

CREATIVITY STIMULATION TOOL

Andreea OPRIȘAN¹, Ștefan TRAUȘAN-MATU²

Abstract. *This paper describes and applies the General Morphological Analysis method for creativity stimulation, in order to support people in their creation process. This tool suggests new ideas, based on the ones already discussed in an XML chat file. The main idea of the application is that it extracts the key words of a conversation and it generates new words based on them. Then, the General Morphological Analysis is applied on these words to generate sets of related words. This method also implies user intervention that leads to even better results, because of the human-computer collaboration.*

Keywords: Creativity, General Morphological Analysis, WordNet, Frequency, Parser, Term Frequency – Inverse Document Frequency

1. Introduction

The Artificial Intelligence domain has shown serious interest in developing artificial creative agents that might reproduce the human ones. Therefore, attempts were made for developing software that generates literary texts, poetries, analogies and neologisms. However, the human mind and its processes of thinking and imagining have yet many unknowns. They influenced by many factors, making these processes to be unpredictable and impossible for the computer to accurately reproduce them. For this reason, human intervention in creativity processes has not been abandoned. Moreover, researchers have been trying a hybrid approach, based on collaboration between the human factor and the artificial one.

New and innovative ideas may come from the most unusual associations, not being based on a certain thinking pattern or on a certain relationship between events or concepts. In order to have new ideas people should have an open mind, not limited by thinking constraints. It is said that a man, in order to maintain his brain trained, must always learn new things, must experience new situations all the time, must try all sort of different activities and must never fall in the everyday routine.

That's why we consider that it is almost impossible to create a software/robot to completely replace human imagination and creativity, because people create their

¹Eng., PhD student, Faculty of Automatic Control and Computers, University Politehnica of Bucharest, Bucharest, Romania, (andreea.oprisan7@gmail.com).

²Corresponding member of AOSR. Prof., PhD, Faculty of Automatic Control and Computers, "Politehnica" University of Bucharest and Senior Researcher, Research Institute for Artificial Intelligence of the Romanian Academy, Romania. stefan.trausan@cs.pub.ro.

imagination on the basis of the large variety of different experiences in life. We expect that, the larger this variety is, the bigger chances are for someone to invent and create original things.

A right approach would be, in our opinion, the one in which these intelligent systems come in people's support, offering them tools to develop their creativity or to give them new ideas concerning a certain problem or opportunity. Using this idea as a start point, we have reached to the idea of developing intelligent aids that brings new ideas in a chat conversation (between two or more persons), based on the ones already discussed in the chat. The application doesn't exclude the contribution of human creativity. Therefore, it allows the interaction between the user and the software, not only allowing the user to make a few selections, but also to make different connections between terms and words. For the development of this application it has been used a creativity stimulation method named General Morphological Analysis (GMA), that has been adapted for the program requirements.

The developed software is specialized on texts written in English. The creativity stimulation process is based on a document in which a chat conversation between two or more persons is saved. We have chosen the chat conversation because some brainstorming sessions are made in online group discussions, especially when the implied persons are in different towns or countries.

The next section of the paper represents an introduction in the discussed issue: research and development of creativity stimulation agents along time, in the field of Artificial Intelligence and motivation for the choice of this domain.

Section 2 presents the context in which the application is developed; more exactly, it shortly presents different methods of creativity stimulation. Here we describe more explicitly the GMA method on which this project focuses.

This section talks also about similar projects in this domain that use different methods of stimulating creativity. These software tools are described in accordance with their specifications on the official site.

Section 3 contains the description of the implementation of the application, containing two parts: the theoretical part and the actual implementation part. The theoretical part presents the utilities used: API, the WordNet database, as long as different techniques of parsing word in a text. For the implementation, there are specified the information flows and the structure of the software, regarding the programming part.

The fourth section describes the application functions as long as the user's interaction with it, step-by-step. There is also presented the graphic interface and the instructions to use the program.

Section 5 analysis two runs of the program, referring to the results obtained for certain input data. Also, there are analysed the running performances on computers with different specifications.

Section 6 concludes the paper, highlighting the application's utility and also future work and extensions.

2. State of the art

According to the definition, creativity refers to a person's capacity to invent something new: a product, a piece of art, a solution, an idea, based on some ideas and concepts already thought. Every human has a certain amount of creativity, and connections between ideas and concepts are made all the time in human's brain. However, only some persons have the gift of inventing something totally innovative and valuable.

Another definition of creativity is „the ability to see a challenge or a problem in a new light and to find solutions for it that weren't obvious, until then. [1, 2,3]

The capacity of a person to be very creative is usually gained in time, preferable from young ages, using different games, techniques, connections. A good way to stimulate creativity for children is to give them books to read, because it creates the process of imagining new situations and landscapes, using only the words, without visual intervention of pictures. That is why stimulating creativity implies stimulating the imagination, lateral, divergent thinking, but also intuition.

