AUTOMATIC TOOL FOR BEHAVIORAL ANALYSIS BASED ON STATEMENT ANALYSIS

Alexandra CERNIAN[1], Dorin BURICEANU[2], Valentin SGARCIU[3]

People have always been concerned to know when they are being lied to. Over time, techniques have evolved, and now we rely on scientific instruments such as polygraph and voice stress analyzer to tell if a person in lying. In recent years, text analysis software applications and natural language processing have significantly evolved, thus facilitating decision making processes of various kinds, including lie detection. This paper presents the design and implementation of a statement analysis application, which aims at identifying deceptive behavior based on a text input provided by the user. The application implements four lie detection methods based on statement analysis and works for texts written in English.

Keywords: statement analysis, deception indicators, behavioral analysis, Wordnet, natural language processing

1. Introduction

People have always been concerned to know if a person is lying or not and thus they experienced various methods of spotting possible lies. For example, in ancient China rice grains were used to tell if a person is lying [1]. Namely, a rice grain was put into the suspect's mouth and, if after chewing it, the grain was wet, then the person was telling the truth, otherwise if it was dry, that was a sign of lying. It is believed and proven fact that a dry mouth is a common side effect of lying.

Over time, techniques have evolved, and now we rely on scientific instruments such as polygraph and voice stress analyzer to tell if a person in lying. Other useful techniques are based on the analysis of words and statements, body language and handwriting in order to detect deception.

[1] Lecturer., Faculty of Automatic Control and Computers, University POLITEHNICA of Bucharest, Romania, e-mail: Alexandra.cernian@aii.pub.ro

[2] Eng., Faculty of Automatic Control and Computers, University POLITEHNICA of Bucharest, Romania

[3] Professor., Faculty of Automatic Control and Computers, University POLITEHNICA of Bucharest, Romania, e-mail: vsgarciu@aii.pub.ro

Statement analysis [1] is the process of analyzing the way people use language in the communication process in order to see if they are sincere or deceitful. Words can easily betray people, if you know what to look for in their declarations. Each person has a so-called baseline, a standard way of speaking and acting. Every small deviation from the baseline can signal deception.

In recent years, text analysis software applications and natural language processing [11] have significantly evolved, thus facilitating decision making processes of various kinds, including lie detection. This paper presents the design and implementation of a statement analysis application, which aims at identifying deceptive behavior based on a text input provided by the user. The application implements four lie detection methods based on statement analysis (the verbs tense method, the articles method, the pronouns method, the keywords method and the dictionary method) and it works for texts written in English.

The rest of the paper is structured as follows: Section 2 describes five statement analysis methods which will be implemented in the system hereby presented, Section 3 presents other similar approaches for detecting deception based on text processing, Section 4 presents the design and implementation of the statement analysis system and discusses the results it generates and Section 5 draws the conclusion of this paper.

2. Techniques for detecting deceptive behavior

Statement analysis is one of the most accurate techniques to detect deception, which is used by police officers or FBI agents to investigate cases and suspects. It is based on examining the words and grammar rules used in statements. Under stress conditions, our behavior and language change. The technique is based on identifying changes in the words choices or syntax, which can signal deceptive behavior or a lie.

Here is a list of the most effective statement analysis methods used to detect lies, as presented by Mark McClish research [1].

- the “past tense” verbs rule method
- the articles method
- the pronouns method
- the specific dictionary method
- the keywords method

.

2.2.1 The “past tense” verbs rule method

A common rule is that, when we tell a story that happened in the past, we use the past tense. For most people, it is a natural choice, since they appeal their memory to describe what happened. However, when someone is making up a story, there is no memory to relate to and thus, they will unconsciously start using

the present tense. Let's assume that someone telling you what they did yesterday correctly uses the past tense, but, at some point, switches to present tense just for once. This is a sign that that particular detail was made up.

Exemple:

"I met with John and went to our favorite restaurant. There, a man jumps and me and attacks me."

2.2.2 The articles method

The English language has clear rules about using articles. The indefinite articles "a" and "an" are used to designate an unknown person or thing. Once the person or thing has been introduced, we use the definite article "the". If a person switched back to using indefinite articles, that is a sign of deception. Most likely, they are making up the story.

