Interactive Coding of Responses to Open-Ended Questions in Russian

Nikita Senderovich(✉) and Archil Maysuradze

Faculty of Computational Mathematics and Cybernetics,
Lomonosov Moscow State University, 2nd Educational Building,
CMC Faculty, MSU, Leninskie Gory, GSP-1, Moscow 119991, Russia
senderovich.nikita@yandex.ru, maysuradze@cs.msu.su

Abstract. We propose an interactive technique to categorize the responses to open-ended questions. The open-ended question requires a response which is a natural language phrase. A typical analysis of the phrases starts with their 'coding', that is, identifying themes of the responses and tagging the responses with the themes they represent. The proposed coding technique is based on interactive cluster analysis. We study theoretically and empirically the hierarchical (agglomerative, divisive) and partitional clustering algorithms to pick the best one for short Russian responses. We address the problem of the short phrase sparseness with thesaurus smoothing. We introduce an iterative process where users can provide some feedback to a clustering result. A domain-oriented system of statements is developed for users' feedback. The system is proved to be able to provide any clusters the user desires. The technique is implemented as a web service for responses in Russian.

Keywords: Open-ended questions · Short text categorization · Interactive clustering · Russian thesaurus

1 Introduction

In last decades qualitative analysis in sociology has become a widely used instrument. Compared to classical quantitative approach, qualitative research is based on unstructured source data. Both approaches are applied in survey analysis, which is commonly used to gather opinion on a wide variety of subjects: governmental agencies conduct surveys to communicate view of different social groups on political, social and economic conditions; commercial structures use survey results to enhance product quality and achieve gain in sales. Two main types of survey questions can be identified by degree of freedom given to the respondent:

1. *closed-ended questions* — questions with a predefined set of answers, respondent has to choose one or more variants
2. *open-ended questions* — questions, where respondent has to write a short textual response in his own words

P. Klinov and D. Mouromtsev (Eds.): KESW 2015, CCIS 518, pp. 195–209, 2015.
DOI: 10.1007/978-3-319-24543-0_15

Opposed to closed-ended questions, open-ended questions cannot be directly analyzed using quantitative methods, their analysis requires more complex approach.

For manual analysis of open-ended responses the following process is commonly used [1,3]:

1. every response is read and a list of main ideas represented in the texts is created
2. the structure of the list created on the previous step is reviewed and refined, short labels (*codes*) are assigned to each item on the list
3. each response is marked with one or several codes according to the contents

This process is called *coding of open-ended question*, it is aimed at converting qualitative information provided by respondents into quantitative form for subsequent analysis. The result of the first two steps is called the *codebook*. The example of coding results can be observed on figure 1 (only a part of result is presented). The question was taken from Sociological Bulletin of Public Opinion Foundation, dated 26 May 2011. The respondents were asked the following question: "Which statements made by Dmitry Medvedev at the press-conference did you memorize and like most?" We see that the set of responses contains the groups "youth policy", "cancellation of vehicle inspection", "innovation, modernization" and "the fight against corruption". For each group the most typical answers are presented.

Что из того, о чём говорил Д. Медведев на пресс-конференции, Вам больше всего запомнилось и понравилось?

Молодёжная политика	«наша молодёжь будет жить лучше»; «о школьниках, студентах»; «Медведев болеет за молодёжь, даёт им работу»; «уделял внимание молодёжи»
Отмена техосмотра	«про техосмотр»; «упрощение системы прохождения осмотров автомобилей»; «он и сказал, что техосмотр теперь будут оформлять не в ГАИ, а при ОСАГО»
Инновации, модернизация	«усовершенствование производства, инновации»; «модернизация»; «надо продолжать процессы модернизации в экономике и политике»; «развитие науки»
Борьба с коррупцией	«о коррупции в рядах чиновников»; «о борьбе с коррупцией»; «реформы надо продолжать и жёстче бороться с коррупцией»; «коррупция»

Fig. 1. Example of open-ended question analysis result. The text on the top is the wording of the question. In the left column of the table the names of the groups discovered (in red) are situated. In the right column several typical answers are presented for each group.