Usually, good ideas are generated in a group, because the ideas get to be correlated, extended and new ideas are created starting from one idea that each person tries to extend it, according to the type of imagination he possesses. To support creativity stimulation, some techniques considered very useful have been developed, such as brainstorming, the 6-3-5 method, "FRISCO", "Philips 66", "Dephi", the Synectics method, and the General Morphological Analysis [1, 2].

The General Morphological Analysis method has been used in developing the application presented herein. It is a method used to stimulate creativity based on analysing and combining concepts from different domains, to generate new ideas. This method has been invented by Fritz Zwicky in 1940's, as a technique of analysing relationships in a multidimensional space. In the last few years, this technique has been adapted and used to create planning and to determine new scenarios. This is achieved through combining and juxtapositioning different scenarios and situations.

In general, "morphology" refers, for certain scientific disciplines, to studying the exterior form and the internal structure. For example, in linguistics is the studying of the rules for word modification for their different usages. In biology is the

study of the exterior form and interior structure of plants and animals organisms. Zwicky wanted to use this term for a more general form: „I have proposed to generalize and systematize the concept of morphological research and include not only the study of the shapes of geometrical, geological, biological, and generally material structures, but also to study the more abstract structural interrelations among phenomena, concepts, and ideas, whatever their character might be.” [4].

Tom Ritchey, who adapted in 1995 the method for solving different complex problems/scenarios, defines it this way: ”Morphological analysis is a method for identifying and investigating the total set of possible relationships or "configurations" contained in a given problem complex. In this sense, it is closely related to typology analysis, although GMA is more generalised in form and has far broader applications.”

We will proceed by describing the steps for applying GMA, as described by Ritchey [5, 6, 7].

2.1 Defining the parameters

Parameters for defining the scenario are established, and after that, a table is created with these parameters and their values.

Each parameter can have any number of values. For example, the number of values for the parameter A doesn't depend on the number of values for the parameter B.

Parametrul A	Parametrul B	Parametrul C
A1	B1	C1
A2	B2	C2
A3	B3	C3
A4	B4	C4
A5	B5	
	B6	

Fig. 1. Defining the problem's parameters.

2.2 Setting the constraints

Starting from the table constructed at the previous step, another table is created, in a matrix structure, in which constraints between parameters values are being established. In this table the cells where two attributes can coexist are checked.

For example, let's consider that we have 2 ideas: "very large space" and "hut". These 2 ideas contradict themselves, so they cannot coexist.

		Parametru A					Parametru B					
		A1	A2	A3	A4	A5	B1	B2	B3	B4	B5	B6
Parametru B	B1											
	B2											
	B3		✓									
	B4		✓									
	B5											
	B6											
Parametru C	C1				✓							
	C2		✓			✓		✓				
	C3		✓						✓			
	C4											

Fig. 2. Setting the constraints.

A table like this one makes sure that the consistency for every value for a certain parameter is analysed with all the other values from the others parameters and that there aren't taken into consideration the pairs of values of the same parameter.

2.3 Determining the outputs

After defining the constraints, we go back to table no.1 and we set the problem's inputs, more exactly, we select values from the parameters columns.

Parametru A	Parametru B	Parametru C
A1	B1	C1
A2	B2	C2
A3	B3	C3
A4	B4	C4
A5	B5	
	B6	

Fig. 3. Determining the outputs.

The outputs are being generated depending on the inputs and on the inputs constraints with the values not used as inputs.

In the example from figure 3, the inputs are coloured in red and the outputs in blue. This example is based on the 2 other figures (figure 1 and figure 2).

Inputs: A2

Consistency between pairs: A2 – B3, A2 – B4, A2 – C2, A2 – C3,
A4 – C1, A5 – C2, B2 – C2, B4 – C3

Outputs: B3, B4, C2, C3

The outputs are determined based on the fact that there is a connection between parameter A2 (input), and the outputs. Multiple inputs can be selected.

2.4 Related work

The GMA method is used in some institutions or companies from multiple domains to identify new scenarios to approach a problem/opportunity. General Morphological Analysis has been adopted by Tom Ritchey, former Research Director for the Institution for Technology Foresight and Assessment at the Swedish National Defence Research Agency (FOI) in Stockholm. He put his effort in solving complex problems and scenarios, using this method. He started researching in this field in 1995, when he developed the first version of the MA software. Since then, he was involved in over 100 projects from Sweden Government's Agencies, organizations or private agencies, as it is stated in the official site of MA Software: <http://www.swemorph.com>.

The software has been used in different scenarios, such as:

- Identifying long term alternatives for evaluating terrorist tentative risks and taking measures against them
- Analysis of economic crimes
- Pharmaceutical planning (<http://www.swemorph.com/pdf/wp-pharma.pdf>) [8]
- Taking actions in case of accidents caused by dangerous substances (<http://www.swemorph.com/pdf/chem2.pdf>) [9]
- Strategies for reducing disasters caused by natural hazards (<http://www.swemorph.com/pdf/multi.pdf>) [10]

MA/Carma (<http://www.swemorph.com/macarma.html> [11]) is a software tool based on morphological analysis, used as a developing strategies platform. This software has been implemented for the first time in 1995 and today it has reached its 5th version.