2.2.3 The pronouns method

The way a person uses pronouns tells a lot about the validity of their words and actions. When someone is willing to assume their actions and take responsibility for what they are saying, they will use the "I" pronoun. The lack of the pronoun suggests a lack of commitment towards the related actions. Moreover, we must also notice when a person tends to overuse the "I" pronoun, which indicates that the person in tensed.

2.2.4 The specific vocabulary method

Each person has her own specific vocabulary and certain choices for words have. A person may refer to her children as "the kids" and another as "my boys". In text analyses, words have unique, specific meaning based on the context. When a person tells the truth, there are no inconsistencies regarding the proper dictionary use, because that person is not subject to stress. Otherwise, when a person intends to deceive, there will be changes in the structure of language, especially when they talk about an invented story which they are not attached to.

2.2.5 The keywords method

Keywords provide additional information and indicate if the suspect is lying. Keywords depend on the context and on establishing a so-called baseline of the suspect [1]. The baseline represents the way the person speaks and acts under no stress. Any deviation from the baseline may indicate that the person is under some stress factors and is feeling uncomfortable, which means that they could have a deceitful behavior.

These words are used in order to replace other words (e.g substitute "no" with "never"), to delay the answer in order to gain some time to make up a story

(“actually”, “well”, “you see”) or to escape incriminating events (“after”, “when”, “then”).

Let’s consider the following example:
Question:
“Did you take the money?”
Answer:
“I never took the money.”

The answer suggests that the subject is likely to be lying, since the question requires a “yes”/”no” type of answer.

3. State of the art. Similar approaches

3.1 Statement Analyzer

Statement Analyzer [2] is a software application which analyses statements using FBI text processing techniques.

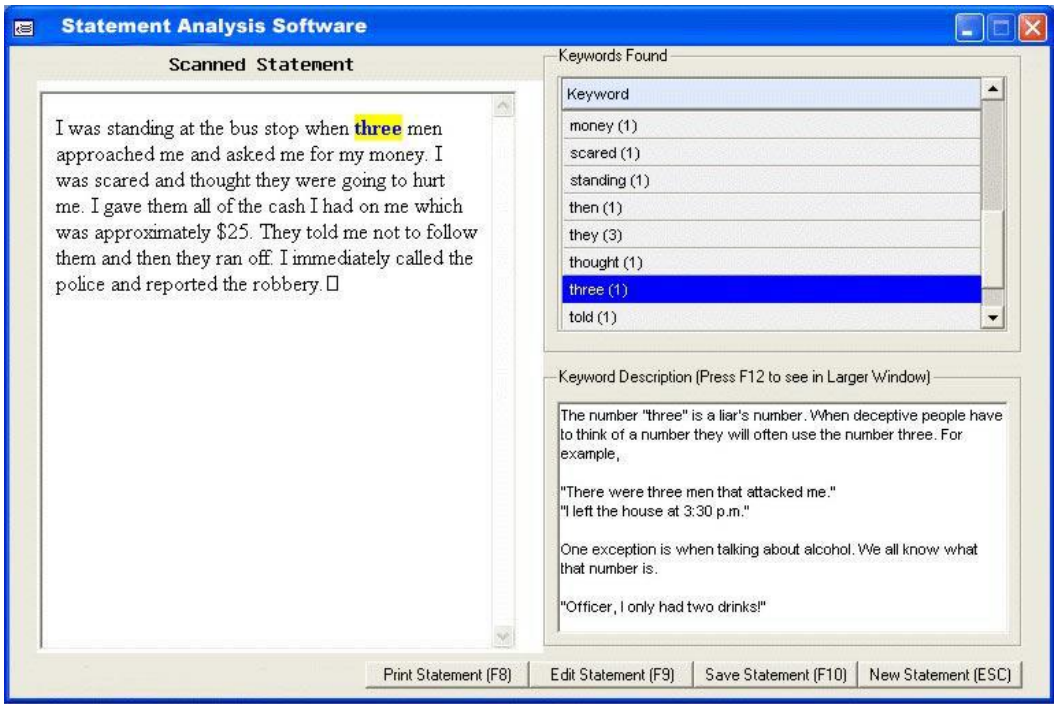

Fig 1. Statement Analyzer (www.statementanalysis.com/analyzer/)

The application returns a set of keywords and the explanations for considering those words as deception clues.