On the one hand, methodology of sociological research lists many cases, when open-ended questions are more appropriate than close-ended [2]. In these cases usage of open-ended questions allows the researcher to approach the inner world of respondent, deeply understand his point of view without imposing boundaries by the strict format of the response. Open-ended questions are known to provide unbiased results, which is especially important when the objects of sociological research are vague and unsystematic views on complex phenomena of social reality. In addition, open-ended questions stimulate respondents to analyze the subject of the questions thoroughly and give more complete answers [1].

On the other hand, the described coding technique induces several serious problems. First, this technique is extremely laborious. Therefore, manual coding is performed by a group of analysts. The following workflow is commonly used:

1. a codebook is compiled according to a subset of answers (this step is usually done by a senior analyst)
2. a full set of responses is divided into parts and categorized by a group of analysts according to the codebook

One problem of this process is that codebook can be incomplete after the first step, so coders have to spend effort to modify it consistently on the second step. Another problem is that both of those steps are prone to subjectivity, which is often used as an argument against the qualitative techniques [2]. Unification of analysis results requires additional time-consuming procedures. The examples of intercoder agreement methods can be found in [4].

To sum up, the cost of manual analysis of open-ended questions is prohibitively high and no proven industry standard for efficient automated analysis is developed by now. That is why open-ended questions are rarely included into the questionnaires, and even if included, then for large volumes of collected data the answers do not get adequate analysis. In these circumstances, efficient automation of coding is a topical problem.

In this paper, we propose a new low-effort workflow for coding open-ended questions based on interaction of analysts with automated system. The rest of the paper is organized as follows. In section 2 related work is considered. In section 3 the system overview is given and proposed workflow is described and discussed in detail. Section 4 contains detailed description of mathematical procedures chosen for each step of data processing in the proposed system. Validation of the system is discussed in section 5. Section 6 concludes the paper.

2 Related Work

Many general-purpose automated systems were proposed to analyze textual information. However, only a few systems take into account specific character of the task considered and can be used to work with open-ended questions.

In the work [5] Japanese authors suggest an automatic method for dividing open-ended responses into 3 classes: positive, negative or request. The method

is based on the use of classification, information provided by analyst is used to build training set.

More general analysis can be performed using IBM SPSS Text Analytics for Surveys. It is a proprietary system, based on a wide variety of prepared domain-oriented linguistic resources: thesauri, lists of synonyms and lists of stop-words. Key feature of the product is work with groups of answers. User can operate with keywords for each group, create subgroups, merge groups and move responses from one group into another. The accent in this product is made on receiving reproducible results and gathering expert opinion to achieve optimal coding. The main problem of the product is that it does not support Russian language.

Several attempts have been made to automatize open-ended response coding in Russian. Systems DISKANT [2] and VEGA [3] were proposed to tackle the challenges of traditional coding methods. Methodologically both of the systems follow the manual process of coding described above, automatizing both codebook creation and codes assignment.

In DISKANT, the representative phrases are chosen by the analyst at the first step of analysis. These phrases define a classifier which is applied on the second step to categorize the responses according to similarities to representative phrases. DISKANT implies an iterative workflow, in which user adds portions of responses and updates classifier until all the data is classified.

VEGA is a newer system based on DISKANT. It provides an automatic procedure that builds groups of responses via comparison of phrases. A dictionary-based semantic analyzer is used to form the groups of relevant responses.

In this paper we present an interactive web-based system for analysis of open-ended questions, which supports a novel workflow oriented on cooperative analysis. The proposed workflow allows for efficient automation at all steps. For each step we selected and implemented appropriate data processing techniques. Analysts perform coding by gradually refining the results proposed by the system rather than grouping the responses manually. The "intellectual core" of the system is text clustering algorithm, which takes into account the opinion of the users. For typical number of responses in a survey the algorithm gives out the result instantly, so the system provides a continuous process of analysis. The system is based on linguistic resources for Russian language.

3 System Overview

As we have discussed above, the key problems of open-ended response coding are laboriousness and subjectivity. In this section we present system which is designed to solve both of the problems simultaneously.

3.1 Interactive Clustering

To deal with the laboriousness we explicitly formulate open-ended coding task as *short texts clustering task with disjoint clusters*. Generally, clustering methods are aimed at building groups of objects (clusters), so that objects in one cluster

are closer to each other than objects in different clusters. In case of text collection it is assumed that the clustering result represents semantic partition of the collection. If open-ended questions are analyzed, we can associate each cluster with a code in a codebook. It should be noted that the result of clustering often can be considered as a classifier, which can be used to process new observations. Similarly, the codebook compiled for one set of responses can be later used to code new portion of data. In this sense all the processes considered above imply solution of clustering problem using specific methods.