It has 3 main phases: editing, consistency and display.

Edit Field refers to setting some parameters, with his values in a table.

Cross-Consistency Field refers to establishing constraints between the values of the parameters in the table.

Display Field refers to highlighting the scenarios, based on the inputs in the initial table.

3. Implementation

The application we have developed has the goal to support the participants of a chat and to help them find new ideas for discussions, based on the ones already discussed.

The creativity stimulation program has the following implementation details:

Programming language: Java

Used APIs:

- Stanford University API, for file parsing
 - Stanford Parser
 - Stanford POS Tagger
- Princeton University API, for accessing WordNet database

Other programs:

- chat parser from the FP7 LTfLL Project - Language Technologies for Lifelong Learning
- WordNet database

The implementation can be divided in 2 stages: in the first one, the XML file is being parsed and the topics are obtained and in the second stage, words from the same field as the topics domain are being generated and the GMA is applied on them.

3.1 Determining the topics

Finding the topics consists in the following steps:

- Parsing the XML file. This implies extracting every word from the text, tagging it with a part of speech and stemming it.
- Processing the extracted words, giving them a score depending on their importance in the text. This thing is achieved with the help of a method described later: Text Frequency- Inverse Document Frequency

We will present below some theoretical details regarding the implementation, and then the actual implementation will be presented.

3.1.1 Theoretical aspects

The topic, according to a definition, is a phrase from a paragraph or a word in a sentence that defines the discussion subject from that sentence/paragraph. Usually, in a sentence, the topic is the same with the subject (as part of the sentence).

In our approach, for finding the topics we first parsed the XML file and we extracted the nouns, taking into consideration the idea that nouns are usually the subject of a sentence. Then we applied the Tf-Idf method for giving each noun a score, according to their relevance in the text. After this, the words are ordered descending by score. The topics with the highest score are the most relevant for the analysed text.

3.1.1.1 Parsing the XML file

In general, a Natural Language Processing parser analyses the grammatical structure of a sentence/text and gives every word different parameters, such as parts of speech, word dependencies, etc. In the first stage, some functionalities of the XML chat parser, developed for the FP7 LTfLL Project (<http://www.ltfl-project.org>) were used - Language Technologies for Lifelong Learning [12]. This parser also uses some libraries distributed for use by The Stanford University: Stanford Parser and Stanford POS Tagger [13]. The Stanford POS Tagger also comes with an API for Java, that reads a text in a certain language and tags every word with a part of speech, such as noun, verb, adjective, adverb, etc. [13]

3.1.1.2 Term-Frequency – Inverse Document Frequency

Tf-Idf (Term-Frequency – Inverse Document Frequency) is a method to obtain the relevance of a certain word for a certain file. To implement Tf-Idf a corpus must be used, to correlate with the word for which the score is required. [14]

Tf-Idf takes into account the fact that some words are more common than others. Practically, Tf reflects which are the most frequent words for the document, while Idf (obtained from the corpus) gives greater scores the rare words. For example, the word „the” would have a very high Tf, but because it is present in almost all the documents in the corpus, the Idf is almost 0.

More exactly, the following notations are used:

tf (Term Frequency) – the frequency of a word in document.

Idf (Inverse Document Frequency) – measures if a word is frequent or rare in the training corpus.

For obtaining tf and idf, the following notations are introduced:

d – the document from which the words are extracted, to apply tf-idf on them

t – a term from the d document

$Count(d,f)$ – number of appearances of the term t in the document d

$Corpus$ – training corpus

$|D|$ – number of documents from the corpus

$|\{d \in D: t \in d\}|$ – the number of documents from the corpus in which the term t appears

Obtaining tf:

There are many ways to calculate tf:

$$\begin{aligned} tf(t, f) &= Count(w, f) \\ tf(t, f) &= 1 + \ln(Count(t, f)) \end{aligned} \quad (1)$$

In this case, $tf(t, f) = 1 + \ln(Count(t, f))$ formula has been used, to diminish the effect of exhaustive repetitions.

Calculation for idf:

$$idf(t, D) = \log \frac{|D|}{|\{d \in D: t \in d\}|} \quad (2)$$

Usually, in case there is no appearance for the term t in the corpus, the formula becomes:

$$idf(t, D) = \log \frac{|D|}{1 + |\{d \in D: t \in d\}|} \quad (3)$$

Calculation for the term score:

$$score(t, d) = tf(t, d) * idf(t, D) \quad (4)$$

For example, if we have the word „flower”, that appears in the document for 7 times, and in the training corpus (that has 3000 documents), appears in 29 of these, the score is being calculated this way:

$$idf(t, D) = \log \frac{|D|}{1 + |\{d \in D: t \in d\}|} = \log \frac{3000}{1 + 29} = \log 100 = 2$$

$$tf(t, f) = 1 + \ln(Count(t, f)) = 1 + \ln 7 = 2.94$$

$$score(t, d) = tf(t, d) * idf(t, D) = 2.94 * 2 = 5.88$$

3.1.2 Implementation details of topics determination

The implementation consists of the following steps:

- The InverseDocumentFrequency object receives a list of topics and each topic is formed by a triplets as this: <word, part_of_speech, score>
- From this list, another one is being generated, with TopicAppearance type elements, and each element from this list is a pair of this form: <topic, word_appearance>, where topic represents a topic from the received topic list and word_appearance is used to determine if that topic appears in a certain document or not: word_appearance is 1 if the topic appears in the document, otherwise it becomes 0.
- A ThreadPool is used, that assigns tasks of type Task, in order to search the list of topics in the document. An object of type Task gets as argument a list of topics with objects of type TopicAppearance (association between the topic and a variable set on 1 if the topic appears in a certain file, otherwise set on 0) and the path of a file in the training corpus. In this way, each word forms the list of topics is searched in parallel in a file. After finishing the execution of all tasks, the results are unified in the list of topics.
- For each document from the training corpus it is created an object of type Task, that receives as an argument the list of topics of type TopicAppearance and the document.
- After all tasks from the ThreadPool finish their execution, the results are unified and for each word it is calculated in how many documents from the training corpus it appears in.
- The score value for each topic from the initial list is updated, according to the formulas previously presented:

$$\bullet \quad idf(t, D) = \log \frac{|D|}{1 + |\{d \in D : t \in d\}|} \quad (1)$$

$$\bullet \quad score(t, d) = tf(t, d) * idf(t, D) \quad (2)$$

A task has the role to settle if a topic list appears in a given document and for each topic from the list, the field word_appearance is being updated at 1 if it appears, otherwise is left to 0. The task follows the following steps:

- Receives as arguments a path to a file and a list of elements of type TopicAppearance (<topic, word_appearance>).
 - Gets every word from the file received as argument and compares it with every word from the list of topics.
 - If it finds the word in the document, it sets word_appearance on 1
-

- At the end, each topic from the list will have the following values for `word_appearance`: 1 – if it appears, 0 – otherwise.

Task assignment by the ThreadPoll is done according to the following scheme:

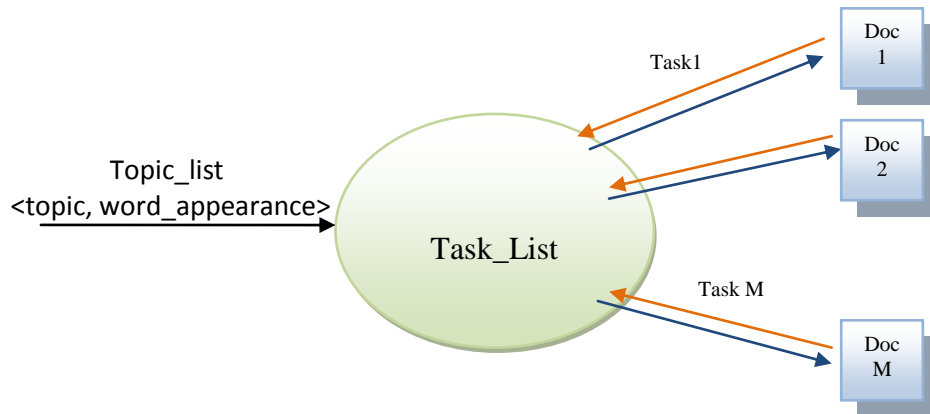


Fig. 4. Task assignment for searching in the document, as it is done in the application.

Topic_list - the argument each tasks receives, and on which it makes changes, returning the same list, with the field `word_appearance` modified for each element, if needed

Task_list - list with M tasks

M - number of documents from the corpus

Doc X - the Document with number X from the corpus

3.2 GMA implementation

Briefly, the GMA implementation consists of the following steps:

- Generates, for each selected topic, maximum 8 similar words, using WordNet.
- Creates a table, similar to the one from the figure 1, in which the parameters are the topics and the attributes are the word previously generated using WordNet.
- Creates a table similar to the one from figure 2, in which the user sets the constraints he wishes between the words.
- Permits the user to select his inputs, according to figure 3.
- Displays the outputs, according to figure 3.

This second part of the program is also divided in two steps. Firstly, words are generated that are from the same field of each selected topic by the user, based on the topics previously determined in the first stage. The WorldNet software is used

in this phase. Secondly, the program allows user interaction, to apply the GMA method.

For this application implementation, hypernyms and hyponyms from the WordNet software have been used, in order to determine similar word with a certain word. Hyponym, as is mentioned in dictionaries, is a word or phrase whose semantic field is included within that of another word, its hypernym (sometimes spelled hyperonym outside of the natural language processing community. In simpler terms, a hyponym shares a type-of relationship with its hypernym.

For example, scarlet, vermilion, carmine, and crimson are all hyponyms of red (their hypernym), which is, in turn, a hyponym of colour.

Unlike the classical method, where the scenarios are defined by the user, with its parameters and values inserted in a table as the one from Figure 1, this program automatically generates the parameters attributes, using the WordNet database.

More exactly, the parameters are the topics selected by the user from a list, and for each of them, hypernyms and hyperonyms are searched in the database files and are automatically inserted in the table.

