration with LLM-based approaches.

1 Introduction

1.1 Background

Type 2 diabetes is a condition where the body can't properly control blood sugar levels, either because it doesn't make enough insulin or can't use it effectively. Over time, this can lead to serious damage to organs like the heart, eyes, and kidneys. [1]. The implications on a patient's life include having to pay attention to their diet and daily habits, as well as having to monitor their blood glucose.

The onset of type 2 diabetes is in part caused by lifestyle. The rise in blood glucose, insulin levels, and insulin resistance cause several health problems, such as weight gain, inflammation, accelerated aging and comorbidity [2]. A lifestyle intervention focuses on improving the patient's lifestyle habits and reducing insulin dosage needed, therefore slowing down the progression of the disease [2].

In a conference paper on lifestyle (e)support for patients with type 2 diabetes, the lifestyle intervention HINTc (High Intensity Nutrition, Training and coaching) showed several improvements for the patients: Health-Related Quality of Life and mental health both increased and there were notable physiological improvements in the first twelve weeks [2]. An example of such an intervention is one that include a chat-based agent that allows the patient to interact with the system through chat messages.

A challenge for self-care programs is patients' adherence to the program is not guaranteed. Monitoring adherence is essential, as highlighted by Mogre et al. in their review of diabetes self-care programs in low- and middle-income countries [3]. The authors also emphasize the need for finding a proper adherence monitoring tool.

This research surrounds a diabetes support system called *Computer Human Interaction Project (CHIP)*. It is being developed by the Hybrid Intelligence (HI) project group [4] in collaboration with the Nederlandse Organisatie voor Toegepast-Natuurwetenschappelijk Onderzoek (TNO). The part of interest for this particular research is the interaction of a patient with the software agent through chat messages. The goal of the software agent is to gather information from the patient concerning their habits, preferences and progress and to use this to give suggestions that can improve the patient's lifestyle.

As a suggestion to solve the adherence problem identified earlier, this paper addresses the issue of patients providing inaccurate self-reports. In [5], the evaluation of nineteen patients' logging behavior revealed that three-quarters of the participants reported their blood glucose levels as being lower than they actually were. Without the correct information, the support system cannot give accurate suggestions to help the patient. The aim of this research is to find a suitable Machine Learning (ML) model that can be trained to recognize untruthfulness from linguistic aspects of a patient's chat input and turn it into useful information. This could be applied in the chat agent of a diabetes support system, where the result can be processed and used by the system to enhance adherence. Since patients communicate with the system through chat messages, only textual information is available for deception detection. As a result, the focus of this research lies on analyzing the use of linguistic features.

1.2 Research Questions

The main Research Question (**RQ0**) is: *"How can linguistic indicators from a patient's chat message be used to detect deception in a diabetes support system?"*.
It is divided into the following sub-questions;

- **RQ1***: "What are linguistic cues that indicate deception in text?"*, which aims to find linguistic cues from literature, which can be used in textual analysis to detect possible deception. Since a software agent in a diabetes support system is a more specific field of lie detection, the next sub-question aims to find which linguistic cues are actually useful for this research.
- **RQ2***: " What aspects of linguistic lie detection are applicable in the context of a diabetes support system?"*. RQ2 aims to investigate which results from general lie detection research are applicable to *CHIP*.
- **RQ3***: "What ML modeling techniques can be used to extract linguistic cues from user input?"*. This research question attempts to find possible techniques that use an ML model to evaluate the presence of linguistic cues in an input text, that are applicable in the *CHIP* system.

1.3 Linguistic deception cues

For this research, a literature study on detecting deception through linguistic cues was conducted. This study yielded a list of usable linguistic cues (see Table 1) [6; 7; 8; 9; 10;

11; 12; 13], as well as insights into the detection thereof [14; 10] and modern day applications using ML models [15; 16]. An important disclaimer to make at this point is that no one-on-one relation has been determined between deceptive cues and actual deception [13; 17; 10; 14]. Therefore, any findings from the current research are not guaranteed to hold across all situations and individuals. See Discussion (Section 6) for more information on this topic.

1.4 Section overview

This paper describes the current research on how linguistic indicators can be used to detect deception in a diabetes support system. First, the methodology and results of the literature study are discussed in Sections 2 and 3. These results summarize the findings from the literature on the topics of linguistic cues of deception and ML models. To support this, Tables 1, 7, 8, 9 and 10 represent key categories of linguistic cues, including motion verbs, contrastive words, negation words, and tentative language.

Subsequently, Section 4 details the experiment methodology, outlining how ML techniques were employed to detect deceptive cues, and Section 5 presents the results of this implementation.

The paper then addresses limitations in Section 6. Section 7 summarizes key insights and outlines directions for further research. The paper closes with acknowledgments in Section 8 and responsible research considerations in Section 9.

2 Literate study methodology

This Section contains a detailed description of the methodology this research followed to answer ***RQ1*** and ***RQ2***. To obtain a proper theoretical basis, literature studies were conducted on the subjects of linguistic cues to deception and ML modeling techniques.

2.1 Acquiring linguistic cues

The first of two literature studies was conducted on the subject of linguistic lie detection. The goal of this study was to obtain a set of cues that indicate the deceptiveness of a text message. The linguistic cues were gathered from eight different sources [6; 7; 8; 9; 10; 11; 12; 13]. Furthermore, the literature included multiple reviews of these cues and their effectiveness in different scenarios, as well as general remarks on the topic of deception detection, which were taken into consideration.

2.2 Dictionary creation

For four of the resulting cues: motion verbs, exclusive words, tentative words, and negation terms, a list of words contained in each category was created (from now on referred to as a dictionary). The main requirements for these dictionaries were completeness and scientific substantiation.

Four studies included in the literature review, [6; 7; 8; 9], used Linguistic Inquiry and Word Count (LIWC) software for their text analysis. This software has dictionaries for the required categories, based on English dictionaries and common emotion rating scales [18]. Because of its quality and common use, the LIWC software would be the preferred source for this projects' dictionaries as well. However, the software can only be acquired through a license that was not available for the scope of this project.

As an alternative solution, the dictionaries were put together manually by collecting words from scientific sources. The exclusive, tentative and negation categories were formed with words used in the following corpus analyses: [19; 20] for exclusives, [21; 22] for negation terms and [23; 24] for tentative words. For the category motion verbs, such a corpus was not found. Instead, an online dictionary [25] and a study into the English language [26] were used.

Instead of specifically searching for tentative words, the term 'hedges' was used, which yielded more results. The appropriateness of this substitution is supported by [27]. The search term used for finding exclusive words was 'contrastive connectives'.