In our research we assume that clusters are disjoint, which means that every response contains only one idea. Observation of real data from surveys confirm validity of this assumption: the answers for clearly posed question rarely consist of more than one thought, and even if such responses occur, analyst can always divide them into homogeneous parts on the stage of data preparation.

Many efficient clustering techniques were recently developed by machine learning and information retrieval communities. In section 4.4 the methods most appropriate for text clustering are considered. Important issue we concentrate on is sparseness of source data. Due to brevity of text in vast majority of responses, additional effort must be made to measure similarity between texts adequately.

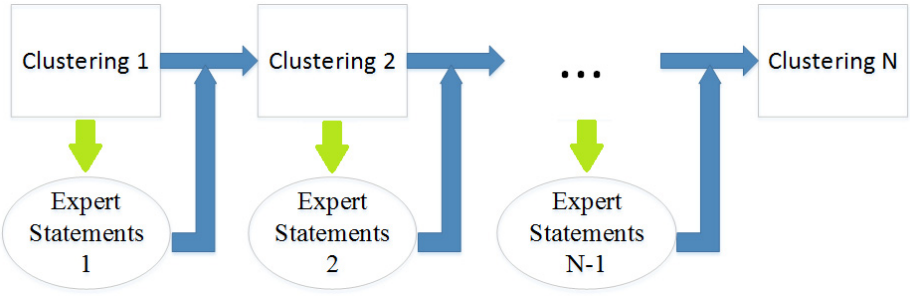

Fig. 2. Interactive clustering process

Usage of clustering procedure is expected to provide efficient automation of basic steps of coding process — creation of the codebook and assigning codes to responses. However, the result of the automatic clustering provided by system can be unsatisfactory. That is why we allow the user to modify the result. We provide analyst with set of instruments, which can be used to express feedback. The statements made by analyst are taken into account by the clustering algorithm. The process is continued until the desired clustering is achieved. The process of gathering analyst's opinion in interactive fashion is shown on figure 2. In the following subsection the set of statements available to user is listed and it is proved, that any clustering can be built with their help. In section 4.4 the statements are formalized for usage in clustering algorithm.

3.2 User Statements

On each step of iterative process the user can alter the clustering using instruments provided by the system. Careful consideration of the situations that can be encountered in analysis of real-world open-ended responses allowed us to develop the interface. The set of statements available to the analyst is the following:

1. select the subset of responses
2. attach selected subset to existing cluster
3. attach selected subset to new cluster
4. detach responses of selected subset from clusters they are attached to
5. withdraw selected subset from consideration
6. complete the formation of cluster
7. continue the formation of cluster
8. remove the cluster

In the beginning of the process all the responses are "free" (*detached*) — they are not attached to any cluster, so on the first step of interactive process the clustering procedure groups responses automatically and provides initial set of clusters. On the next step analyst can fix responses in clusters or move them between clusters using attach statement. After the statements are made by the analyst, the clustering procedure can be launched, which groups all the detached responses automatically.

It should be emphasized that all the statements made by analyst on each step are accumulated, i.e. on the stage of clustering all the statements from previous steps are taken into consideration and all the requirements are satisfied in the resulting clustering. Let us briefly comment on each of the statements.

The first statement supports next 4 operations on the list. Second statement tells the system to move selected responses to another cluster or fix them in current cluster. The third statement leads to increase in number of clusters and can be used to split cluster containing two different ideas. The fourth statement is opposite to the second and the third and makes the responses "free", they can be attached to any cluster by automatic clustering procedure. This operation is necessary if user made a mistake or changed his opinion. The fifth statement is required to remove insignificant responses from the analysis. In most surveys there are respondents, who answer with cliche or off-topic sentences. Statement 5 proposes a way to eliminate such responses. It is important that this statement can be applied to each response only once, that is, it is irreversible.

Last three statements manipulate with clusters. Statement 6 tells the system that the cluster is ready, and automated clustering procedure must not add or remove any responses from it. Statement 7 is opposite to statement 6. The last statement decreases number of clusters and withdraws all the responses of deleted cluster from consideration.

