ROBOTS IN ASSISTED
LIVING ENVIRONMENTS

UNOBTRUSIVE, EFFICIENT, RELIABLE AND MODULAR SOLUTIONS FOR INDEPENDENT AGEING

Research Innovation Action

Project Number: 643892 Start Date of Project: 01/04/2015 Duration: 36 months

DELIVERABLE 3.8

Social network analysis component

Dissemination Level	**Public**
Due Date of Deliverable	Project Month 24, March 2017
Actual Submission Date	31 March 2017
Work Package	WP3, *Modular conceptual home architecture design and ICT method development for efficient, robust and flexible elder monitoring and caring*
Task	T3.4, *Developing social network analysis as a sensor*
Lead Beneficiary	NCSR-D
Contributing Beneficiaries	
Type	R
Status	Submitted
Version	Final

Project funded by the European Unions Horizon 2020 Research and Innovation Actions

Abstract

This report documents the development of the RADIO mood monitoring method. The method is based on the RADIO tool for collecting statistics about social network messages that match user-defined patterns. The tool is validated by using it to define patterns that correspod to interRAI assessment items.

History

Version	Date	Reason	Revised by
01	6 Feb 2017	Document structure	NCSR-D
02	15 Mar 2017	The RadFa Web app (Section 2)	NCSR-D
03	28 Mar 2017	Experiments on interRAI mood items (Section 3)	NCSR-D
04	30 Mar 2017	Internal review	TWG
Fin	31 Mar 2017	Addressed review comments, final document preparation and submission	NCSR-D

Executive Symmary

This report documents the development of the RADIO mood monitoring method. The method is based on RadFa, the RADIO tool for collecting statistics about social network messages that match user-defined patterns. RadFa is a Web application that uses the Facebook Graph API to access the posts and comments feed of a specific Facebook group. RadFa can be configured for different group, assuming access to the group administrator's credentials.

RadFa is a Web application where clinical staff can monitor aggregate statistics about the content of the social network interactions of the primary end-users they supervise. By configuring RadFa to display statistics about vocabulary items that are relevant to interRAI assessment items, clinical staff can get indications relevant to the behaviours and moods that are part of interRAI assessmen.

The RadFa application was approved by the Facebook Team under the *user_managed_groups* permission, prohibiting the application of sentiment analysis tools. Therefore RadFa can only display (without further processing) statistics about the content of a specific person's posts. This makes it against the Facebook terms and conditions to tightly integrate RadFa in the RADIO Home system, but RadFa and the clinical information it produces will be part of the RADIO ecosystem of monitoring tools.

Abbreviations and Acronyms

Graph API	Programmatic interface that allows Web apps access to the Facebook Graph, including the posts and comments made in a group by any user.
interRAI LTCF	The interRAI Long-Term Care Facilities Assessment System
JSON	JavaScript Object Notation, a lightweight data-interchange format.
NLTK	Platform for developing Python programs to work with human language data.
WordNet	Multi-lingual lexical database, providing the semantic relations (synonimy, hyperonimy, etc.) between terms in many languages, including English and Greek.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Purpose and Scope

This document reports on the social network monitoring method developed in Task 3.4 for collecting statistics about vocabulary that is characteristic of mood and other behaviour features. This document is complemented by the RadFa prototype implementation (also developed in Task 3.4) that is available as open source software on `https://bitbucket.org/dataengineering/radfa`

The prototype on the public repository is coupled to the Facebook API but is not configured for a particular Facebook group. The fully configured prototype will be included in D3.10 *Integrated data analysis system*, where will be setup for monitoring the forum created for the RADIO pilot users.

1.2 Approach

This deliverable is prepared in Task 3.4, *Developing social network analysis as a sensor*. This task first developed a general-purpose Facebook application that computes statistics about thematic facets over the posts and comments appearing in a Facebook group. This application is then validated on facets defined to match interRAI assessment items.

In order to motivate interaction, the project will set up a forum where end-users can exchange their experience with RADIO and also get technical support. This will be done in the context of Task 6.3.