3 Literature study results

This Section will discuss the results of the literature study. Subsection 3.1 covers findings regarding the effectiveness of linguistic lie detection. Subsection 3.2 gives an overview of the linguistic cues found in the study and Subsection 3.3 briefly covers different deception detection methods that were found in similar experiments. Subsection 3.4 discusses limitations to deception detection found in literature and their connection to the current and similar work, and in Subsection 3.5 a design for a deception-detection module in the *CHIP* system is proposed, combining the findings from the literature research.

3.1 Effectiveness of linguistic cues in lie detection

Two empirical studies into a human's lie detection skills, have found accuracy levels ranging from around chance, when participants had no guidance [10], to 59 to 79% when participants used heuristics [28]. The Oxford Handbook of Lying found that lie-detection accuracy is in the range of 45 to 60% from academic reference compiling research on deception [17].

Lie detection through the use of verbal cues has shown more promising results. For example, it is stated that verbal cues are more consistent than non-verbal cues in [14], an empirical research comparing the accuracy of non-verbal signs with verbal cues. Meta-analyses have demonstrated that verbal, speech-related cues tend to be more reliable indicators of deception than the relatively infrequent and less diagnostic nonverbal cues [10]. For the similarities or differences between spoken and written language when it comes to deception, no literature was found.

In the context of fake news, a machine learning model trained on linguistic features reached up to 99.99% accuracy [29].

3.2 Linguistic cues that indicate deception

The cues in Table 1 have been derived from the literature study into linguistic cues for lie detection. The left column contains a brief description of the linguistic cue. The middle column contains references to papers that contain supporting evidence for the correlation of this cue to deception, and the right column contains references to papers containing contradicting evidence. The cues shown in bold were selected for analysis in this research due to their relevance to the project and to avoid overlap with the work of other researchers in the same research group.

For the cues *fewer words used, more negation terms, fewer first-person pronouns* and *more second- and third-person pronouns*, contradicting evidence was found in [8; 17; 9]. However, since for each of these cues there was more than one study with supporting evidence, the decision has been made to include them nonetheless.

3.3 Similar experiments

Two main approaches for processing input were considered. These approaches were evaluated for both effectiveness and feasibility of implementation in this project to determine the optimal approach.

Input Analysis by Large Language Models

A prior study into the use of Large Language Models (LLMs) for verbal lie detection [15] produced favorable results, reaching up to 79.31% (st. dev. ± 1.3) lie detection accuracy using their LLM. In this study, the FLAN-T5 model was used, fine-tuned on three external datasets containing genuine and fabricated texts from participants [30][31][32]. The model was then used to classify the short narratives as either genuine or fabricated, and its precision was tested. A limitation of this study is that the deceptive texts were not produced in a natural setting. When generating false texts for experimental purposes, the stakes are typically lower than in real-life situations, which may reduce the associated psychological stress. This potential lack of mental strain could lead to differences between fabricated lies and genuine deception. [14].

Despite the results produced in [15], another ML model was chosen for this project. The use of an LLM would not only be resource-intensive, but there was also no suitable dataset available for training in the context of a diabetes support system. Additionally, LLMs have a big disadvantage in terms of explainability. The model's decisions would be difficult to trace and might not be based on the linguistic cues identified in the literature.

Input Analysis Using Stylometry

Instead of using an LLM, the design for the current research uses stylometry to identify linguistic cues of deception [33]. Computational stylometry is described as a method for extracting meta-knowledge, such as authorship, from text. In [15], stylometry is employed as a linguistic deception detection method and compared with the results of their LLM analysis. The stylometric analysis performed in [15] uses LIWC categories (collections of words belonging to a certain category, such as emotional states [18]) to score textual inputs and calculate their correlation with deception.

As this approach makes use of linguistic features, it was adapted for the design chosen in the current research into linguistic deception detection for diabetes support systems.

The Mafiascum experiment

A big source of inspiration for this research has been the Mafiascum experiment [16]. Similarly to the current research, the Mafiascum study employed linguistic cues, which were used as input features for a Support Vector Machine (SVM) model trained on the Mafiascum dataset[1]. The linguistic cues used in the Mafiascum study were all retrieved from a meta-analysis of linguistic cues to deception [34]. The data for the experiment consists of messages from players in online games of Mafia: a game where players get distributed roles of either townspeople or mafia. The goal for mafia is to keep their role hidden while the townspeople try to uncover them. To reach this goal, players with the mafia role need to deceive the other players, which is what made their messages suitable input for the experiment.

The model trained and evaluated in the Mafiascum experiment demonstrated improved performance in detecting deception compared to chance. Specifically, it achieved an average precision of 0.39 for the deceptive class, surpassing the baseline chance level of 0.26. In addition, the experiment found significant correlations between messages from deceptive roles and several features, including six linguistic cues: message length, amount of messages per 24 hours, third-person pronoun ratio, second-person pronoun ratio,"but"-ratio, sentence length, sensory word ratio, and quantifier ratio. Interestingly, only two of the features showing significant correlations: sentence length and third-person pronouns, showed a correlation in the same direction as in the meta-analysis they were retrieved from [34].

The experimental setup of the current research is largely based on the Mafiascum experiment for two main reasons. Firstly, the experiment makes use of linguistic cues, which fits the research questions of the current research. Secondly, unlike most other experiments, the deceptive accounts used as input data are generated in a more natural setting. Higher stakes for successful deception, i.e. winning the game, along with the potential stress of being discovered and a higher cognitive load as a result, may increase the presence of deceptive cues [14; 10].

In contrast, a limitation of the Mafiascum experiment is the potentially misclassified input messages. The messages are labeled according to the user that sent them, meaning all messages from a player with a deceptive role get labeled as deceptive. However, even deceptive players may occasionally speak the truth. Those messages would still be labeled as deceptive, which might impact the accuracy of deception detection model.

ML model comparison

A study into the accuracy of ML models in classifying fake news with the help of linguistic features [29], compared four different models: Extreme Gradient Boosting (XGBoost), SVM, Decision Tree and Naive Bayes. The two best performing models were XGBoost, achieving 99.99% accuracy with a standard deviation of 0.0002 and SVM, achieving 96,33% accuracy with a standard deviation of 0.0390. The best performing training-testing ratios according to the study are 70-30 and 90-10. Both splits achieved an accuracy of 96.73%.

[1] https://bitbucket.org/bopjesvla/thesis/src

Because of these results and the fact that the SVM model was used in two similar researches as well [15; 16], an SVM model was chosen for the current research.