Also an operation of renaming of the clusters is available. This operation does not affect the result and is needed for convenience of analyst.

For proposed system of statements the following theorem can be formulated.

Theorem 1. *Statements 1, 3, 5, 8 allow to achieve arbitrary clustering.*

Proof. We are going to demonstrate, how to achieve any predefined clustering $C_1, \ldots, C_k$. First, using statements 1 and 5 all the insignificant responses are removed. Next, for $i = 1, \ldots, k$ with statements 1 and 3 the desired cluster C_i is built. Finally, all the initial clusters (which are empty by now) are removed using statement 8.

This proof demonstrates an algorithm, in which number of user actions required to build arbitrary clustering depends linearly on the number of analyzed responses, i.e. the complexity of the process is comparable with reading all the responses. However, in average case implemented automatic clustering method allows to reduce amount of user input substantially, which is demonstrated in Section 5.

3.3 Presentation of the Current State

After the primary clustering and every clustering performed in accordance with expert's wishes the data is organized in the following way.

There are two types of clusters:

- *fully formed clusters*: these clusters are stable, their contents cannot be changed by automatic clustering procedure
- *candidate clusters*: these clusters contain both attached and detached responses and can be modified by automatic clustering procedure

Every cluster has a name or a code, representing its main idea.

Also, three types of responses can be identified:

- *attached responses*: these responses were fixed in their clusters by expert, their cluster cannot be changed by automatic clustering procedure
- *detached responses*: cluster of these responses are determined by automatic clustering procedure
- *withdrawn responses*: these responses were considered insignificant by the user and removed from the rest of data set

Every attached and detached response belongs to exactly one cluster. The described structure is presented to analyst on each iteration of coding, he estimates the quality and takes measures to enhance the results.

The coding process is finished, when there are no detached responses left, that is, all the responses are either fixed in a cluster or withdrawn.

3.4 Cooperative Workflow

Interactive system increases productivity of the analyst dramatically and allows to build the desired clustering without tedious procedures of manual codebook compilation. Obviously, single analyst can perform the coding himself, but in this case the issue of subjectivity arises. To speed up the refining of automatically

built clustering and deal with the problem of subjectivity we propose the following cooperative workflow. Several analysts work on open-ended response coding via web interface simultaneously in real time. Each expert can make statements from the list given in section 3.2 and run automatic clustering procedure. All the statements form a growing pool of expertise to be considered by the clustering method at each iteration of the proposed interactive process.

Main advantage of the workflow is that no preliminary intercoder agreement is required. All the experts observe the presentation of current state all the time, and control the coding process. In case if matters of opinion arise during the process, experts are notified by the system, they can discuss the problem, reach an agreement and continue the work. Opposed to traditional approach to coding, the result of the analysis in this case is guaranteed to be consistent and there is no need to spend additional effort to compile the results of individual analysts after they finished their work. Also, participation of several experts guarantees the increase of objectivity of the coding result compared to research performed by an individual analyst, because all the expert decisions are reviewed by other experts, the mistakes can be discovered, discussed and corrected.

It is important to emphasize that the proposed workflow in combination with web-interface of the system provides an opportunity for crowdsourcing, that is, achieving result through feedback given by volunteers. However, we believe that in most cases more adequate coding will be performed by ones of qualified researches working in a group and not by hundreds of people unfamiliar with the subject of research.

4 Data Processing Steps

In course of analyst's work with answers for one open-ended question the data comes through many stages, which must be supported in the system. In following subsections the opportunities and solutions for each step are discussed.

4.1 Data Import

Overview of survey industry standards showed that the raw data received from respondents, is commonly represented on the computer in one (and often for the convenience of researchers in more than one) of the following formats: SPSS, SAS, CSV, Microsoft Excel. Due to the widespread use of the Microsoft Excel format, data import in our system is currently implemented for this format. The proposed web-interface allows to operate with surveys containing multiple questions, and start multiple analyses for each question. For each entity standard CRUD interface is supported and additional text information can be provided by the user.