3.2 LSAT Auto Analysis

LSAT Auto Analysis [3] (Linguistic Statement Analysis Technique) is a complex text processing tool, which is able to identify: connections between phrases, missing information, persons, emotions, communication elements. It is often used as a training tool for future investigators.

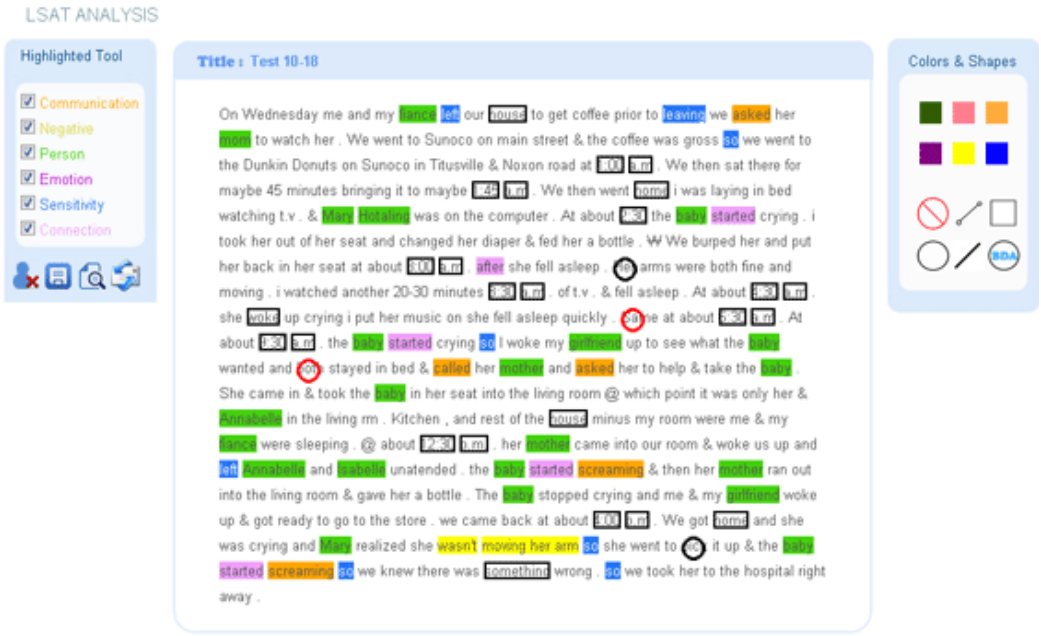

Fig 2 LSAT Auto Analysis (http://linguisticstatementanalysis.com)

3.3 Forensic Statement Analysis

Forensic Statement Analysis [4] uses text analyses techniques to detect deceitful behavior and identify hidden meaning behind statements.

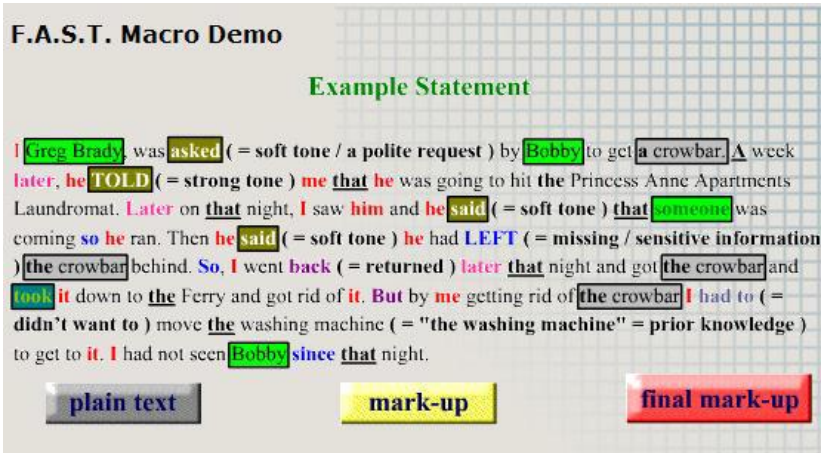

Fig 3 Forensic Statement Analysis

The following section presents the design and implementation of the statement analysis system and discusses the results it generates.

4. The design and implementation of the statement analysis system

The deception detection tool presented in this paper is based on statement analysis techniques, namely on the 5 methods presented in Section 2. The application allows the user to introduce a statement to be analyzed. The application processes it and identifies the parts of the phrase that indicate deception.