It is important to note that these results were obtained using a dataset consisting of fake and real news articles, and the study incorporated a broader set of features beyond purely linguistic ones. As such, the reported performance metrics cannot be directly generalized to the current research context, which focuses on linguistic cues in a different domain.

3.4 Limitations of Deception Detection

In addition to identifying linguistic cues to deception and reviewing related studies, the literature review conducted in this research revealed several important limitations to the general task of deception detection.

As previously discussed, at least four studies have emphasized that the relationship between linguistic cues and deception is inherently probabilistic rather than deterministic [13; 17; 10; 14]. The behavior of individuals who lie does not typically differ substantially from that of truth-tellers. Emotional states experienced during deception can increase the presence of certain cues [14]; however, when individuals have time to prepare their lies, these cues may be reduced. In the context of a chatbot, the motivation to lie convincingly may be lower than when deceiving real people, and patints often have time to prepare their responses. This suggests that the presence of detectable cues in such interactions may be relatively limited.

A second key limitation is the issue of non-transferability. This concept was highlighted in a study on verbal lie detection using LLMs [15]. In that research, an experiment tested the performance of an LLM in detecting deception across different datasets. While the LLM performed well when tested on the same dataset it was trained on, its classification accuracy dropped when evaluated on a different dataset. Moreover, the study reported inconsistencies in both the magnitude and direction of linguistic cue effects between datasets. For example, the cue “concreteness score of words” was strongly associated with truthful statements in one dataset but correlated with deceptive statements in another.

The combination of a limited number of detectable cues expected in the *CHIP* system’s text messages, and the absence of a representative training dataset, presents challenges to the effectiveness of deception detection methods in this research.

Test data

Test data used in [11], [9], [15] and [16] was reviewed. In the first study, nursing students were asked to lie during interviews [11]. The transcripts of these interviews were then coded using Criteria Based Content Analysis (CBCA) categories to identify correlations between verbal cues and deception. In [9], five studies were conducted in which participants were asked to type out opinions different from their own. The resulting texts were analyzed using LIWC. The third study, [15], also used statements generated by participants of the study. Of these four experiments, only the Mafiascum experiment ([16]) used test data that was not generated for the purpose of the experiment.

As mentioned in Section 3.3, deceptive accounts that are not generated in a natural setting may differ from actual lies. As a result, conclusions drawn from these experiments might not hold for real-life applications.

3.5 Proposal for a lie-detection module in *CHIP*

Figure 1 shows the design of the existing *CHIP* system (colored boxes with a pink background), along with the integrated lie-detection module (outlined boxes with white background).

Existing *CHIP* system

The *CHIP* system is made up of several components, called modules. These modules include the User Interface (UI), which is the part the user interacts with. Input in the UI gets converted by the Text2Triple module, which translates information from the user input into the Triple data structure. A Triple is a data structure consisting of three elements: a subject, a predicate, and an object [35]. These Triples are stored in the next module: the Knowledge Graph (KG), a structure that represents knowledge in a form that can be utilized by the application [36]. Finally, the *CHIP* system includes the Reasoner and Response Generator, which generate the next system response based on the current knowledge in the KG.

Lie-detection module

The proposed lie-detection module gets as input a single text message from the user. It does not save state between messages, nor does it keep information extracted from anything else. The flow of the proposed module is as follows: the user’s text message is first entered via the UI, then passed as a string into the lie-detection module. This module analyzes the message for linguistic cues. These cues are classified by a trained ML model, and the result is converted into a Triple, which is then added to the KG for further use by the system.

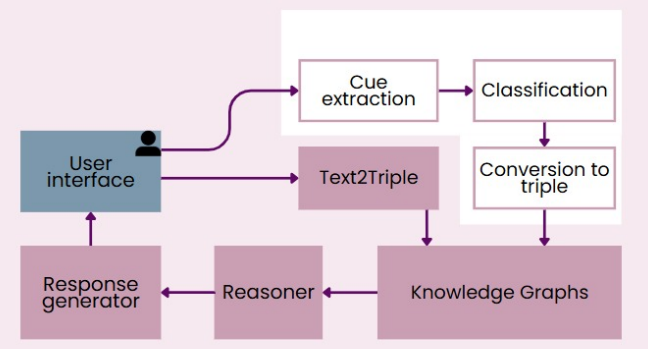

Figure 1: The modules of the *CHIP* system (colored boxes), with lie-detection module (outlined boxes on white background). The user interacts only with the blue module, not with the pink modules.

Linguistic cue detection is performed using the SpaCy library [37], a Python Natural Language Processing (NLP) library that uses its own Neural Networks. SpaCy can identify and count occurrences of deceptive linguistic cues in a given text.

As discussed in Section 3.3, the SVM was chosen for this research, based on high performance in a research comparing models [29] and similarities with the Mafiascum experiment [16]. The SVM model is first trained on data from the

Mafiascum dataset [16]. After this training phase, new input messages can be classified by the model. The previously extracted cues are saved as a feature vector, with each feature being the occurrence count of a cue divided by the total words in the message for normalization. Running the model on this feature vector then results in a prediction: truthful or deceptive. This prediction can then be forwarded into the next module in *CHIP*.

4 Experiment methodology

This Section outlines the methodology for the linguistic deception detection experiment. The experiment evaluates a proposed lie-detection module. The module is tested on an existing dataset containing texts annotated as truthful or deceptive [16]. A comparison with the Mafiascum experiment is made in Subsection 4.1 and Subsection 4.2 covers the setup of the experiment.

4.1 Methodological Differences from the Mafiascum experiment

All of the linguistic cues used in this study are also evaluated in the Mafiascum experiment. However, there are notable differences in the processing certain cues. In the Mafiascum experiment, the category of negation terms is reduced to the single term “not”, while the exclusive terms are represented solely by the words “but” and “or”. In the current research, these categories are extended through the use of dictionaries. Additionally, the Mafiascum study includes several linguistic cues not considered in the current research.

All of the linguistic cues used in the Mafiascum experiment were derived from a single meta-analysis on the effectiveness of computer-assisted lie detection [34]. In contrast, the cues employed in the current research were sourced from eight different empirical studies, providing a broader and more diverse foundation.

Regarding the experimental implementation, only the preprocessing component was directly adapted from the Mafiascum repository. All other code was developed specifically for the current research, using the Mafiascum experiment as a guideline. Finally, in the Mafiascum experiment, the training and evaluation of the SVM model were conducted using a 20-fold stratified shuffle split, whereas the current study employed a single train-test split for model training and evaluation.

4.2 Experimental Setup

This Section describes how the SVM model was trained and evaluated. The repository can be found on GitHub[2].