4.2 Natural Language Processing

The raw data given on the input is a collection of short texts in Russian. Before the responses can be analyzed using clustering techniques, each text undergoes the following processing stages:

– *segmentation*: division of the text into separate sentences
– *tokenization*: division of the text into separate words (terms)
– *normalization*: finding special normal form for each term

The first two stages are technical tasks, they are carried out using regular expressions. However, the normalization stage is a non-trivial and extensively studied recognition problem. Normalization allows to consider forms of the same word as one term, which is especially important for sparse source data. Normalization task can be formulated as a task of building equivalence relation in the set of all terms. One approach widely used for normalization of texts is stemming (see the review [6]). However, for Russian language it is known to show inferior results. In our system we use an approach based on usage of OpenCorpora: open corpus of Russian language, which provides grouping of word forms. Another problem, which requires solution, is the problem of disambiguation. We addressed this issue via part-of-speech tagging.

4.3 Text Model

When natural language processing is finished, the model of each text is built. On this step several models are available: models, based on representing texts as term sequences [7] or usage of frequent itemsets [8]. For our system we chose standard Vector Space Model with binary features. If D — set of responses, W — dictionary of all terms, which occurred in texts of responses, n_{dw} — number of occurrences of term w in document d, then every document d is represented as a vector of length $|W|$ (cardinality of set W):

$$d = [f_1^d, \ldots, f_{|W|}^d]^T, \text{ where } f_w^d = [n_{dw} > 0], \quad d \in \mathbb{R}_+^{|W|} \tag{1}$$

We use binary features instead of commonly used frequencies and TF-IDF values because short responses rarely contain repeating words.

The main issue complicating further processing is the problem of *sparseness*. Collections of short texts are characterized by the lack of statistical information on the occurrence of words and lack of common context information. It makes difficult the selection of adequate similarity measure and building of clustering algorithm. To overcome this difficulty, different approaches are used to enrich the model with additional information. To take into account semantic similarities between terms two sources of additional information are commonly used: set of auxiliary relevant texts and semantic graphs and nets.

In studies [11] and [12], the first approach was successfully used to build thematic models for short texts using Twitter and Wikipedia respectively. In work [9] the output of search engines was used to expand the context of short texts.

In [13], the second approach is applied, lexical database for the English language Word Net is used and proximity matrix P that reflects semantic relations between the terms is built. $P_{ij} \in [0, 1]$ and the bigger value P_{ij} is, the closer in sense the terms i and j are. The ontology was considered as an undirected

graph of semantic connections with terms as vertices and relations hyponym-hyperonym and synonym-synonym as edges. To estimate value P_{ij}, the shortest distance between corresponding vertices in the graph was calculated and inverted.

For our project we used a similar approach and used RuThes — thesaurus of Russian language [10]. Matrix P is obtained using the same idea. In the work [14], links in Wikipedia are used to estimate the proximity between terms, but we believe that general-purpose thesaurus is more appropriate for the task of analyzing open-ended responses, which mainly consist of general vocabulary. In the following subsection the use of matrix P for solving clustering problem is described in detail.

4.4 Text Clustering

Generally, clustering algorithms can produce *partitional* (or flat) clustering or *hierarchical* clustering. Hierarchical clustering implies a tree of nested clusters. The construction of such a tree is also called the task of taxonomy. Unlike hierarchical clustering, flat clustering does not form nested clusters, they are all are on the same level.

Text clustering is a specific clustering task. In this subsection we are going to make a brief overview and analysis of several classical text clustering algorithms and address the issue of source data sparsity for each of them.

Traditional Hierarchical Approaches. The classical approaches to the construction of hierarchical clustering are *agglomerative* and *divisive* clustering algorithms. When building a hierarchical clustering using agglomerative algorithms, objects are gradually merged into bigger clusters. Thus, a partition evolves from the configuration where each object is a separate cluster to the configuration with single cluster containing all objects. When using divisive algorithms, on the contrary, configuration evolves from larger to smaller clusters.

If the distance between each pair of objects is chosen and a formula to recalculate intercluster distances after two clusters are merged is given, then an agglomerative clustering algorithm is fully defined. Indeed, on each step, two closest clusters can be merged, and all the distances from new cluster to other clusters can be recalculated for use on the following steps.