Then, the user must decide on his own the constraints between these words, so he completes a table as the one from Figure 2, checking the cells for each pair he considers that have a connection and can coexist in a situation/context.

These constraints are saved and the original table is shown again. Now, the user is allowed to select the inputs and the program displays the outputs, according to the constraints previously selected by the user. In this way, new connections between words are created, proposing new ideas and new subjects for discussion.

4. Functionalities

For the user, there are four main steps to use the program:

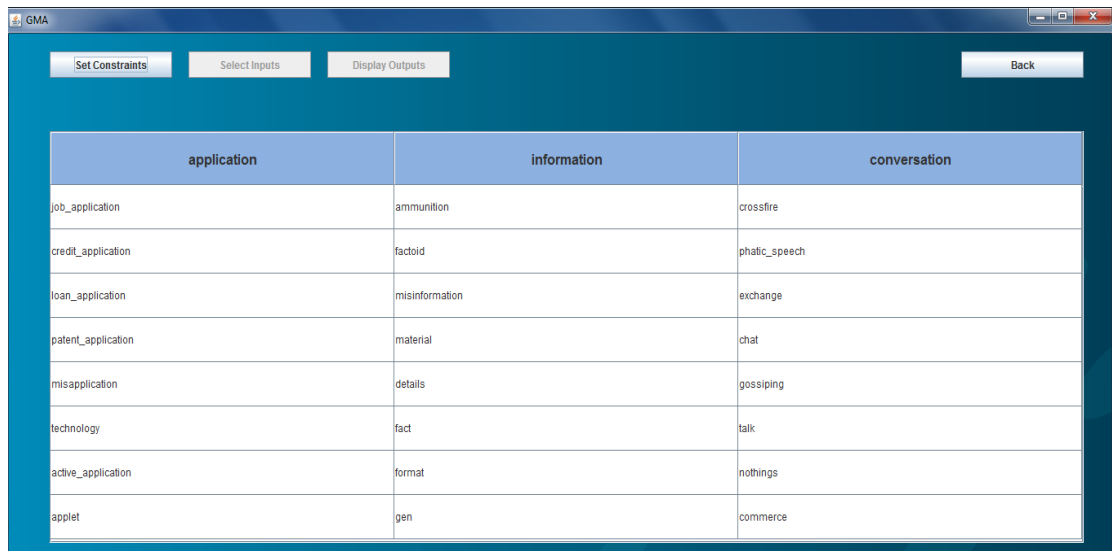
- Selection of the file and topics
- Display of the table with the generated words
- Defining the constraints
- Selection of the inputs and display of the outputs

4.1 Selection of the file and topics

The topics extracted from the text are displayed in a table, with their associated part of speech and score and the user can select the topics he prefers to be processed with the GMA method. The list of topics is sorted descendant by score.

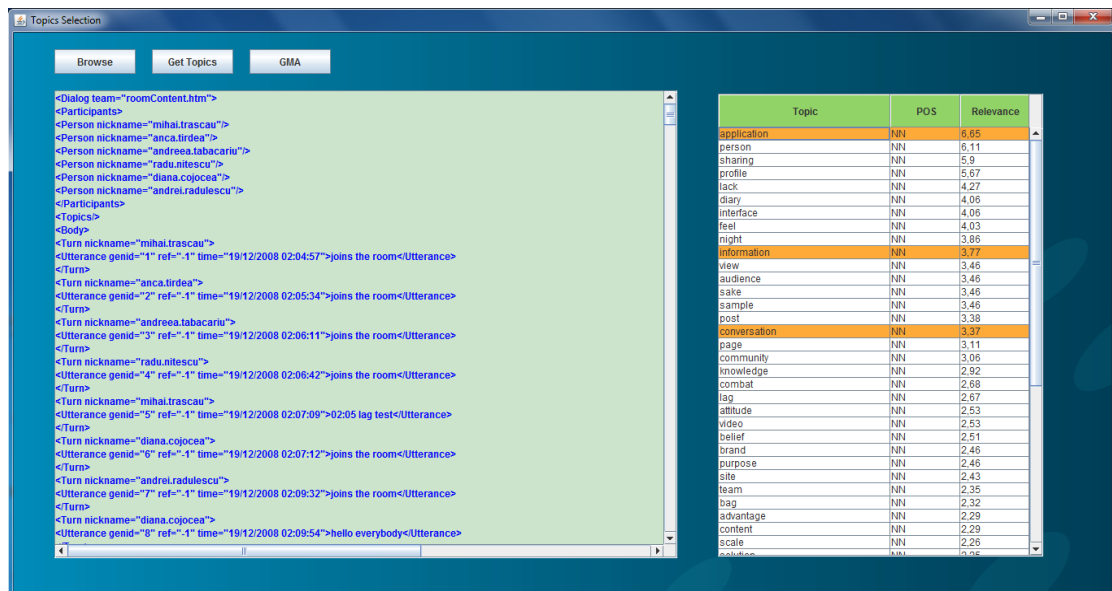
4.2 Table display with the generated words

For the second step, a table is created, having as headers the topics previously selected. On each column, there are generated words that are related with the column names. There are generated a number of maximum 8 words for each topic. There has been chosen a number of maximum 8 words because a larger number of words would confuse the user when he would have to choose the pairs of words in the constraints table.



application	information	conversation
job_application	ammunition	crossfire
credit_application	factoid	phatic_speech
loan_application	misinformation	exchange
patent_application	material	chat
misapplication	details	gossiping
technology	fact	talk
active_application	format	nothings
applet	gen	commerce

Fig. 5. Selection of the file and topics window.