To evaluate the effectiveness of the linguistic cues identified in this study, an SVM model was trained and tested on the Mafiascum dataset [16]. The implementation used was based on the original Mafiascum repository[3] from the experiment described in Subsection 3.3, which was updated for compatibility with current versions of its dependencies. The feature extraction process was modified to reflect the specific linguistic cues identified in this research. The model was then trained on the dataset using the updated features and subsequently tested to evaluate its accuracy.

The code for preprocessing the documents was copied from the Mafiascum codebase. Preprocessing consisted of removing messages that were not useful for the classification task, such as admin messages or the discussion after a game. Games with fewer than 50 words were excluded, and all messages from a single player within a game were aggregated into one document.

The training and testing phases were conducted using the Scikit Learn Library [38]. This library offers tools for classification by an SVM model that can be used in Python applications, such as the *CHIP* system. The model was trained and tested on a subset of the Mafiascum dataset as well as the whole set, as described in the next section.

Datasets

The entire Mafiascum dataset consists of three subsets of games; One set of large games[4] and two sets of mini games[5][6]. Large games were played with more experienced players and typically lasted longer. In Table 2, a summary of these sets can be found, containing the number of games in each subset, the number of documents they resulted in after preprocessing and the so-called scum ratio. The scum ratio refers to the amount of players that were assigned the role “scum” divided by the total amount of players in a game. The documents from players with this role were seen as deceptive in this experiment and the documents from players with the “town” role were seen as truthful.

Both sets were trained with two training and testing dataset splits: 70-30 and 90-10 (referring to the percentage of data in the training set compared to the test set, respectively). These ratios were obtained from the aforementioned experiment into fake news classification [29].

5 Experiment results

After the training and testing phase, a classification report was generated using Scikit Learn [38]. The classification report presents the precision $\left(\frac{true\ positives}{predicted\ positives}\right)$, recall $\left(\frac{true\ positives}{actual\ positives}\right)$ and F1-score (harmonic mean of precision and recall) for both truthful and deceptive classes. It also gives the (macro and weighted) accuracy.

Tables 3, 4, 5 and 6 contain the results for both of the datasets and both of the training and testing dataset splits.

For the deceptive class, the model achieved an F1-score of 0.22 in the full dataset case with split 70-30 and an F1-score of 0.21 in the other three cases. These results suggest that the trained SVM did not perform well in recognizing deceptive players through their messages.

In contrast, the F1-score for the truthful class was higher (0.84 and 0.78). This indicates the model performed well in

[2] https://github.com/reneelanatove/bep_svm_pipeline

[3] https://bitbucket.org/bopjesvla/thesis/src

[4] large-normal.json

[5] mini-normal1.json

[6] mini-normal2.json

Cue	Count	Supporting Citations	Contradictions
Fewer words used	3	[10] [6] [7]	[8]
More sentences, fewer distinct words	1	[6]	
Fewer exclusive words (but, except)	2	[9] [6]	
Fewer tentative words (may, perhaps)	1	[6]	
More negation terms (no, never)	2	[9] [6]	[8]
More negative emotion words	4	[9] [6] [10]	
Fewer first-person pronouns	4	[12] [8] [9] [6]	[13]
More second-person pronouns	1	[6]	[9]
More third-person pronouns	3	[12] [7] [6]	[9]
More motion verbs	2	[12] [6]	
Fewer insight/cognitive words	2	[6] [13]	[10]
Speech errors and disfluencies	1	[10]	[6]
Indirect/ritualized speech	1	[10]	
Self-deprecation	1	[10]	
Fewer sensory details	1	[10]	
Fewer details	1	[14]	
Fewer causation words (only relevant in omission lies)	1	[7]	

Table 1: Overview of linguistic deception cues, including supporting and contradicting evidence. The 'Count' column indicates the number of studies with supporting findings, followed by columns containing citations for both supporting and contradicting studies

Subset	No. of games	No. of docs	Scum ratio
Large	39	889	0.246
Mini 1	210	3173	0.232
Mini 2	397	5077	0.229

Table 2: Subsets of Mafiascum game messages used in the experiment, with the number of games in each subset, number of documents they produced and ratio of scum (deceptive) and town (truthful) roles

Class	Precision	Recall	F1-Score	Support
Truthful (0)	0.77	0.93	0.84	201
Deceptive (1)	0.40	0.15	0.22	66
Accuracy			0.73	267
Macro avg	0.58	0.54	0.53	267
Weighted avg	0.68	0.73	0.69	267

Table 3: Classification report for SVM trained on the large games subset with training and testing dataset split 70:30

Class	Precision	Recall	F1-Score	Support
Truthful (0)	0.77	0.79	0.78	2232
Deceptive (1)	0.22	0.20	0.21	671
Accuracy			0.65	2903
Macro avg	0.50	0.50	0.50	2903
Weighted avg	0.64	0.65	0.65	2903

Table 4: Classification report for SVM trained on full dataset with training and testing dataset split 70:30

Class	Precision	Recall	F1-Score	Support
Truthful (0)	0.75	0.82	0.79	67
Deceptive (1)	0.25	0.18	0.21	22
Accuracy			0.66	89
Macro avg	0.50	0.50	0.50	89
Weighted avg	0.63	0.66	0.64	89

Table 5: Classification report for SVM trained on the large games subset with training and testing dataset split 90:10

Class	Precision	Recall	F1-Score	Support
Truthful (0)	0.77	0.80	0.78	744
Deceptive (1)	0.23	0.20	0.21	224
Accuracy			0.66	968
Macro avg	0.50	0.50	0.50	968
Weighted avg	0.64	0.66	0.65	968

Table 6: Classification report for SVM trained on full dataset with training and testing dataset split 90:10

recognizing truthful messages, which were the majority of the messages (around 80%-85%).

The highest F1-score for the weighted average was achieved after training the SVM on only the large games with a split of 70-30 (Table 3). In this case, the model's precision in predicting the deceptive class was also relatively high (0.40 compared to 0.22-0.25 in the other three cases). However, this is contrasted by its low recall score (0.15), indicating that although 40% of the times the model predicted a deceptive class, it was correct, it only predicted a deceptive class for 15% of the deceptive class documents.

6 Discussion

This section reflects on the broader implications and limitations of the current research. During the literature study and experiment, several constraints were encountered, both in the method itself and in its application to the *CHIP* system and the healthcare context.

6.1 Limitations to linguistic lie detection

A notable finding from the literature study into linguistic cues for deception detection is that there are many limitations to this method (See Subsection 3.4).