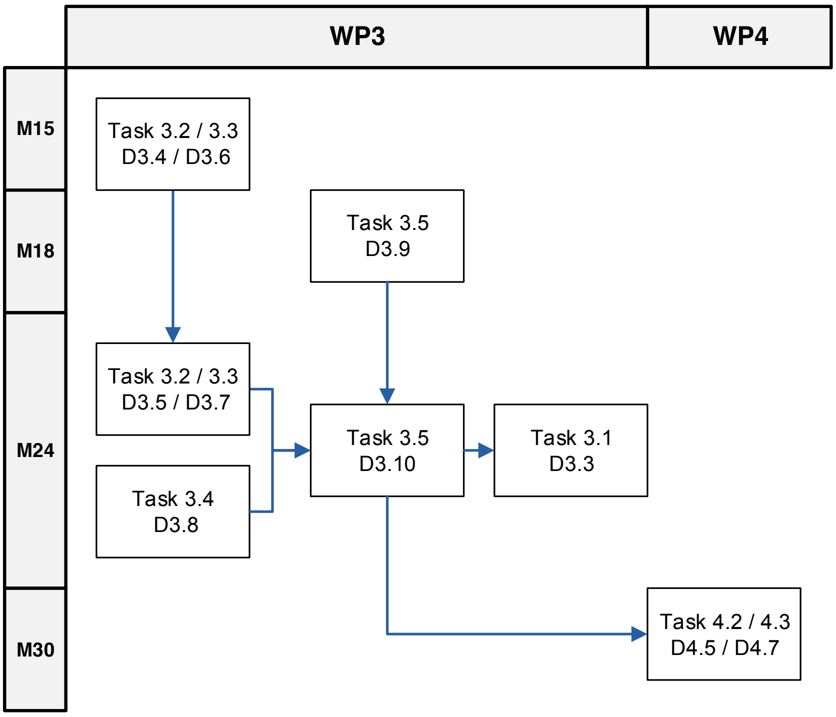


Figure 1: Dependencies between this deliverable and other deliverables

1.3 Relation to other Work Packages and Deliverables

This document is part of a cluster of closely related deliverables. Work around ADL and Mood recognition is organized as follows:

- *Conceptual architecture II (D3.2):* Report, documenting the design and interconnections of the recognition components and used to guide development work.
- *ADL and mood recognition methods (D3.4 and D3.5)*, *Network robustness and efficiency methods (D3.6 and D3.7)*, and *Social network analysis component (D3.8, this document):* Reports, documenting the methods developed in order to recognize the activities and mood of the primary users. These reports are complemented by their open source prototypes found at the repositories indicated in each report and also collected in the *Conceptual architecture* documents.
- *Integrated data analysis system II:* Software, integrating the prototypes above into a coherent ADL/mood recognition system.
- *Conceptual architecture III (D3.3):* Report, documenting the final design and interconnections of the recognition components after changes and adjustments carried out during integration work.

These dependencies are also graphically depicted in Figure 1.

2 The RadFa Application

2.1 Facebook Apps Platform

The *Facebook for Web Developers platform*[1] is a system of APIs for integrating a web app with Facebook. In particular, the *Graph APIs*[2] is an HTTP-based API that allows web apps to get access to the Facebook Graph, including the posts and comments made in a group by any user. The *Facebook Graph* represents Facebook's information as a set of:

- **nodes**: basic components such as User, Group, Post etc
- **edges**: the connections between those components such as User's groups, Group's feed, Post's comments and likes etc
- **fields**: information and attributes about those components such as the name of a User, the members of a Group, the message of a Post etc

All nodes and edges can be read by making HTTP GET requests to the relevant endpoint and all fields can be collected by setting the appropriate parameters. The host of Graph API is *graph.facebook.com* and commonly a version identifier (vX.Y) is set to the start of the request path. RadFa application uses the APIs of the version 2.8, therefore the prefix of each URL endpoint is *graph.facebook.com/v2.8*.

As the main scope of RadFa application is to provide an interface to users to collect some basic statistics based on the posts of the groups that are admins of, specific API calls are required in order to gather all the suitable information needed. Those API calls require the Facebook Login (*FB.login*) to obtain an access token which provides temporary, secure access to Facebook APIs. Specifically, the *Facebook SDK for JavaScript* [3] obtains and persists user access tokens automatically in browser cookies, where the access token property is included in the response of *FB.getAuthResponse* call. After the successful login, a call to the endpoint **/me* is required in order to translate the access token of the user you made the request to the user ID of the person. The user ID or the access token are necessary parameters for the API calls that correspond to the retrieval of user's specific information.

Since RadFa app needs to obtain the groups in which the user is an admin of, an API call to the node *User* with the edge *groups* is made. The endpoint for this call is */{user ID}/groups* , while the fields that are needed are the id (group ID) and the name of each group.

In addition, to retrieve the feed of posts in a group and information about the members who posted the respective messages, a batched request is made based on the access token. Facebook Graph allows batching performance by making a HTTP POST request to the Graph API endpoint. Specifically, batching allows to pass instructions for several operations in a single HTTP request or to specify dependencies between related operations [4].

The batching request in RadFa consists of two parts. The first part of the batch, results in the feed of posts in a group during a specific period of time. This operation is performed on the node *Group* with the edge *feed*, where the time range is applied on the *updated_time* field. The endpoint of this operation is **/{group ID}/feed?since={since_timestamp}&until={until_timestamp}*, where *since* and *until* parameters define the limits of the time range.