Topic	POS	Relevance
application	NN	6.65
person	NN	6.11
sharing	NN	5.9
profile	NN	5.67
lack	NN	4.27
diary	NN	4.06
interface	NN	4.06
feet	NN	4.03
night	NN	3.86
information	NN	3.77
view	NN	3.46
audience	NN	3.46
sale	NN	3.46
sample	NN	3.46
post	NN	3.38
conversation	NN	3.37
page	NN	3.11
community	NN	3.06
knowledge	NN	2.92
combat	NN	2.68
lag	NN	2.67
attitude	NN	2.53
video	NN	2.53
belief	NN	2.51
brand	NN	2.46
purpose	NN	2.46
site	NN	2.43
team	NN	2.35
bag	NN	2.32
advantage	NN	2.29
content	NN	2.29
scale	NN	2.26
scale	NN	2.26

Fig. 6. GMA Window.

The generated words are hyponyms and hyperonyms for the word in the header of the table for each column. Initially, the hyponyms are searched in the WordNet database and are taken as much as it exists.

Then, if it doesn't reach the limit of 8 words, hyperonyms are searched and is taken only the first one, because if it would go deeper in the hyperonyms hierarchy, the words would distance themselves from the original sense of the topic.

4.3 Defining the constraints

In the third phase, the user has to check the pairs of words he thinks that could be related. Initially, the table contains all the cells unchecked. The constraints between words are considered the ones that remain unchecked.

It can be noticed that the topics are in the top and left cells, and for each topic, the words generated in the previous stage are displayed. As it was previously presented at the description of the GMA method, this table permits settling a relation between any two words of each topic.

Defining the constraints will later influence the combinations of words that will arise. This process is very subjective, mostly depending on the imagination and perspective of the user.

Aside from using this program by a single user, a good idea would be to compare the results with different users, for the same selected topics.

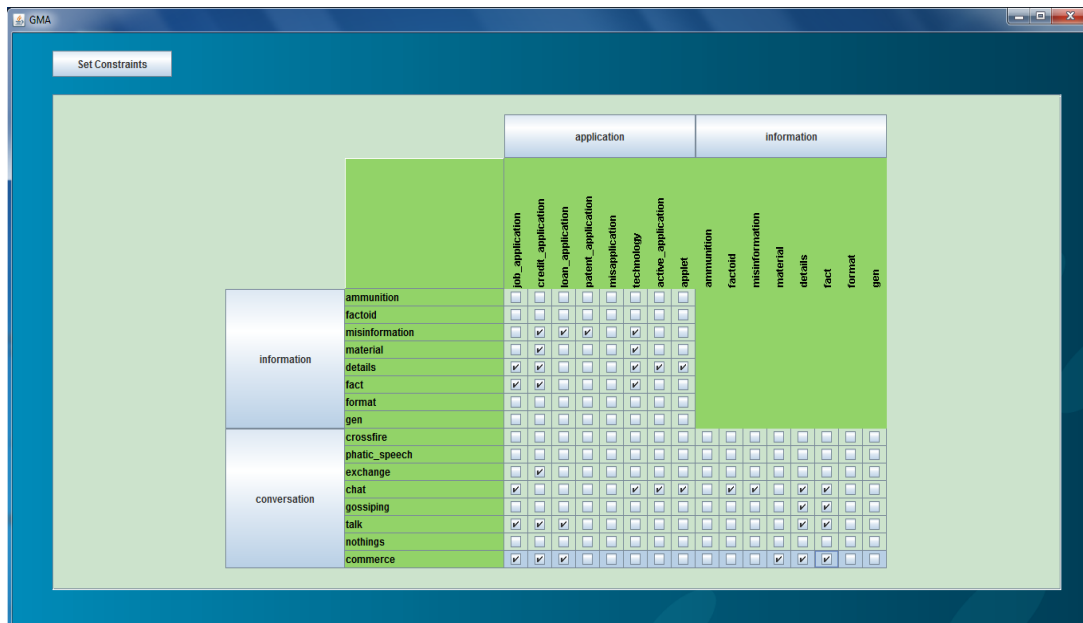


Fig. 7. Constraints Window.

4.4 Input selection and output display

In the last stage, the user goes back to the table from step 2, but he has the possibility to select inputs. After he selects the inputs, the outputs are displayed, being coloured in a different colour.