Linguistic cues to deception are not universal; they can vary across languages and cultures [13], between individuals [17], and even among different types of lies [14]. This variability implies that no fixed set of cues can reliably ensure accurate detection across all people or all forms of deception.

Furthermore, for the pronoun-related cues used in the current research, empirical studies have shown conflicting evidence [17; 9]. Due to the lack of cues uniquely associated with deception [14], cues with contradictory findings were still included in this study. However, it must be noted that the correlation between these linguistic cues and actual deception has been challenged.

6.2 Limitations to lie detection in the *CHIP* system

In addition to the general challenges of linguistic lie detection, the implementation within the *CHIP* system faces specific limitations.

A combined approach using both verbal and non-verbal cues is estimated to yield the most accurate results [14]. However, the *CHIP* system only receives a user's text input, limiting the scope of available information for lie detection.

Moreover, this research evaluates individual text messages rather than full conversations. This decision was made partly to keep the project within scope, and partly to avoid overlapping with other studies within the same research group that focused on conversational lie detection and conflict resolution within the Knowledge Graph (KG). As a result, inconsistencies across a patient's statements which are potential indicators of deception [14] are not captured.

Finally, the classifications made by the lie-detection module within *CHIP* cannot be externally verified. The system does not offer feedback on whether a user's input was actually truthful. Consequently, the real-world performance of the module cannot be fully evaluated without additional validation once deployed.

6.3 Limitations to lie detection in a diabetes support system

These limitations are further compounded by challenges specific to the healthcare context in which the system operates.

Currently, no suitable training data exists that accurately mirrors the type of messages processed by *CHIP*. As a result, the model must be trained on unrelated data.

Cues to deception differ greatly between contexts [14]. This means that a model trained on the Mafiascum dataset [16] may not perform as effectively when classifying deception in messages about diabetes. This issue was also observed in research on lie detection using LLMs [15], where a model trained on one dataset failed to deliver accurate results when applied to a dataset with different statement types highlighting the problem of poor cross-domain transferability.

7 Conclusions and Future Work

Based on the results presented in Section 5, it can be concluded that the SVM model proposed in this study performs poorly in detecting deceptive messages. This conclusion is supported by consistently low F1-scores for the deceptive class, ranging from 0.21 to 0.22 across all tested scenarios, including both the subset of 39 games and the full dataset of 646 games, with training-to-testing split ratios of 70:30 and 90:10. These results indicate that the model struggles to accurately identify deceptive messages.

In contrast, the model achieved relatively high F1-scores for the truthful class, ranging from a score of 0.78 to a score of 0.84. However, this difference may be attributed to the class imbalance in the dataset, where approximately 80% of the messages originate from truthful players, and only 20% from deceptive ones. As a result, the model is more likely to classify messages as truthful, which contributes to its higher performance on that class.

To understand these results, they can be compared with the performance of a baseline model that reflects the underlying class distribution. Take a model that randomly predicts the truthful class 80% of the time and the deceptive class 20% of the time, close to the actual class distribution. Since the predictions are random, the model is expected to correctly classify 80% of the truthful instances and 20% of the deceptive instances. For the truthful class, the precision and recall (See Section 5) would both be approximately 0.80, resulting in an F1-score, or harmonic mean, of 0.80. For the deceptive class, both the precision and recall would be around 0.20, yielding an F1-score of 0.20. These values reflect the performance of a model that has no real predictive power, yet mimics the base rates in the data. Comparing this to the results of the current model as mentioned above, indicates that it performs only slightly better, or potentially not better at all, than a naive model that guesses based on class proportions.

7.1 Combination with other work

To increase effectivity of the diabetes support system, this work could be combined with work from other researchers on the same subject.

Conversation summaries generated by an LLM

One of the complimenting studies, is one on generating context files for the patients' chat conversations using LLMs. This research has overlap with the current research and has features that can be used to extend the proposed lie-detection module's functionality. The overlap is in the area of linguistic analysis. Both the current research and the research into context files analyze word usage of patients. Some cues derived from the current literature study, such as short responses and use of hedges, are used as guidelines for the LLM's summary generation.

A suggestion would be to include both the summary generation and linguistic deception detection to evaluate patients' messages for possible deception.

Response generation for reducing deceptive tendencies

Another complimenting study is on the linguistic framings of an AI agent's chat messages to reduce patients' tendencies to deceive diabetes support systems. Detecting deception alone will not do anything to solve the problem central to this research. Instead, creating an environment where patients feel comfortable to be honest will allow the system to give optimal recommendations and might ultimately improve the diabetes care provided by the system.

7.2 Recommendations for future work

The results of the current experiment did not indicate a promising solution for detecting deception in the *CHIP* system. A more promising result was achieved in a study on the use of LLMs for lie detection [15]. Therefore, a recommendation for future studies is to combine the linguistic cue analysis from the current study with LLM-based designs to obtain more accurate analyses.

An alternative suggestion is to evaluate the accuracy of the proposed SVM model when trained on more datasets. Three options for this are the corpora used in the aforementioned study [23; 24; 39]. Ideally, data from the specific diabetes support system is gathered on which the SVM is trained. As concluded in the literature study results, deception detection may differ from context to context, so training an SVM on data that is similar to the data that needs to be classified in the future might improve the performance of the model.

For more accurate and extensive cue categories, LIWC should be used instead of the dictionaries in this research [18]. LIWC's collections have been validated by scientific labs and the software is used in over twenty-thousand scientific articles. Furthermore, combining cues instead of looking at the effects of each cue by itself could lead to more accurate research [14].

Conversion to Triples

Initially, converting the results of the lie-detection module into the Triple data structure (see Section 3.5) was part of the scope of this research. However, since a functional lie-detection model could not be developed, the conversion was not implemented. Nevertheless, several ideas have been considered for potential future implementation.

The first consideration is how to represent the results of the deception detection process. In this research, an SVM model classified input as either truthful or deceptive. This outcome could be mapped to a single integer in the resulting Triple (e.g., 0 for truthful and 1 for deceptive). A disadvantage of this approach is the lack of nuance in the representation. For example, a message that is highly likely to be deceptive and one that is only slightly above the decision threshold would both be labeled as deceptive, thereby losing valuable information about the degree of deception.

An alternative approach involves calculating a probability estimate and incorporating that into the Triple. This method allows for a more granular and informative representation of the detection results.

The second consideration concerns which components to include in the Triple. One option is to use the message ID as the subject, a predicate such as ”has deceptive probability,” and the deception detection result as the object. The main drawback of this approach is that the result is associated with the message itself, rather than with the specific information contained within the message. A more informative alternative would be to use the relevant Triples (representing the extracted information from the message) as the subject, maintaining the same predicate and object structure as previously suggested.