The second operation of the batch depends on the first one, because it references the result of the first one. In particular, the second operation uses *JSONPath* syntax to extract information of each post from the output of the first operation, such as the post's comments and likes as well as attributes of the user

[1] See `https://developers.facebook.com/docs/web`

[2] See `https://developers.facebook.com/docs/graph-api`

[3] See `https://developers.facebook.com/docs/facebook-login/access-tokens`

[4] See `https://developers.facebook.com/docs/graph-api/making-multiple-requests`

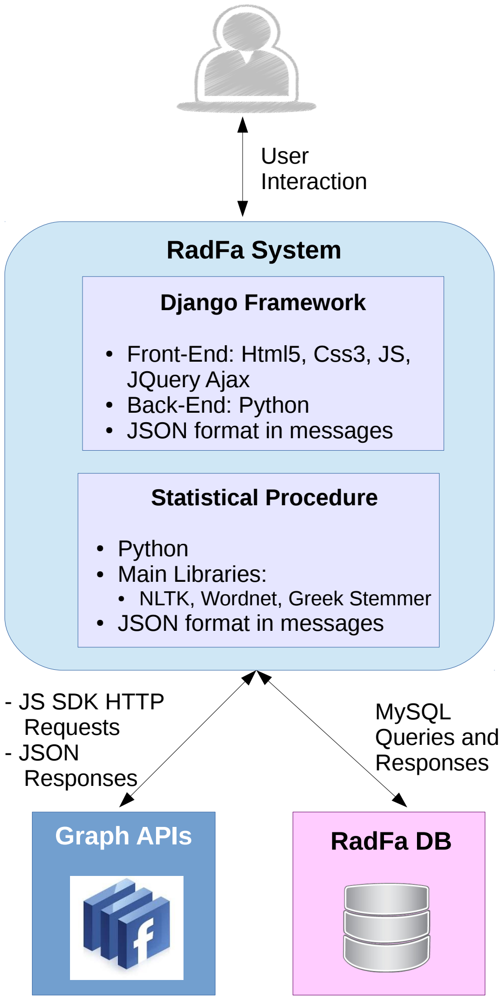


Figure 2: Abstract Structure of RadFa System

who posted it (user id and name). Finally, the response of the batch request encapsulates the result of both operations.

As we mentioned above, the *Graph API* calls that are made in the RadFa app require the use of access an token, where RadFa can generate by implementing Facebook Login. Regarding which *Graph API* entities are needed, different permissions should be determined and requested in the user's access token. By taking into account the RadFa needs, the *Group* node and the *feed* edge are requested for reading access. Therefore, the *user_managed_groups* permission should be passed along with the login request. As a result, the users should accept this permission when the are about to log in to RadFa the first time.

The RadFa application was approved from the Facebook Team under the *user_managed_groups* permission. According to the terms and policies of Facebook Team, the specific permission can not be used to apply sentimental analysis upon the data. Therefore RadFa app can solely display discriminative statistics based on the content of people's groups posts , where the people are members in this group.

2.2 RadFa Design

RadFa is a Web application where clinical staff can monitor aggregate statistics about the content of the social network interactions of the primary end-users they supervise. Specifically, clinical staff can create a Facebook group, whose members are a set of participants that they supervise. As the participants share their experience or thoughts through the group, the clinical staff see statistics that may indicate

Table 1: The schema of the RadFa database

Table	Field Name	Data Type	Description
Facet	id	Integer	Auto Increment primary key
	name	Varchar	The name of the facet (maximum 30 characters)
	id_user	BigInteger	The user ID that represents the user who created the facet
	keywords	Longtext	The keywords of the facet. Each keyword is separated by comma
	phrasesIn	Longtext	The included phrases of the facet. Each phrase is separated by comma
	phrasesOut	Longtext	The excluded phrases of the facet. Each phrase is separated by comma
History	id	Integer	Auto Increment primary key
	f_id	Integer	Foreign key of the Facet table
	g_id	BigInteger	The group ID
	id_user	BigInteger	The user ID that represents the user who created the facet
	date	Varchar	The date of the storing
	period	Varchar	The time range where the group's messages are posted
	num_partic	Integer	The total number of participants in the results
	details	Longtext	The resulted data and statistics as plain text, compatible for JSON conversion

the participants' behavioural mood. These statistics are based on the vocabulary that is indicative for behaviour symptoms or mood state, as specified in the interRAI LTCF [5]. Therefore, the clinical staff can choose characteristic keywords or phrases (called *Facet* in RadFa and been described in Section 2.2.2), which express mood and behavioural assessment items, and get the observations automatically through RadFa. This Section displays the architecture and the functionality of RadFa System, while Section 2.3 describes the statistical procedure in more detail.

An abstract structure of the RadFa System and its external calls is displayed in Figure 2. As we can see there, RadFa is implemented in Python language and is composed of two parts. The first part represents the web application and is developed with the use of the Django framework [6]. In particular, Django is an open source web framework written in Python which follows the model–view–controller (MVC) architectural paradigm and is suitable for both front-end and back-end development. The second part of RadFa System corresponds to the statistical procedure. The statistical process uses tools and libraries for Natural Language Processing (NLP) in English and in Greek language, such as NLTK,[7] WordNet,[8] and the NCSR-D stemmer for Greek.[9] The messages that are exchanged between the different entities and through the different layers of the whole system are in JSON format.