The overall flow of the application is depicted in the figure below. The development phase is divided in two parts, the second one depending on the first part. The first part consists of dividing the text into sentences and tokens. The second part involves the development of specific algorithms to filter the tokens and to generate and compute the deception indicators.

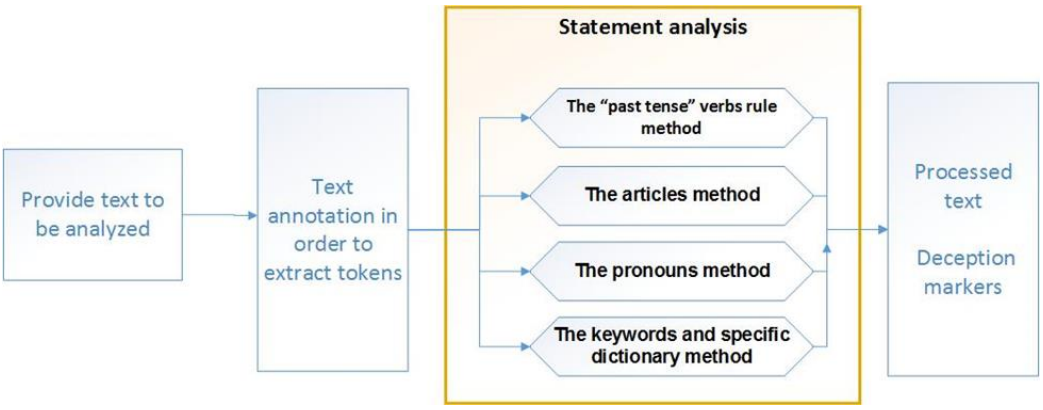

Fig. 4. Statement analysis application flow

Step 1. The user provides the text to be analyzed.

Step 2. The text is processed using several annotation techniques in order to extract tokens.

Tokens [11] are words with specific characteristics, such as part of speech, start index, end index. These properties are extremely useful in natural language processing, as they facilitate the modelling and analysis of natural language, through the identification of syntactical and morphological values and dependencies. In order to extract tokens, the text must first be split into sentences.

For our statement analysis system, we chose to use StanfordCoreNLP [7], which is an integrated system dedicated to natural language processing, including tools for part of speech tagger, named entities recognition, parser and co-reference

issues. For the current application, we have used two of the tools available in StanfordCoreNLP, namely Stanford Part of Speech Tagger (POS Tagger) and Stanford Parser (Parser). **Stanford POS tagger** [6] is the software component which assigns parts pf speech for each word in the text, such as noun, adjective, verb and so on. **Stanford Parser [7]** is a Java implementation of a probabilistic natural language parser. A natural language parser is a program which identifies the grammatical structure of a sentence, such as a group of words forming phrases.

Here are the steps for text annotation:

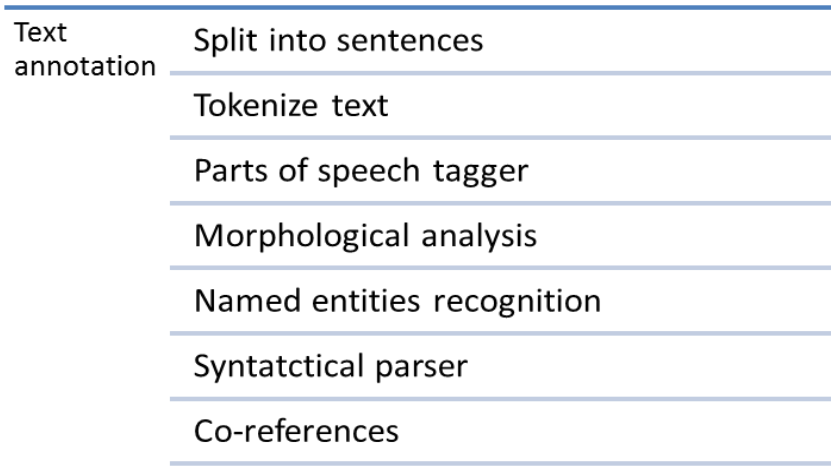

Fig. 5. The steps for text annotation

Step 3. Statement analysis.