8 Acknowledgments

I would like to express my gratitude to my responsible professor, Prof. Catholijn M. Jonker, and my supervisor, J.D. Top, MSc, for their support and guidance throughout this project. I am also thankful to Prof. Rineke Verbrugge and Prof. Harmen de Weerd for their valuable feedback and advice.

Furthermore, I would like to thank TNO and the Hybrid Intelligence project group for providing the opportunity to collaborate on the CHIP project, which has been a valuable part of this research experience.

9 Responsible Research

When considering the ethical implications of this research, several important issues arise. The most prominent issue is the potential consequences for patients whose text data is analyzed and flagged for deception. This could lead to a loss of authority, when patients cannot give false information to protect their privacy. The investigation into this topic should be done by the HI project group in consultation with doctors and patients when reaching further stages of the project and will not be further discussed in this paper.

Another important note is that the lie-detection module will possibly encounter private information from patients, such as their daily habits or glucose levels. Therefore, the information must be protected and communication should be secure.

This research aims to produce scientifically correct and reproducible results. To ensure the soundness of information presented in this paper, sources were selected with care. The origin of each source was verified and only scientific papers were consulted. On top of that, sources were compared to find overlapping and contrasting information with the goal of analyzing the validity of the information.

For reproducibility, each step of the of the experiment is carefully documented. Decisions made and alternatives considered are also included, to allow readers to make their own decisions. Since the developed code for this research as well as the code consulted can be found, readers can rerun the current experiment and adapt it for their own project.

9.1 Ethical implications of deception detection

When deciding to implement deception detection, it is important to consider how the results should be treated. There are going to be consequences when a message is flagged as deceptive, for example sharing the information with a doctor or changing behavior toward the patient. Not only should the effects of these consequences be considered in general, but the effects of these consequences in case of a false positive (a truthful message being incorrectly flagged as deceptive) require extra attention. A patient might feel wrongfully treated or mistrusted if their truthful accounts are flagged as deceptive and this could lead to a negative perception of the system.

9.2 Usage of AI for this research

While conducting this research, Generative AI model ChatGPT 4o-mini was used to assist in the writing process. This was done mainly by prompting the LLM to find grammatical and spelling errors and suggest improvements for readability. These suggestions were then taken into consideration and applied to the text if they did not change its meaning and had a positive effect on the quality of the text.

On top of that, ChatGPT 4o-mini was used for converting the lists of cues and sources into the proper overleaf format for the tables in this paper.

Lastly, portions of the Mafiascum code were difficult to interpret due to insufficient documentation. To facilitate adaptation for this research, assistance was sought from ChatGPT 4o-mini to aid in understanding the code.

The prompts used can be found in Appendix A.

When using ChatGPT 4o-mini, the risks of working with an LLM were taken into careful consideration. Each response was evaluated for its accuracy and potential biases from the LLM's training data [40] were taken into account. Information presented by the LLM was always reasoned about and compared with scientific data, and code suggestions were analyzed.

A ChatGPT Prompts

For the following prompts, ChatGPT 4o-mini was accessed between 29-04-2025 and 15-06-2025.

Prompt for text refactoring

"For this text, list and suggest changes for: - grammatical errors - spelling errors - sentences with bad readability - informal language only include problematic parts and leave as much as possibile of the original text"

Prompt for table creation

"For this list of cues and their respective sources, please make an overleaf table in the following format: [format]. *The table should fit in a two-column project and overflow into two columns if it does not fit on the page. If a cue is found in two sources, please add both of the sources in the second column."* [list of cues]

Prompt for Mafiascum code explanation

"Explain the following code and what each part does"

B Dictionaries

Word	Source(s)
amble	[25; 26]
ascend	[25]
arrive	[25]
bolt	[25; 26]
bob	[25]
book	[25]
bound	[25; 26]
canter	[25; 26]
cavort	[25; 26]
charge	[25; 26]
climb	[25; 26]
creep	[25; 26]
dash	[25; 26]
dance	[25]
depart	[25]
descend	[25]
enter	[25]
escape	[25]
exit	[25]
flee	[25; 26]
float	[25; 26]
frolic	[25; 26]
gallop	[25; 26]
gimp	[25]
hike	[25; 26]
hobble	[25; 26]
hop	[25; 26]
hurdle	[25]
jog	[25; 26]
jump	[25; 26]
leap	[25; 26]
leave	[25]
limp	[25; 26]
lolligag	[25]
lope	[25; 26]
lumber	[25; 26]
lurch	[25; 26]
march	[25; 26]
meander	[25; 26]
pad	[25; 26]
pounce	[25]
prance	[25; 26]
scoot	[25; 26]
scour	[25]
scurry	[25; 26]
scuttle	[25; 26]
shlep	[25]
shuffle	[25; 26]
slink	[25; 26]
slog	[25; 26]
sneak	[25; 26]
speed	[25; 26]
sprint	[25; 26]
stagger	[25; 26]
stalk	[25]
steal	[25; 26]
step	[25]
stomp	[25; 26]
storm	[25; 26]
straggle	[25; 26]
stroll	[25; 26]
strut	[25; 26]
stumble	[25; 26]
swagger	[25; 26]
swim	[25; 26]
tiptoe	[25; 26]
traipse	[25; 26]
trot	[25; 26]
trudge	[25; 26]
waddle	[25; 26]
wade	[25; 26]
walk	[25; 26]
wander	[25; 26]
zip	[25]
zoom	[25; 26]
zigzag	[25; 26]

Table 7: Motion verbs and the research(es) they were extracted from

Word	**Source(s)**
however	[19; 20]
instead	[19]
then	[19]
in contrast	[19; 20]
nevertheless	[19; 20]
rather than	[19]
on the other hand	[19; 20]
instead of	[19; 20]
on the contrary	[19; 20]
although	[20]
though	[20]
although	[20]
but	[20]
contrary to	[20]
conversely	[20]
despite	[20]
in comparison	[20]
in spite	[20]
nonetheless	[20]
rather than	[20]
still	[20]
whereas	[20]
yet	[20]

Table 8: Contrastive words and the research(es) they were extracted from

Word	**Source(s)**
never	[21; 22]
neither	[21]
nobody	[21]
no	[21; 22]
none	[21]
nor	[21]
nothing	[21]
nowhere	[21]
not	[22]
don’t	[22]
doesn’t	[22]
without	[22]
didn’t	[22]
isn’t	[22]
can’t	[22]
wasn’t	[22]