Moreover, RadFa takes advantage of the Facebook Graph APIs to get the appropriate information for the statistical process. As we mentioned in Section 2.1, Facebook provides a set of client-side functionality through Facebook SDK for JavaScript. Therefore, RadFa's HTTP requests to the Graph APIs are made with the use of Facebook's SDK JavaScript, where the SDK is asynchronously loaded into RadFa's

[5] Cf. Section 3.5, D2.2 Early Detection Methods and Relevant System Requirements II

[6] See `https://www.djangoproject.com`

[7] See `http://www.nltk.org`

[8] See `https://wordnet.princeton.edu`

[9] See `https://bitbucket.org/dataengineering/greek-stemmer`

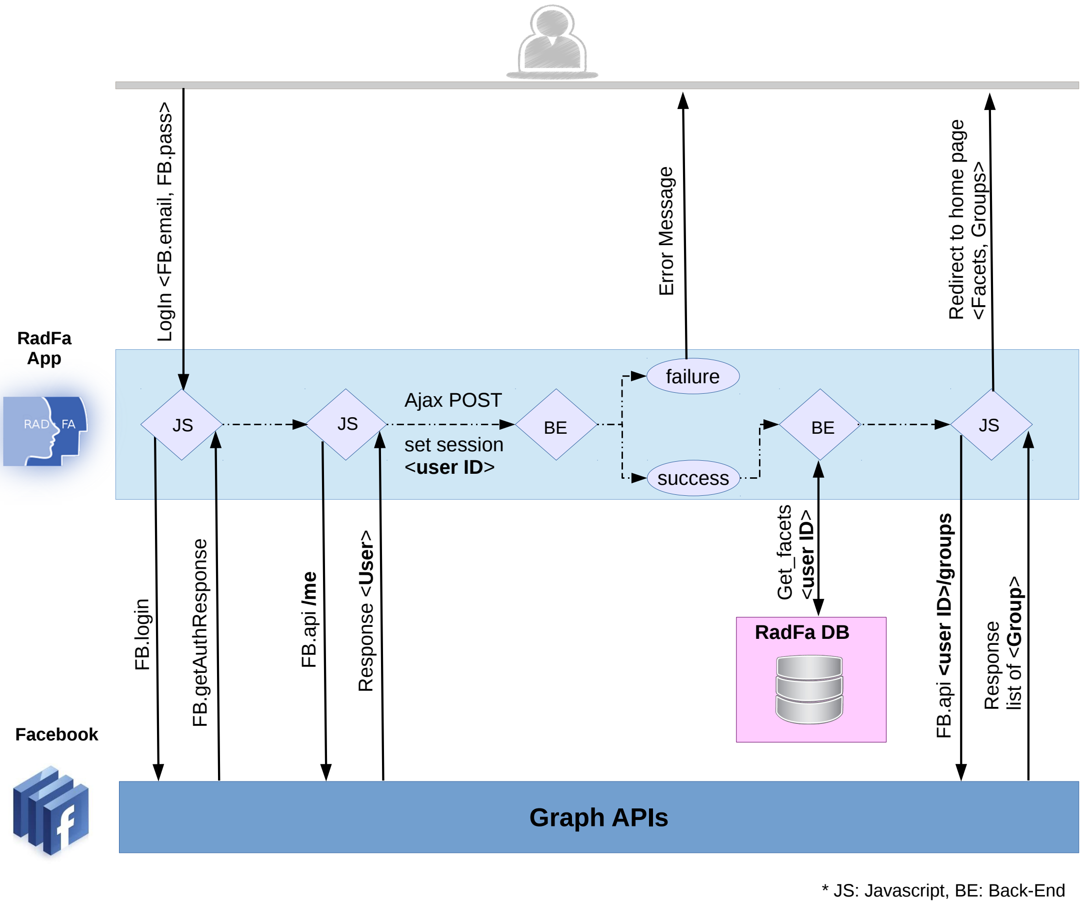


Figure 3: Diagram of the login procedure

pages with JavaScript. The responses of the Graph API contain the results in JSON format.

In addition, RadFa System requires a database where the clinical staff (or RadFa's users) can store their Facets and the searching results that contain the statistics (history). The RadFa database is a relational MySQL database described by two tables — Facet and History. Table 1 gives the schema of the RadFa database.

2.2.1 Login

Figure 3 depicts the flow of the login procedure in RadFa. The clinical staff can log in to RadFa application with their Facebook credentials by clicking the characteristic *Log In* Facebook button. When the login is successful, the response of the Graph API encloses an object which contains the access token of the logged in user. A call to the endpoint **/me* is necessary, as explained in the Section 2.1, thus in the end the system receives the basic user's information such as user ID, user's name. In addition, RadFa System makes an Ajax POST request in order to establish a session by setting the user ID. User ID is necessary to make the next calls to the Graph APIs and to the RadFa's database. The successful response of the request leads the user to the core of RadFa app - *home page*, where the main functionality of the system exists. This functionality, which is described in more details in the Sub-Section 2.2.3 , requires the user's facets and groups. Therefore in the Back-End of the system, a call in the RadFa's database is required in order to get the facets that the user have already added in the system. Then the system redirects the user to the home page, where the loaded JavaScript of the page makes a call to the endpoint **/groups* to get the groups of the user that is admin of.