A widely used family of agglomerative clustering is defined by the formula from the famous work of Lance and Williams [15]:

$$d(U \cup V, S) = \alpha_U d(U, S) + \alpha_V d(V, S) + \beta d(U, V) + \gamma |d(U, S) - d(V, S)|. \quad (2)$$

After choosing the values α, β and γ, this formula can be used to recalculate the intercluster distances on the steps of agglomerative algorithm. Examples of widely used special cases of formula (2) are single linkage distance:

$$d(C_i, C_j) = \min_{x \in C_i, y \in C_j} d(x, y) \quad (3)$$

and unweighted average distance:

$$d(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{x \in C_i} \sum_{y \in C_j} d(x, y). \tag{4}$$

To measure the similarity between texts represented as vectors in euclidean space, the vectors are normalized and the cosine measure is used:

$$s(u, v) = (u, v), \qquad ||u|| = ||v|| = 1. \tag{5}$$

Studies [16], [17] analyze a number of agglomerative and divisive algorithms for text clustering (the texts considered are long enough). In these studies clustering with unweighted average distance (UPGMA) shows best results among agglomerative algorithms.

The formula (5) does not take sparsity into account, however, we solve this problem using proximity matrix P:

$$s'(u, v) = \frac{(Lu, Lv)}{||Lu||||Lv||}, \qquad L^T L = P. \tag{6}$$

Here text responses u and v are not necessary of unit length, because the formula provides normalization. The matrix L can be obtained using Cholesky decomposition, because matrix P can always be made positively definite by increasing the numbers on the main diagonal. This addition can be interpreted as increase in significance of term coincidence in responses. It is important to emphasize that similarity (5) is a special case of (6), when $P = I$.

After this modification any hierarchical algorithm that takes distances between objects on the input can be used for short text clustering.

Spherical k-Means. Among partitional clustering algorithms the spherical k-means algorithm [18] is successfully used for text clustering.

Generally, formulation of the problem is the following. We want to partition a sample of n objects $x_1, x_2, \ldots, x_n$ into k clusters. Let us denote with $r_1, r_2, \ldots, r_n$ labels of clusters for each object ($r_j \in \{1, 2, \ldots, k\}$), and with $C_i = \{x_j | r_j = i\}$, $i = 1, 2, \ldots, k$ — sets of objects in clusters. The clusters must be disjoint and contain all the objects:

$$C_i \cap C_j = \varnothing, \ i \neq j, \quad \bigcup_i C_i = \{x_1, x_2, \ldots, x_n\}. \tag{7}$$

The algorithm finds cluster centroids and minimizes the intracluster distances by maximizing the similarity between objects and these centroids and determining values of labels r_j.

Spherical k-means assumes that all the objects belong to unit sphere $x_i \in \mathbb{S}_+$, $\forall i = 1, 2, \ldots, n$, and the following functional with constraints is maximized:

$$\sum_{i=1}^{k} \sum_{x_j \in C_i} (x_j, c_i) \longrightarrow \max_{\boldsymbol{c}, \boldsymbol{r}}, \ ||c_i|| = 1 \tag{8}$$

Formulas for the steps of optimization process are the following:

$$\mu_i = \frac{1}{|C_i|} \sum_{x_j \in C_i} x_j, \qquad c_i = \frac{\mu_i}{||\mu_i||}, \qquad r_j = \underset{i}{\operatorname{argmax}}(x_j, c_i) \tag{9}$$

where c_i — unit vectors, in case of text clustering they are believed to correspond to "central notions" of text collection.

To use semantic smoothing provided by matrix P in spherical k-means the following modification is proposed. Let us use P in optimization problem (8) by changing x_j to y_j, where y_j is defined by formula:

$$y_j = \frac{Px_j}{||Px_j||}. \tag{10}$$

As formula (10) touches only input and guarantees that $||y_j|| = 1$, this change allows to use standard spherical k-means without any modification. Such change can be interpreted as expansion of response texts by adding semantically related terms. Indeed, the construction of matrix P guarantees that vector y_j will have additional positive components, which correspond to the terms that are close in sense to the original terms of the response x_j.

Experimental Evaluation. In this section the experimental evaluation of 4 different short text clustering methods discussed above is performed. UPGMA and single linkage clustering represent agglomerative clustering, DIANA (DIvisive ANAlysis clustering) represents divisive algorithms, spherical k-means represents partitional methods.

Model data sets M1–M4 of increasing complexity simulating real-world data were prepared. Our data generation procedure allows to control size and structure of the sample: number of responses N, number of clusters k, number of keywords in each cluster (real-world clusters are usually formed around 1-10 keywords), number of general vocabulary words common for all clusters. We also control different levels of noise, which make the sample more complex: amount of general vocabulary in responses, amount of keywords of other clusters in responses. Along with the responses, corresponding proximity matrix P is generated. The noise level for matrix P—amount of semantically related words appearing in different clusters—is chosen as well.