Fig. 8. GMA Window

5. Results

The execution of the first phase lasts some minutes, depending on the user's system. The reason for this is that the process of parsing the file to extract relevant words from text is done for a large number of files. This step is processed in parallel, on multiple threads, so the performance increases depending on the number system's processors. Below is presented an execution example, with the results obtained for certain inputs. For the topics extraction request, the following table is obtained:

Table 1

List of Topics		
Topic	POS	Relevance
writer	NN	6.92
application	NN	6.65
person	NN	6.11
profile	NN	5.67
lack	NN	4.27
bunch	NN	4.06
diary	NN	4.06
interface	NN	4.06
night	NN	3.86
technology	NN	3.86
information	NN	3.77

people	NNS	3.61
view	NN	3.46
audience	NN	3.46
sample	NN	3.46
sake	NN	3.46
conversation	NN	3.37
bed	NN	3.22
page	NN	3.11
community	NN	3.06
knowledge	NN	2.92
setback	NN	2.79
combat	NN	2.68
game	NN	2.68
article	NN	2.67
lag	NN	2.67
attitude	NN	2.53
video	NN	2.53
belief	NN	2.51
brand	NN	2.46
purpose	NN	2.46
bit	NN	2.35
bag	NN	2.32
content	NN	2.29
scale	NN	2.26
solution	NN	2.25
type	NN	2.21
example	NN	2.18
course	NN	2.12
security	NN	1.91
damage	NN	1.91
progress	NN	1.85
question	NN	1.73
aim	NN	1.6
mean	NN	1.58
name	NN	1.51
control	NN	1.38
support	NN	1.32
day	NN	1.15
world	NN	1.15
time	NN	1.08
feedback	NN	1
fun	NN	1
info	NN	1
moderator	NN	1
exchange	NN	0.9

Scenario 1:

The following topics are selected: writer, application, information (these ones have a big score, so they are relevant for the conversation).

Based on them, the following table is generated:

Table 2

writer	application	Information
abstractor	job_application	Ammunition
alliterator	credit_application	Factiod
authoress	loan_application	Misinformation
biographer	patent_application	material
coauthor	misapplication	details
commentator	technology	fact
contributor	active_application	format
Cyberpunk	applet	gen

The following pairs of words are chosen as having a connection between them, checking them in the constraints table:

job_application - biographer	details – biographer
job_application - contributor	details – coauthor
credit_application - contributor	details – commentator
loan_application - contributor	details – contributor
technology - abstractor	details – job_application
technology - coauthor	details – credit_applicatio
active_application - coauthor	details – loan_application
applet - abstractor	details – misapplication
applet - coauthor	details – technology
misinformation - commentator	fact – biographer
misinformation – misapplication	fact – commentator
material – biographer	fact – technology
material – technology	fact – active_application

Inputs set 1:

From table 1, it is chosen as an input the word: *biographer*

The following set of ideas is obtained:

biographer – job_application - material
biographer – job_application - details
biographer – job_application – fact

Inputs set 2:

From table 1, it is chosen as an input the word: *misinformation*

The following set of ideas is obtained:

commentator – misapplication - misinformation

Inputs set 3:

From table 1, it is chosen as an input the word: *technology*

The following set of ideas is obtained:

abstractor – technology - material

abstractor – technology - details

abstractor – technology - fact

coauthor – technology - material

coauthor – technology - details

coauthor – technology - fact

Scenario 2:

The following topics are selected: knowledge, audience, conversation (these ones have a big score, so they are relevant for the conversation).

Based on them, the following table is generated:

Table 3

knowledge	audience	conversation
mind	gallery	crossfire
place	grandstand	phatic_speech
public_knowledge	house	exchange
episteme	readership	chat
ability	viewing_audience	gossiping
inability	gathering	talk
lexis	multitude	nothings
vocabulary	opportunity	commerce

The following pairs of words are chosen as having a connection between them, checking them in the constraints table:

gallery – place

galery – public_knowledge

grandstand – public_knowledge

viewing_audience – place

gathering – mind

gathering – place

gathering – public_knowledge

gathering – ability

gathering – lexis

gathering – vocabulary

opportunity – ability

exchange – gathering

exchange – opportunity

chat – mind

chat – public_knowledge

chat - lexis

chat - vocabulary

chat – house

chat - gathering

chat – opprotunity

gossiping – vocabulary

gossiping – gallery

gossiping – grandstand

gossiping – gathering

gossiping – opportunity

talk – lexis

talk – vocabulary

commerce – mind

commerce – ability

commerce – gallery

commerce – grandstand

commerce – viewing_audience

commerce – gathering

commerce - opportunity

Inputs set 1:

From table 1, it is chosen as an input the word: *commerce*

The following set of ideas is obtained:

mind – gallery - commerce
mind – grandstand – commerce
mind – viewing_audience - commerce
mind – gathering - commerce
mind – opportunity – commerce
ability – gallery - commerce
ability – grandstand – commerce
ability – viewing_audience - commerce
ability – gathering - commerce
ability – opportunity - commerce

Inputs set 2:

From table 1, it is chosen as an input the word: *public_knowledge*

The following set of ideas is obtained:

public_knowledge – gallery - chat
public_knowledge – gathering - chat

Inputs set 3:

From table 1, it is chosen as an input the word: *gossiping*

The following set of ideas is obtained:

vocabulary – gallery – gossiping
vocabulary – grandstand - gossiping
vocabulary – gathering - gossiping
vocabulary – opportunity - gossiping

6. Conclusions

According to the results, it can be noticed that different combinations are obtained, depending on the domain the user leans more to. Also, it can be observed that in a chat conversation, more interpretations can be obtained. For example, for the scenario number 2 previously presented, new correlations from 2 different areas were obtained: commerce (inputs set 1) and communication (inputs set 2 and 3).