2.2.2 Facet

The clinical staff can create and store their own Facets, in order to use them in the statistical searching in frequent periods of time. Each Facet consists of characteristic words or phrases that describe an assessment item in an abstract way. In particular, those keywords and phrases are the representative vocabulary or user's utterance, which contribute to the determination of the statistical searching.

RadFa app provides a suitable interface where the users can Add/Edit/Delete their Facets. The creation of a Facet requires the filling of the following fields:

- **Name**: the name of the Facet
- **Keywords**: characteristic words for the statistical search
- **Included Phrases**: characteristic phrases for the statistical search

We shall collectively call the *Keywords* and the *Included Phrases* fields the *attributes* of the Facet. The user can fill one or both attributes, with one or more values, as long as at least one attribute has at least one value. All attribute values are logically OR'ed, so that satisfying any one attribute value is enough to include a message in a Facet.

Specifically for the *Keywords* field, it is also possible to add as value a list of terms separated by '&'. This option denotes that all keywords in the list must be present in the message (in any order) for the message to match the facet.

The user can add one new value (one term or a list of terms) at a time by pressing the respective button, thus making more complex sets in the *Keywords* field. On the other hand, the user can add only one phrase at a time into the *Included Phrases* field. Moreover, the added values are updated and displayed in a specific text area at the same web page, every time the user adds a new value.

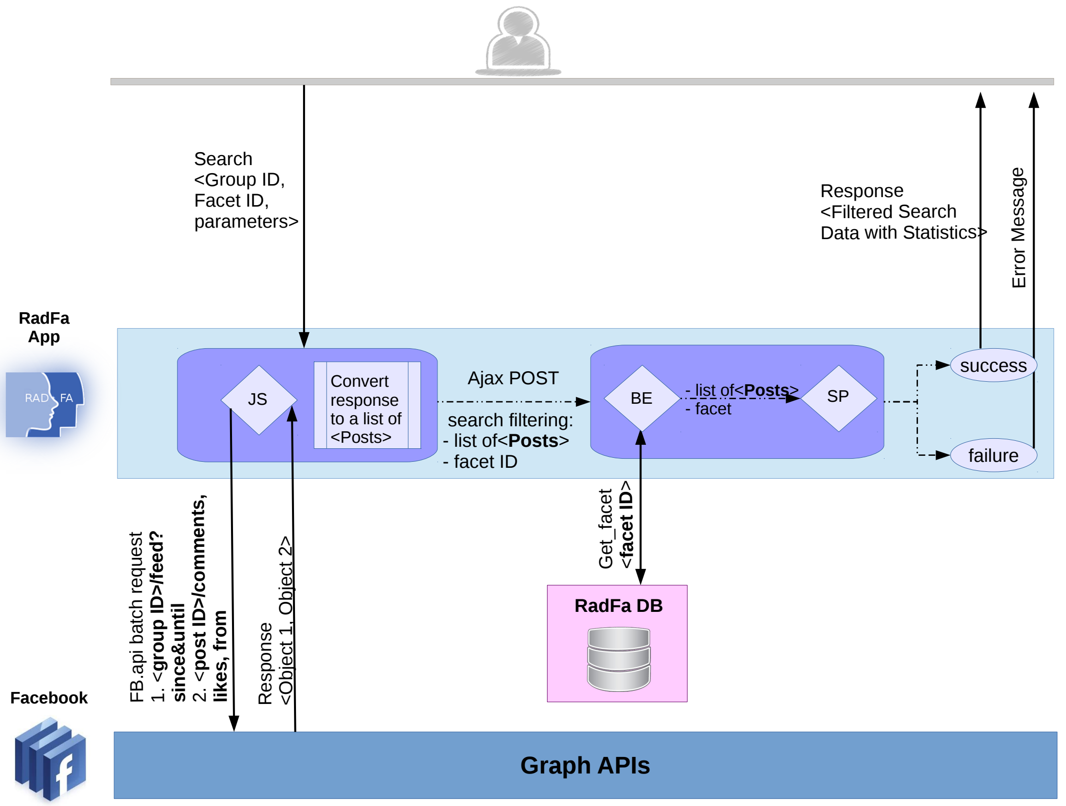
The system has been tested for English and Greek, but there is no reason to believe it would not be appropriate for other languages.