The characteristics of data sets and clustering algorithms performance can be found in table 1. For model data the sample sizes were chosen as common numbers of respondents participating in the survey (several hundreds). The quality is measured using standard F-measure, which we can calculate because we know the true classification of objects. If we denote by n number of objects clustered, by n_i size of i-th ground true class, by n_j size of j-th found cluster, by n_{ij} number of objects of class i in cluster j, then the precision, recall and F-measure of correspondence of cluster j for class i can be calculated according to following formulas:

$$P(i,j) = \frac{n_{ij}}{n_j}, \; R(i,j) = \frac{n_{ij}}{n_i}, \; F(i,j) = \frac{2R(i,j)P(i,j)}{R(i,j) + P(i,j)}. \tag{11}$$

Table 1. Text clustering algorithms performance on model data sets of increasing complexity, F-measure

			Method	UPGMA		SL		DIANA		SKM	
N	k	Complexity factors	Smoothing	no	yes	no	yes	no	yes	no	yes
300	5	none	**M1**	1.00	**1.00**	1.00	**1.00**	1.00	**1.00**	1.00	**1.00**
500	10	+general vocab.	**M2**	0.89	0.90	0.78	0.87	0.66	0.93	1.00	**1.00**
1000	15	+common keywords	**M3**	0.61	0.75	0.35	0.53	0.48	0.56	0.79	**0.81**
1000	15	+proximity noise	**M4**	0.61	0.67	0.35	0.47	0.48	0.57	0.79	**0.80**

The F-measure of overall correspondence between classes and clusters for partitional algorithm:

$$F_p = \sum_i \frac{n_i}{n} \max_j F(i,j) \tag{12}$$

For hierarchical clustering we follow [16] and for each class calculate correspondence to each cluster in the cluster tree T and select maximum:

$$F_h = \sum_i \frac{n_i}{n} \max_{j \in T} F(i,j) \tag{13}$$

As we can see from Table 1, as expected, the quality decreases with the increase of data set complexity and noise levels (addition of proximity noise in experiments without smoothing does not decrease the quality as no proximity matrix is used). In all cases the proposed methods of smoothing allowed to achieve significant gain in quality. Spherical k-means demonstrated the best results among clustering methods compared. This algorithm was chosen for the system and adapted for interactive clustering, as shown below.

Formalization of User Statements. In our system for open-ended responses coding we use spherical k-means as a basic clustering algorithm. Below we show, how user statements announced in 3.2 are taken into account by the automated clustering algorithm.

Let us denote detached responses with $x_1, x_2, \ldots, x_l$, and attached responses with $x_{l+1}, x_{l+2}, \ldots, x_{l+q}$. All the statements made by experts modify the contents of these two sets, labels r_j for each attached response and number of candidate clusters. To take these modifications into account we use a semi-supervised approach. If we have k candidate clusters in the response, then formulas (9) take the form:

$$\mu_i = \frac{1}{|C_i|} \sum_{x_j \in C_i} x_j, \;\; c_i = \frac{\mu_i}{||\mu_i||}, \;\; i \in \{1, \ldots, k\} \tag{14}$$

$$r_j = \operatorname*{argmax}_i (x_j, c_i), \;\; j \in \{l+1, \ldots, l+q\} \tag{15}$$

These formulas show, how to assign detached responses to candidate clusters taking into account the information about attached objects provided by experts.

5 System Validation

In this section the results of experiments with real-world survey data are presented. Performance evaluation was conducted in order to ensure that implemented system meets the requirements of the domain and provides efficient reduction of analyst's effort in the process of coding.

Responses to three open-ended survey questions (Q1–Q3) were coded by an expert using the implemented web service. In course of the work the information about user's interaction with the system was gathered. In particular, we were interested in amount of effort required to achieve the clustering that satisfied the expert. To measure laboriousness of the process we use total number of mouse clicks and total number of iterations of the interactive process described in section 3.1, i.e. number of automatically built clusterings. The clicks were counted only in the process of analysis, the clicks made for authorization in the system and data import were not included. The data sets contain different number of responses and different number of topics touched by respondents. The latter number can be estimated as number of clusters in the final data partition. Table 2 contains information about data sets and results of the experiments.