Moreover, on the basis of a single chat conversation, two different scenarios were obtained, having different inputs sets. So, different ranges of ideas, that apparently don't have any connections between them, have been generated.

In conclusion, the results of the program have a big range of variety, being different from one user to another. This variety is achieved because imagination and vision of every user play an important part. Also something that influences the outcome is the user's preference for certain domains: for example, he leans to exact domains (mathematics, informatics), artistic domains (poetry, painting,

theatre). Usually, people that tend more to artistic domains have a wider perspective over the relationships that can appear between words, but it is not a general valid rule. Also, there are people who have both these sides well developed.

In the future, this application could be extended, integrating the brainstorming technique in the one that uses GMA. More exactly, a good idea would be the simultaneous discussion of multiple users in a chat.

One future implementation idea would be that all the users select the same topics and then, each of them applies the GMA method on the topics, settling the combination of words according to their own perspective and imagination. After these combinations are obtained for every user, they can be analysed in the group in order for the best ideas to be selected.

Another method would be that every user to select topics that seem to them as being the most interesting (according to the score, but also to their own opinion) ,and then to run the GMA method on them. The results will also be compared and the best ideas will be selected, after they have been discussed with everyone.

In this field, the creativity field, nothing is quite exact and nothing follows a certain rule or pattern. Therefore, any method that helps creating new ideas is welcomed.

REFERENCES

- [1] Ș. Traușan-Matu, *Studiu asupra Tehnicilor de Stimulare a creativității folosind instrumente informatice*, Academia Romana, June 2010
- [2] Ș. Trăușan-Matu, Computer Support for Creativity in Small Groups using chats, in *Annals of the Academy of Romanian Scientists, Series on Science and Technology of Information*, vol.3, nr.2, 2010, pp. 81-90
- [3] A. Weston, *Creativity for Critical Thinkers*, Ed. All, Bucuresti, 2008
- [4] T. Ritchey, *General Morphological Analysis – A general method for non-quantified modelling*, June 2010. www.swemorph.com, Swedish Morphological Society (Visited on 2012/04/04)
- [5] T. Ritchey, *Modelling Complex Socio-Technical Systems Using Morphological Analysis*, June 2010. www.swemorph.com, Swedish Morphological Society. (Visited on 2012/04/04)
- [6] T. Ritchey, *Problem Structuring using computer-aided morphological analysis*, 2006: www.swemorph.com, Swedish Morphological Society. (Visited on 2012/04/04)
- [7] T. Ritchey, *Future Studies using Morphological Analysis*, 2009: www.swemorph.com, Swedish Morphological Society. (Visited on 2012/04/04)
- [8] N. Hussain, T. Ritchey, *Wicked Problems – Achieving clarity in complex pharmaceutical problems: from innovation to policy formulation*, European Industrial Pharmacy, September 2011. www.swemorph.com, Swedish Morphological Society. (Visited on 2012/06/10)
- [9] T. Ritchey, M. Stenstrom, H. Eriksson, *Using Morphological Analysis for evaluating Preparedness for Accidents Involving Hazardous Materials*, Swedish Defence Research Agency (FOI), November 2002 www.swemorph.com, Swedish Morphological Society. (Visited on 2012/06/10)
- [10] T. Ritchey, *Modelling Multi-Hazard Disaster Reduction Strategies with Computer-Aided Morphological Analysis*, January 2005. www.swemorph.com, Swedish Morphological Society. (Visited on 2012/06/10)
- [11] ***, Swedish Morphological Society, 2005-2012. <http://www.swemorph.com/macarma.html>. (Visited on 2012/05/28)
- [12] S. Trausan-Matu, P. Dessus, T. Rebedea, S. Mandin, E. Villiot-Leclercq, M. Dascalu, A. Gartner, C. Chiru, D. Banica, Dan; D. Mihaila, B. Lemaire, Benoît; V. Zampa, E. Graziani, *LTfLL - D5.2 Learning support and feedback*, 2009, <http://dspace.ou.nl/handle/1820/2251>
- [13] ***, The Stanford Natural Language Processing Group, Stanford University <http://www-nlp.stanford.edu/software>. (Visited on 2012/04/14)
- [14] ***, Document and query weighting schemes, Cambridge University Press, 2008. <http://nlp.stanford.edu/IR-book/html/htmledition/document-and-query-weighting-schemes-1.html>. (Visited on 2012/05/15)
-