2.2.3 Search

The core functionality of RadFa is the statistical search. The clinical staff can get the statistical results by filling in some fields in the appropriate *Search* form, which is located in the *home page* of the app. The main components for the statistical search is a Facebook group of participants and a stored Facet. During the search process, the group's feed of posts are filtered by the attributes (keywords and phrases) of the selected Facet in a manner that is described in Section 2.3. The user can also fill in some other parameters in the *Search* form, such as the time range of the group's posts that are taken into consideration, whether the comments are included or not etc. The total fields of the *Search* form are described below:

- **Group**: the Facebook group of participants that will be taken into account
- **Facet**: the Facet that will be taken into account
- **From** (optional): the date of the message (post or comment) that will be taken into account
- **To** (optional): the date of the last message (post or comment) that will be taken into account
- **Comments** (optional): whether the comments are included into the feed of posts or not
- **Likes** (optional): whether the likes are included into the feed of posts or not

When the search procedure is completed, the results are displayed in the same web page (*home*) grouped by participant. The clinical staff can observe the number of times each keyword and phrase occur for each participant, as well as the respective message that contributed to the determination of the statistical searching. Moreover, the users can store the statistical results to the history of their accounts.



* JS: Javascript, BE: Back-End, SP: Statistical Procedure

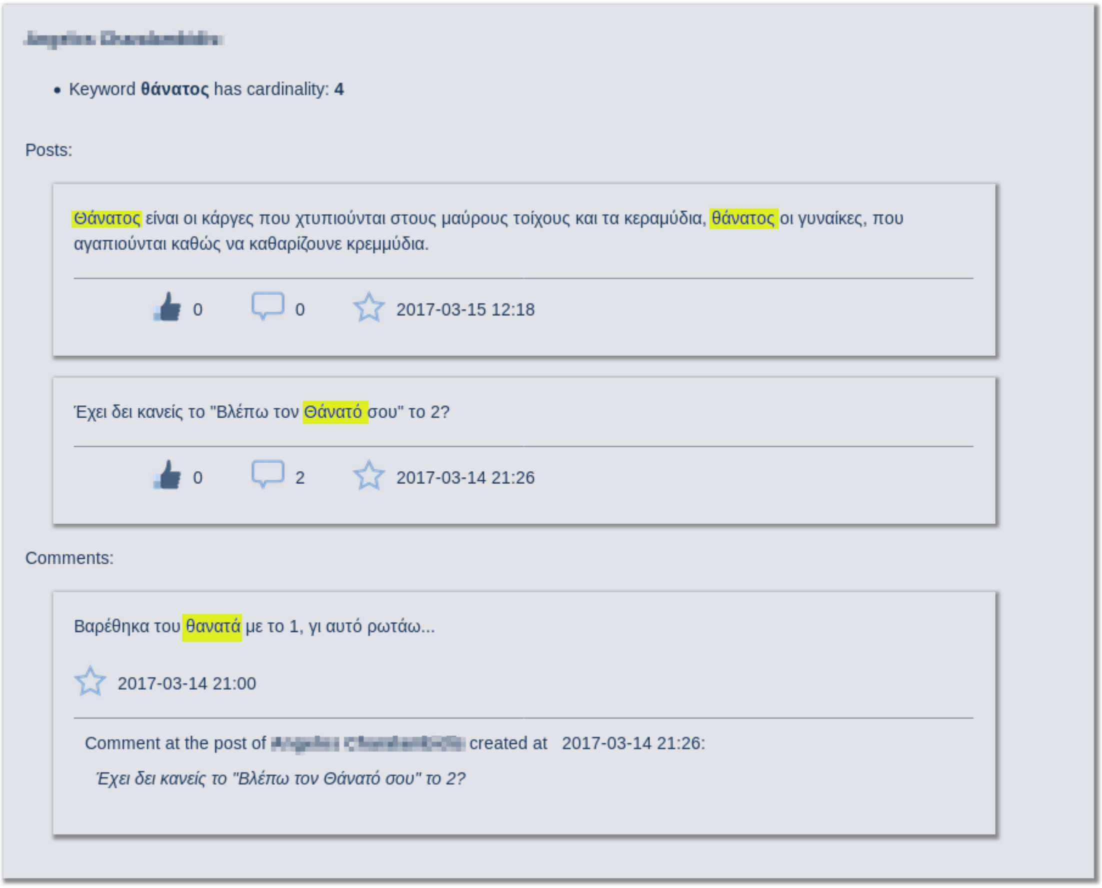
Figure 4: Diagram of the statistical search procedure

Figure 4 shows the diagram of the statistical search in RadFa. As the user calls the statistical search and passes the appropriate variables from the *Search* form, the system makes a batch request to the Graph API as explained in Section 2.1. The parameters set by the user are translated appropriately in the operations of the batch request. The returned response consists of two Objects, each one corresponding to the results of the two operations of the batch request. After the transformation of the response into a single list of objects (list of *Posts*), an AJAX POST request is made in order to filter the returned data. The data will be filtered by a statistical procedure described in Section 2.3 and based on the Facet's attributes. The successful execution of the statistical procedure will return the statistics along with the filtered search data.

2.3 RadFa Statistics

Through the statistical search of RadFa, the clinical staff can collect some discriminative statistics that indicate abstract behaviour features of each participant via their text utterance. In particular, the statistics specify the number of times each participant uses each value of the attributes (keywords and phrases) of a specific Facet during the period of time. Figure 5 shows an example of how the results are displayed in the user's home page after the statistical search. Specifically it depicts the statistic of a single value (ϑάνατος - death) for one participant. As you can notice from the figure, the specific value and its derived words are taken into account for the computation of the total cardinality of the staticitic.