During this phase, the tokenized text is further processed in order to establish if there are any deception clues present. The five methods described in Section 2 are applied and the suspicious words or parts of phrase are visually marked for the user to see them.

- The past tense rule method

This method checks each token from a part of speech point of view to verify if it a verb. The Stanford POS Tagger has been used to check if a verb is at present tense or past tense. Whenever a verb is at present tense, it is marked as a deception indicator.

- The articles method

The method checks the use of articles in the text. During the annotation phase, articles are marked as POS: "DT" and possessive pronoun as POS "$ PRP". These two parts of speech are those that may precede a noun to introduce it into the story. However, only the indefinite article "a" / "an") introduces a person or

thing according to the English grammar. Also, between a possessive pronoun or an article and the noun they precede, adjectives can be interspersed. The method checks for the first occurrence of a noun within the text and marks it in a list of nouns. When the noun is encountered again, the article used to introduce it is verified. If it is an indefinite article, the noun is marked as a deception indicator.

- The pronouns method

This method focuses mainly on the use of the "I" pronouns. As described in Section 2, there are two cases of interest: the absence of the "I" pronoun, which indicates that the person is trying to separate themselves from the events described and an overuse of the "I" pronoun, which indicates that the person is lying or making up the story.

In this case, we needed a syntactical analysis of the text instead of a morphological analysis, as for the first two methods presented above. Thus, a strong asset for this purpose is the Stanford Parser [7], which makes an accurate representation of dependencies and grammatical relationships in sentences and phrases. The algorithm uses a graph approach to represent words and relationships among them. The algorithm counts the number of "I" pronouns, the number of subject-predicate relationships and the number of coordination relationships. If the number of pronouns is equal to 0 and the number of subject-predicate relationships is also 0, then we mark the absence of the "I" pronoun. If the number of pronouns is greater than the number of coordination relationships, then we mark the overuse of the "I" pronoun.

The steps of the algorithm are the following:

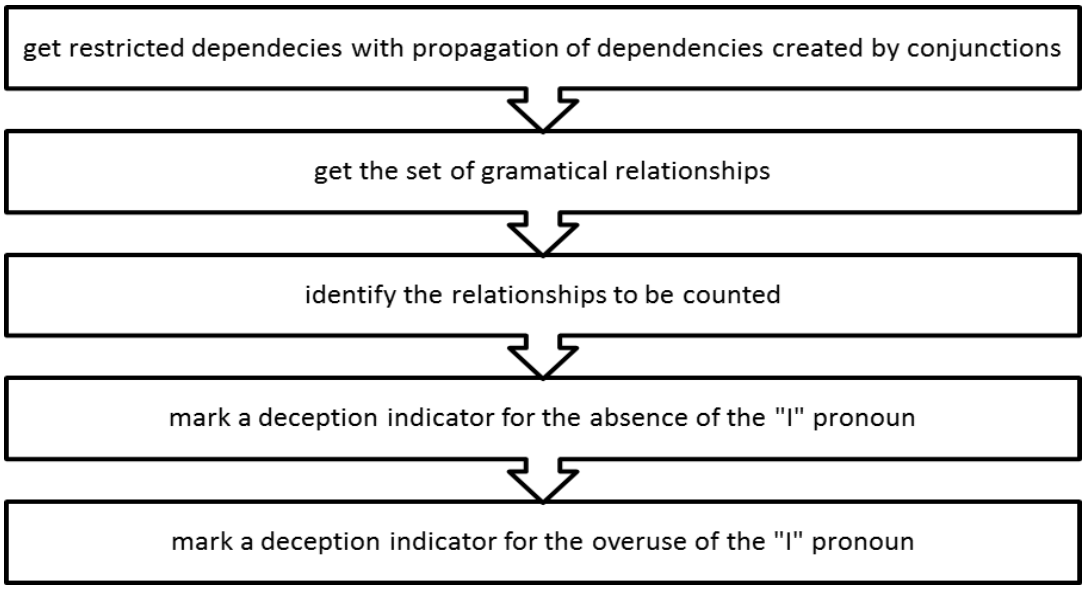

Fig. 6. The pronouns method algorithm

Step 4. The keywords and specific dictionary method.