Table 2. Metrics of open-ended questions coding in the system

Data Set	Responses	Clicks	Iterations	Clusters
Q1	43	17	4	3
Q2	125	49	15	10
Q3	727	130	27	17

As we can see from the results, the number of clicks is significantly less than the number of responses, which indicates the utility of the proposed system compared to manual coding.

6 Conclusion

In this paper we have reviewed the domain of open-ended response coding and identified problems researchers face in course of their work. We proposed a novel workflow based on interactive computer-aided process which aims to increase the objectivity and reduce the laboriousness of research. For each step of data processing possible variants were analyzed and appropriate solutions were developed based on empirical results and needs of the domain considered. Although much more research must be carried out before the state-of-the-art system for open-ended response coding for Russian language is developed, we are optimistic that our contribution will serve as a basis for further studies.

Acknowledgments. The reported study was partially supported by the Russian Foundation for Basic Research (RFBR), research projects No. 13-01-00751a and No. 15-07-09214a.

References

1. Geger, A. E.: The use of open-ended questions in the measurement of value orientations. In: Sociology Yesterday, Today, Tomorrow: 2-nd Sociological Readings in Memory of V.Golofast, pp. 48–60. SPb (2008)
2. Saganenko, G.I.: A comparison of non-comparable: study of comparative research on the basis of open-ended questions. Sociological Journal **3-4**, 144–156 (1998)
3. Boyarsky, K.K., Kanevsky, E.A., Saganenko, G.I.: On the issue of automatic text classification. Economic-mathematical studies: mathematical models and information technology, 253–273(2009)
4. Carey, J.W., Morgan, M., Oxtoby, M.J.: Intercoder agreement in analysis of responses to open-ended interview questions: Examples from tuberculosis research. Cultural anthropology methods **8**(3), 1–5 (1996)
5. Sakurai, S., Orihara, R.: Analysis of Textual Data based on multiple 2-class Classification Models. International Journal of Computational Intelligence **4**(4) (2008)
6. Jivani, A.G.: A comparative study of stemming algorithms. Int. J. Comp. Tech. Appl. **2**(6), 1930–1938 (2011)
7. Zamir, O., Etzioni, O.: Web document clustering: a feasibility demonstration. In: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 46–54. ACM (1998)
8. Fung, B.C., Wang, K., Ester, M.: Hierarchical document clustering using frequent itemsets. SDM **3**, 59–70 (2003)
9. Sahami, M., Heilman, T.D.: A web-based kernel function for measuring the similarity of short text snippets. In: Proceedings of the 15th International Conference on World Wide Web, 377–386. ACM (2006)
10. Loukashevich, N.V.: Thesauri in problems of information retrieval. Moscow University Printing House (2011)
11. Hong, L., Davison, B.D.: Empirical study of topic modeling in twitter. In: Proceedings of the First Workshop on Social Media Analytics, pp. 80–88. ACM (2010)
12. Jin, O., Liu, N.N., Zhao, K., Yu, Y., Yang, Q.: Transferring topical knowledge from auxiliary long texts for short text clustering. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, pp. 775–784. ACM (2011)
13. Siolas, G., d'Alché-Buc, F.: Support vector machines based on a semantic kernel for text categorization. In: Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks, IJCNN 2000, vol. 5, pp. 205–209. IEEE (2000)
14. Varlamov, M. I., Korshunov A. V.: Computing semantic similarity of concepts using shortest paths in Wikipedia link graph. JMLDA, 1107–1125 (2014)
15. Lance, G.N., Williams, W.T.: A general theory of classificatory sorting strategies 1 Hierarchical systems. The Computer Journal **9**(4), 373–380 (1967)
16. Zhao, Y., Karypis, G.: Evaluation of hierarchical clustering algorithms for document datasets. In: Proceedings of the Eleventh International Conference on Information and Knowledge Management, pp. 515–524. ACM (2002)
17. Steinbach, M., Karypis, G., Kumar, V.: A comparison of document clustering techniques. KDD workshop on text mining **400**(1), 525–526 (2000)
18. Buchta, C., Kober, M., Feinerer, I., Hornik, K.: Spherical k-means clustering. Journal of Statistical Software **50**(10), 1–22 (2012)