The RadFa statistics are determined from the *statistical procedure*, which takes as input the feed of posts of the respective Facebook group and the Facet. In particular, the statistical procedure is composed of two parts: the *filtering process*, which is a pre-processing step to keep only the feed of posts that are interesting for the analysis (filtered data) ; and the *analysis process* which determines the statistics of each participant based on the filtered data.



[redacted]

- Keyword **θάνατος** has cardinality: **4**

Posts:

Θάνατος είναι οι κάργες που χτυπιούνται στους μαύρους τοίχους και τα κεραμύδια, θάνατος οι γυναίκες, που αγαπιούνται καθώς να καθαρίζουνε κρεμμύδια.

👍 0 💬 0 ☆ 2017-03-15 12:18

Έχει δει κανείς το "Βλέπω τον Θάνατό σου" το 2?

👍 0 💬 2 ☆ 2017-03-14 21:26

Comments:

Βαρέθηκα του θανατά με το 1, γι αυτό ρωτάω...

☆ 2017-03-14 21:00

Comment at the post of [redacted] created at 2017-03-14 21:26:

Έχει δει κανείς το "Βλέπω τον Θάνατό σου" το 2?

Figure 5: Example of the results from the statistical search

In the *filtering process* the text data of the posts are analysed in a level of language processing and is described in the following steps:

1. Given the vocabulary from the keywords of the selected Facet find their synonyms and their derived words (stemming process)
2. Given the group's feed of posts and the results from Step 1, *filter_by_keywords* algorithm is executed
3. Given the phrases of the selected Facet, gather the words of each one and find their synonyms
4. Given the group's feed of posts and the results from Step 3, *filter_by_phrases* algorithm is executed

The *filter_by_keywords* algorithm tokenizes each message in the group and then stems every token. For instance, lets take as an example the message: 'Έχει δει κανείς το Βλέπω το Θάνατό σου 2;' (Has anyone watched I see your death 2?).[10] The system first tokenizes this message as follows:

{ ΕΧΕΙ, ΔΕΙ, ΚΑΝΕΙΣ, ΤΟ, ΒΛΕΠΩ, ΤΟ, ΘΑΝΑΤΟ, ΣΟΥ, 2, ? }

Then it applies the stemming process for the Greek language, therefore the previous token list becomes:

{ ΕΧ-, ΔΕΙ, ΚΑΝΕΙΣ, ΤΟ, ΒΛΕΠΩ, ΤΟ, ΘΑΝΑΤ-, ΣΟΥ }

In addition, it compares each token with each stemming synonym resulted from Step 1. If the two stemmed words are derived from the same root, the respective message belongs to the filtered data. The *filter_by_phrases* algorithm begins with the same process of tokenizing each message, but it does not stems the tokens. Then, it checks whether the synonyms and the tokens are derived from the same root and at the same time if the tokens belong to the same sentence and in raw. Under these conditions the

[10] The actual original title is 'Final destination'

respective message belongs to the filtered data, if it does not already belong there.

Since the filtering process is completed, the analysis process aggregates the filtered data by participant and determines the statistics. The values of each attribute and their stemmed synonyms, are derived and counted for every post of each participant. In addition the total cardinality of each value in the Facet is computed as the sum of its synonyms cardinality in the feed of posts. Finally the the analysis process converts the aggregated filtered data to a JSON object, which is returned as the filtered search data result.

Moreover, the system keeps only the synonyms that have a very similar sense and that are not syntactically far away from the given keyword. As we mentioned in Section 2.2, the language processing is achieved with the use of NLTK, WordNet tools, and English and Greek language stemmers. Specifically the NLTK *tokenize()* method is used for the tokenization of the text data, while in parallel the WordNet corpus returns the synonym of words into different languages. For the stemming pre-process, we use two different stemmers regarding the language - *SnowballStemmer* [11] for English and *Greek Stemmer* [12] for Greek language. It is necessary to point out that the Greek Stemmer takes the Part Of Speech tagging (POS) as an extra input parameter. Therefore, when the Greek language is selected, the filtering process needs an extra pre-processing step where the word is first tagged and then stemmed.