In order to use this method, it is first necessary to create a profile of the so called "suspect", namely of the person's whose statements are being analyzed. For the application to detect any inconsistency in the statements of the suspect, we must first detect the base line of the suspect and create a database of the specific words for each suspect. Afterword, when statements are analyzed, the application uses Wordnet [10] in order to find related words for the dictionary in the database.

The algorithm is based on the tokens obtained in the annotation phase and aims to identify in the analyzed statements various forms of the words in the suspect's dictionary. The steps are the following:

- Adapt the tokens for Wordnet search
- Extract the list of hyperonyms and hyponyms from Wordnet
- Check if the related words are found in the analyzed statement

The architecture of the system is presented in figure 7, illustrating the complete structure of components and their interactions.

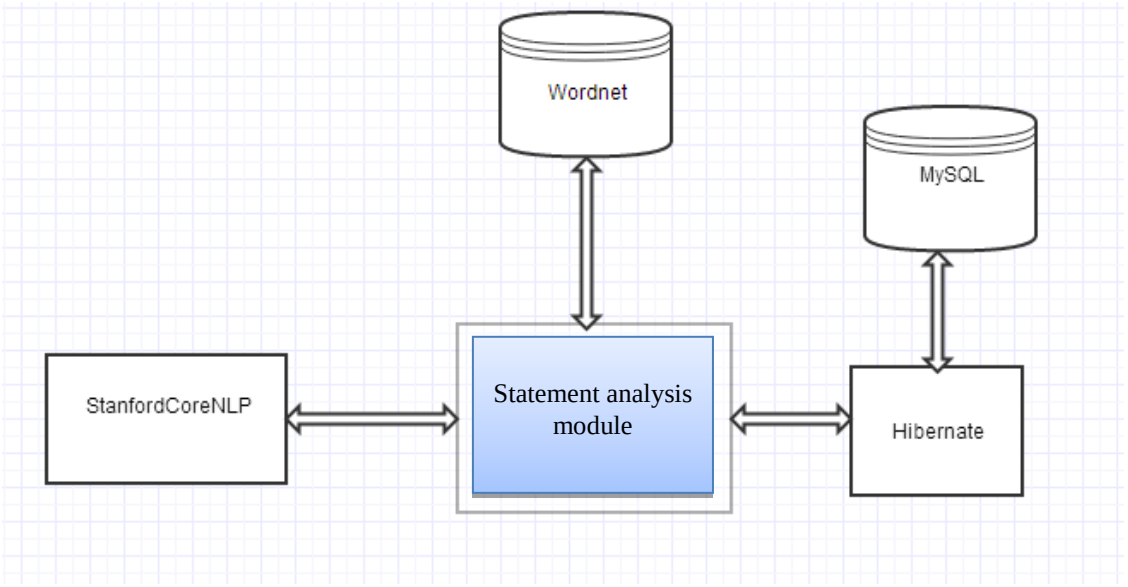

Fig. 7. The system architecture

The system can be used using only one of the statement analysis methods or different combinations. The most advanced way to process statements is using all the methods described above. Here is an example of using all 4 deception detection techniques which confirms that the statement analysis system is accurate and provides relevant results.

Let us consider the following statement as input for the system:

"Woke up, got dressed and went to John's restaurant. After a time, a strange man entered. When I left the restaurant this guy attacks me with the knife."[1]

As it can be noticed, we are confronted here with deception indicators corresponding to all 4 statement analysis methods:

- In the first phrase, the subject is missing and we are dealing with the absence of the "I" pronoun (the pronouns method)
- The verb "attacks" is at present tense (the past tense verbs method)
- The use of a synonym for the word "man" (the specific dictionary method)
- The use of a definite article when "the knife" is introduced in the story (the articles method)