Table 9: Negation terms and the research(es) they were extracted from

Word	Source(s)
can	[23; 24]
can’t	[23]
could	[23; 24]
might	[23; 24]
may	[23; 24]
will	[23]
would	[23; 24]
should	[24]
assume	[23; 39]
assumption	[23; 24]
believe	[23]
belief	[23]
believed	[24]
estimate	[23]
expect	[23]
expecting	[23]
expected	[23]
expectation	[23]
feel	[23]
feeling	[23]
guess	[23; 39]
guessing	[23]
think	[23; 39]
thinking	[23]
understand	[23]
understanding	[23]
about	[23; 24; 39]
almost	[23; 39]
approximately	[23; 39]
around	[23; 24]
a little bit	[23]
at least	[23]
basically	[23]
broadly	[23]
comparatively	[23]
essentially	[23]
fairly	[23]
generally	[23; 24]
in general	[23]
kind of	[23; 39]
largely	[23]
likely	[23; 24]
mainly	[23]
maybe	[23; 39]
mostly	[23]
more or less	[23]
nearly	[23; 39]
normally	[23]
often	[23; 24]
on the whole	[23]
perhaps	[23; 39]
possible	[23; 24]
possibly	[23]
probably	[23; 24; 39]
quite	[23; 24; 39]
rather	[23; 24]
relative	[23]
relatively	[23; 24]
roughly	[23; 39]
seem	[23; 24]
seems	[23]
seemed	[23]
slight	[23]
slightly	[23]
some	[23; 24]
sometimes	[23]
somewhat	[23; 39]
typically	[23]
usually	[23]
so	[23]
well	[23]
I mean	[23; 39]
sort of	[23; 39]
you know	[23]
as you well know	[23]
to my knowledge	[23]
from my perspective	[23]
from our perspective	[23]
in my view	[23]
in our view	[23]
to...extent	[23]
certain amount	[23]
certain level	[23]
suggest	[24]
indicate	[24]
appear	[24]
tend	[24]
argue	[24]
seen as	[24]
perceive	[24]
predict	[24]
(not) always	[24]
frequently	[24]
most	[24]
consistent	[24]
seldom	[24]
plausible	[24]
rare	[24]
questionable	[24]
probable	[24]
possibility	[24]
tendency	[24]
probability	[24]
implication	[24]
prediction	[24]
doubt	[24]
(in) theory	[24]
contention	[24]
conjecture	[24]
entirely	[39]
according to	[39]
presumably	[39]
say that	[39]
as far as I	[39]

Table 10: Tentative words and the research(es) they were extracted from

References

[1] A. M. Egan and S. F. Dinneen, "What is diabetes?," *Medicine*, vol. 47, no. 1, pp. 1–4, 2019. https://doi.org/10.1016/j.mpmed.2018.10.002.

[2] L. P. A. Simons, H. Pijl, J. Verhoef, H. J. Lamb, B. van Ommen, B. Gerritsen, M. B. Bizino, M. Snel, R. Feenstra, and C. M. Jonker, "Intensive lifestyle (e)support to reverse diabetes-2," in *Proceedings of the 29th Bled eConference: Digital Economy* (B. Gregor and S. Stanovnik, eds.), vol. 29, (Bled, Slovenia), pp. 339–352, AIS Electronic Library, 2016.

[3] V. Mogre, N. A. Johnson, F. Tzelepis, J. E. Shaw, and C. Paul, "A systematic review of adherence to diabetes self-care behaviours: Evidence from low- and middle-income countries," *Journal of Advanced Nursing*, vol. 75, no. 12, pp. 3374–3389, 2019.

[4] B. J. W. Dudzik, J. S. van der Waa, P.-Y. Chen, R. Dobbe, Íñigo M.D.R. de Troya, R. M. Bakker, M. H. T. de Boer, Q. T. Smit, D. Dell'Anna, E. Erdogan, P. Yolum, S. Wang, S. B. Santamaria, L. Krause, and B. A. Kamphorst, "Viewpoint: Hybrid intelligence supports application development for diabetes lifestyle management," *Journal of Artificial Intelligence Research*, vol. 80, pp. 919–929, 2024. https://doi.org/10.1613/jair.1.15916.

[5] R. S. Mazze, H. Shamoon, R. Pasmantier, D. Lucido, J. Murphy, K. Hartmann, V. Kuykendall, and W. Lopatin, "Reliability of blood glucose monitoring by patients with diabetes mellitus," *The American Journal of Medicine*, vol. 77, no. 2, pp. 211–217, 1984.

[6] V. Hauch, I. Blandón-Gitlin, J. Masip, and S. L. Sporer, "Linguistic cues to deception assessed by computer programs: A meta-analysis," in *Proceedings of the Workshop on Computational Approaches to Deception Detection* (E. Fitzpatrick, J. Bachenko, and T. Fornaciari, eds.), pp. 1–4, Association for Computational Linguistics, 2012.

[7] L. M. V. Swol, M. T. Braun, and D. M. and, "Evidence for the pinocchio effect: Linguistic differences between lies, deception by omissions, and truths," *Discourse Processes*, vol. 49, no. 2, pp. 79–106, 2012.

[8] J. T. Hancock, L. E. Curry, S. Goorha, and M. W. and, "On lying and being lied to: A linguistic analysis of deception in computer-mediated communication," *Discourse Processes*, vol. 45, no. 1, pp. 1–23, 2007.

[9] M. L. Newman, J. W. Pennebaker, D. S. Berry, and J. M. Richards, "Lying words: Predicting deception from linguistic styles," *Personality and Social Psychology Bulletin*, vol. 29, no. 5, pp. 665–675, 2003. https://doi.org/10.1177/0146167203029005010.

[10] B. M. DePaulo, J. J. Lindsay, B. E. Malone, L. Muhlenbruck, K. Charlton, and H. Cooper, "Cues to deception," *Psychological Bulletin*, vol. 129, no. 1, pp. 74–118, 2003. https://doi.org/10.1037/0033-2909.129.1.74.

[11] A. Vrij, K. Edward, K. Roberts, and R. Bull, "Detecting deceit via analysis of verbal and nonverbal behavior," *Journal of Nonverbal Behavior*, vol. 24, pp. 239–263, 2000. https://doi.org/10.1023/A:1006610329284.

[12] Ö. Yeter, B. Kooi, H. de Weerd, R. Verbrugge, and P. Hendriks, "Semantic leakage enables lie detection, but first-person pronouns and verbosity can get in the way of detection," in *Proceedings of the Annual Meeting of the Cognitive Science Society*, vol. 46, pp. 2768–2775, The Cognitive Science Society, 2024.