[11] See `https://pypi.python.org/pypi/snowballstemmer`

[12] See `https://bitbucket.org/dataengineering/greek-stemmer`

3 Technical Validation

3.1 Experimental Setup

For the technical validation of the RadFa System we have created two Facebook groups (one for English-language and one for Greek-language posts) with 5 members/participants each. The participants were asked to share their thoughts in English or in Greek, depending on the group. Moreover, we set two simple scenarios:

- **Scenario 1**: The participants should post at the group by taking into consideration the assessment items of interRAI LTCF in D2.2 and by using indicative words or phrases
- **Scenario 2**: The participants should post phrases that contain indicative words from interRAI, but expressed in a figurative way

The total number of textual data collected was 63 posts and 52 comments during 10 weeks. Of these, 33 posts and 49 comments were made in the Greek language group and are taken into account in the experiments below. We apply two different experiments by taking into account items found in Section E *Mood and Behaviour* of the interRAI LTCF[13]. Based on the first assessment item of that section, 'Indicators of possible depressed, anxious, or sad mood', we have created two facets. These correspond to the two categories in the first assessment item that are defined by verbal behaviours, namely:

1a Made negative statements, e.g., Nothing matters; Would rather be dead; What's the use; Regret having lived so long; Let me die

1h Recurrent statements that something terrible is about to happen, e.g., believes he or she is about to die, have a heart attack

Table 2 gives the relevant RadFa facets defined as a proof of concept. Note that the "&" character works as an AND operator between the keywords in the set, i.e. a message (post or comment) belongs to the filtered data if all the keywords of the set are included in it. On the other hand, multiple attributes for the same facet work as an OR operator, i.e., a message belongs to the filtered data when any of the facet attributes match the message. Moreover in both experiments all the posts and comments of the group are taken into account, thus the *Comments* option is selected in the Search form.

3.2 Results and Discussion

Here we give and discuss the results of the above experiments. Table 3 shows the synonyms and the stemming part of each keyword, as well as the phrases with a synonym meaning. Based on those, we evaluate the resulted filtered data of each experiment.

Due to stemming and synonym expansion, retrieval is greatly improved by comparison to simple token

[13] Cf. Section 3.5, D2.2 Early Detection Methods and Relevant System Requirements II

Table 2: Facet attributes

Facet	Attributes	Value	Meaning
interRAI 1a	keyword	θάνατος	death
interRAI 1a	phrase	τίποτα δεν έχει νόημα	nothing matters
interRAI 1h	keyword	δυστυχία	unhappiness
interRAI 1h	keyword	πεθαίνω & αγωνία	to die & anxiety

Table 3: Expansion and examples of facet attributes

Facet	Expansion and stemming	Recognition examples
ϑάνατος (death)	ϑανατ-	ϑάνατος (death), ϑάνατο (death-ACCUSATIVE), ϑανατικό (deadly incident)
τίποτα δεν έχει νόημα (nothing matters)	τίποτα δεν έχει {νόημα,σημασία} (nothing {matters, means anything})	τίποτα δεν έχει σημασία (nothing means anything)
δυστυχία (unhappiness)	δυστυχι-	δυστυχία (unhappiness), δυστυχισμένος (unhappy)
πεϑαίνω (to die)	πεϑαίν-, ψοφ-	πεϑαίνω (die-1-PERS-SG), πεϑαίνουν (die-3-PERS-PL), ψόφος (death-COLLOQUIAL)
αγωνία (anxiety)	αγωνι-	αγωνία (anxiety), αγωνίας (anxiety-GENITIVE), αγωνιώ (worry-1-PERS-SG)

matching. For instance the resulted filtered data from the experiments contain not only the vocabulary item provided by the user, but also words that are derived from the keywords and their synonyms. The same is observed in phrases too. In Table 3 we can see the recognised words and phrases, including cases of non-trivial matching that relies on text processing (stemming) and semantics (synonyms).

On the other hand, there are some situations where precision is not perfect, although this is never as the result of the expansion process above. For example, quoted material (from other people or cultural items such as titles of songs, movies, etc.) is difficult to recognize as such and might be attributed to the poster. We have observed in our posts dataset (but is also a know problem in social media text processing) that quotation marks and similar syntactic indications are often omitted. Human readers can refer to their pragmatic knowledge to understand such references, but a computational analysis that relies on syntax (or even semantics) cannot.

A similar well-known issue is the recognition of figurative expressions that contain matching keywords but under a radically different semantics and the failure to recognize expressions that do not contain matching keywords, but mean something equivalent in the expression's context. Expressions such as 'bored to death' are 'kick the bucket' are examples of expressions that are hard to reject and select, respectively, when searching for references to 'death'.

Both these issues are research topics by themselves in the computational linguistics community and are treated by developing and maintaining extensive knowledge bases that list names and commonly used referring expressions for the entities in the context of discourse (people, places, cultural items, etc.) as well as idiomatic multi-word expressions that have a non-compositional semantics. In our validation experiment, RadFa demonstrates the ability to take into account such resources by:

- Expanding keywords using an external lexical semantics resource such as the WordNet lexical database. Further databases for entity names and other references can be accessed using the same mechanism.
- Recognizing multi-word expressions as a unit attribute, regardless of the semantics of its constituent terms. Databases for idiomatic multi-word expressions can be accessed using the same mechanism.

Given the above, the RadFa System has been technically validated as appropriate to be included in the FZ pilots. The definition of a richer set of Greek-language facets will be done in WP6, in conjunction with the development of the Greek translations for the interRAI assessment items that are relevant to the FZ pilots.[14]

[14] Cf. D6.3 Piloting Plan III