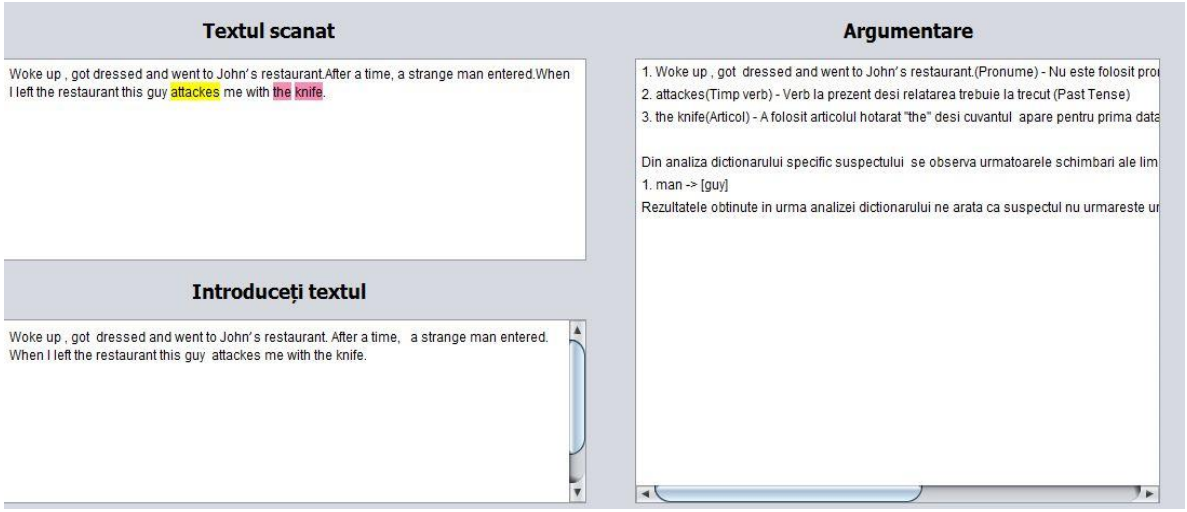

Fig. 8. The results of the statement analysis system

In order to validate the statement analysis system, we used a set of 50 statements between 20 and 250 words and we obtained an accuracy ranging between 82% and 100%. For shorter and clearer the text, the accuracy was higher, the system being able to correctly identify all deception indicators. Compared to other similar applications, our tool performed in the same range of accuracy. Statement Analyzer [2] reported an average deception detection rate of 84%, reaching even 100% accuracy in several situations, while LSAT [3] claimed to be on average 92% accurate in determining deception.

5. Conclusions

People have always been concerned to know when they are being lied to. Over time, techniques have evolved and significant progress has been made with

natural language processing, which addresses a lot of issues with regard to decision making processes, including lie detection. This paper presents the design and implementation of a statement analysis application, which aims at identifying deceptive behavior based on a text input provided by the user. Statement analysis is one of the most accurate techniques to detect deception, based on a clear set of theoretical rules and techniques. It is based on examining the words and grammar rules used in statements. Under stress conditions, our behavior and language change. The technique is based on identifying changes in the words choices or syntax, which can signal deceptive behavior or a lie.

Based on the results obtained, it can be concluded that deceptive behavior can be automatically identified with a high degree of accuracy, ranging between 82% and 100%, given that the techniques addressed are strictly based on the rules of English grammar. The application identifies repeated mistakes of the same type without producing erroneous results and generates accurate interpretations of the statements. Advanced analysis can be conducted using the four methods implemented. The results mark the deception indicators, together with the corresponding explanations and arguments.

Acknowledgements

The work has been partly funded by the Sectoral Operational Programme Human Resources Development 2007-2013 of the Ministry of European Funds through the Financial Agreement POSDRU/159/1.5/S/132395.

R E F E R E N C E S

[1]. *Mark McClish,* 10 Easy ways to spot a liar, The Marpa Group Inc. ,2011

[2] Statement Analysis: http://www.statementanalysis.com/analyzer/

[3] The LSAT Auto-Analysis statement analysis program: http://linguisticstatementanalysis.com/index.php

[4] Forensic Statement Analysis: http://www.deceptionanalysis.com/truth_tools_fast_macro.html

[5] Stanford CoreNLP : www-lp.stanford.edu/software/corenlp.shtml#History

[6] Stanford Log-linear Part-Of-Speech Tagger: http://www-nlp.stanford.edu/software/tagger.shtml

[7] The Stanford Parser: A statistical parser: www-nlp.stanford.edu/software/lex-parser.shtml

[8] *Marie-Catherine de Marnee & Christopher D. Manning,* Stanford typed dependencies manual, 2008

[9] Using Hibernate in a Java Swing Application: https://netbeans.org/kb/docs/java/hibernate-java-se.html#03a

[10] Wordnet: http://wordnet.princeton.edu/wordnet/

[11] *Bing Liu*, "Sentiment Analysis and Opinion Mining", Morgan & Claypool Publishers, 2012