[13] K. J. Hardin, *The Oxford Handbook of Lying*, ch. 4. Linguistic Approaches to Lying and Deception. Oxford University Press, 2018.

[14] A. Vrij, P. A. Granhag, and S. Porter, "Pitfalls and opportunities in nonverbal and verbal lie detection," *Psychological Science in the Public Interest*, vol. 11, no. 3, pp. 89–121, 2010. https://doi.org/10.1177/1529100610390861.

[15] R. Loconte, R. Russo, P. Capuozzo, P. Pietrini, and G. Sartori, "Verbal lie detection using large language models," *Scientific Reports*, vol. 13, no. 1, p. 22849, 2023. https://doi.org/10.1038/s41598-023-50214-0.

[16] B. de Ruiter and G. Kachergis, "The mafiascum dataset: A large text corpus for deception detection," 2019.

[17] S. Mann, *The Oxford Handbook of Lying*, ch. 31. Lying and Lie Detection. Oxford University Press, 2018.

[18] M. Francis and R. J. Booth, "Linguistic inquiry and word count," *Southern Methodist University: Dallas, TX, USA*, 1993.

[19] C. Nan, "A corpus-based study on the application of connectives in Chinese college students' English writing," in *Proceedings of the 2021 International Conference on Social Sciences and Big Data Application (ICSSBDA 2021)*, pp. 69–75, Atlantis Press, 2021. https://doi.org/10.2991/assehr.k.211216.014.

[20] B. Fraser, "What are discourse markers?," *Journal of Pragmatics*, vol. 31, no. 7, pp. 931–952, 1999.

[21] A. Kaufmann, "Negation and prosody in British English: a study based on the London–Lund corpus," *Journal of Pragmatics*, vol. 34, no. 10, pp. 1473–1494, 2002. https://doi.org/10.1016/S0378-2166(02)00074-7.

[22] N. Cruz Diaz, N. Konstantinova, S. Castilho, M. Maña, M. Taboada, and R. Mitkov, "A review corpus annotated for negation, speculation and their scope," 2012.

[23] R. Fu and J. Tan, "Hedges in interpreted and non-interpreted english: A cross-modal, corpus-based study," *Interpreting and Society*, vol. 4, no. 1, pp. 44–66, 2024. https://doi.org/10.1177/27523810231224149.

[24] S.-P. Wang, "Corpus research on hedges in linguistics and efl journal papers," *International Journal of Education*, vol. 9, p. 44, 2016. https://doi.org/0.17509/ije.v9i1.3717.

[25] J. Lawler, "Lexical frequency and function: Motion verbs." https://websites.umich.edu/~jlawler/words/,

2003. University of Michigan, [Online; accessed 2025-05-22].

[26] I. N. i Ferrando, "Capítulo 6.1. categorías léxico-gramaticales con significado textual: organizadores del discurso." http://elies.rediris.es/elies11/cap61.htm, 2002. [Online; accessed 2025-05-22].

[27] K. H. and, "Hedges, boosters and lexical invisibility: Noticing modifiers in academic texts," *Language Awareness*, vol. 9, no. 4, pp. 179–197, 2000.

[28] B. Verschuere, C.-C. Lin, S. Huismann, B. Kleinberg, M. Willemse, E. Mei, T. Goor, L. Loewy, O. Appiah, and E. Meijer, "The use-the-best heuristic facilitates deception detection," *Nature Human Behaviour*, vol. 7, p. 718–728, 2023. https://doi.org/10.1038/s41562-023-01556-2.

[29] E. Puraivan, R. Venegas, and F. Riquelme, "An empiric validation of linguistic features in machine learning models for fake news detection," *Data Knowledge Engineering*, vol. 147, p. 102207, 2023. https://doi.org/10.1016/j.datak.2023.102207.

[30] M. Sap, A. Jafarpour, Y. Choi, N. A. Smith, J. W. Pennebaker, and E. Horvitz, "Quantifying the narrative flow of imagined versus autobiographical stories," vol. 119, p. e2211715119, 2022. https://doi.org/10.1073/pnas.2211715119.

[31] P. Capuozzo, I. Lauriola, C. Strapparava, F. Aiolli, and G. Sartori, "Decop: A multilingual and multi-domain corpus for detecting deception in typed text," in *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020* (N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, and S. Piperidis, eds.), pp. 1423–1430, European Language Resources Association, 2020.

[32] B. Kleinberg and B. Verschuere, "How humans impair automated deception detection performance," *Acta Psychologica*, vol. 213, p. 103250, 2021. https://doi.org/10.1016/j.actpsy.2020.103250.

[33] W. Daelemans, "Explanation in computational stylometry," in *Computational Linguistics and Intelligent Text Processing - 14th International Conference, CICLing 2013, Samos, Greece, March 24-30, 2013, Proceedings, Part II* (A. F. Gelbukh, ed.), vol. 7817 of *Lecture Notes in Computer Science*, pp. 451–462, Springer, 2013. https://doi.org/10.1007/978-3-642-37256-8_37.

[34] V. Hauch, I. Blandón-Gitlin, J. Masip, and S. L. Sporer, "Are computers effective lie detectors? a meta-analysis of linguistic cues to deception," *Personality and Social Psychology Review*, vol. 19, no. 4, pp. 307–342, 2015. https://doi.org/10.1177/1088868314556539.

[35] R. Cyganiak, D. Wood, and M. Lanthaler, "Rdf 1.1 concepts and abstract syntax." https://www.w3.org/TR/rdf11-concepts/, 2014. [Online; accessed 2025-05-27].

[36] X. Chen, S. Jia, and Y. Xiang, "A review: Knowledge reasoning over knowledge graph," *Expert Systems with Applications*, vol. 141, p. 112948, 2020. https://doi.org/10.1016/j.eswa.2019.112948.

[37] M. Honnibal, I. Montani, S. V. Landeghem, and A. Boyd, "spaCy: Industrial-strength Natural Language Processing in Python." https://spacy.io, 2020. Version 2.3+, [Online; accessed 2025-05-27].

[38] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[39] E. Leláková and D. Praženicová, "Exploring hedging in spoken discourse: Insights from corpus analysis," *Arab World English Journal*, vol. 15, no. 3, pp. 270–282, 2024. https://dx.doi.org/10.24093/awej/vol15no3.16.

[40] C. Head, P. Jasper, M. McConnachie, L. Raftree, and G. Higdon, "Large language model applications for evaluation: Opportunities and ethical implications," *New Directions for Evaluation*, vol. 2023, pp. 33–46, 2023. https://doi.org/10.1002/ev.20556.