CLASSIFICATION OF PRIMARY MEDICAL RECORDS WITH RUBRYX-2: FIRST EXPERIENCE

Olga Kaurova, Mikhail Alexandrov, Ales Bourek

Abstract: *RUBRYX is a document classifier developed in 2000s for processing large volumes of Web information. RUBRYX uses weighted sum of n-grams (n=1,2,3) extracted from a very limited number of samples (about 5-10) and takes into account their mutual position in a given text. This sophisticated algorithm proves to be very effective in classifying primary medical records presented in a free text form. In the paper we study possibilities of RUBRYX (version 2.2) on a limited document set in Spanish. These documents are medical histories related to stomach diseases. Such area should be considered as a narrow subset of medical records. The high quality of archived results (accuracy 80%-90%) allows us to recommend RUBRYX for similar applications.*

Keywords: *natural language processing, medical diagnostics, document classification*

ACM Classification Keywords: *I.2.7 Natural Language Processing*

Introduction

1.1 Problem setting

The subject under consideration is classification of primary medical records presented in a free text form as usually produced by medical professionals. Each document used here is related to a certain disease. So, in this case the medical record classification can be considered as a means of document based medical diagnosis decision support. The solution of this problem allows:

- To monitor medical doctors responsible for primary medical observation and to help reduce medical errors;
- To facilitate data exchange between different medical centers and to coordinate the storage and retrieval of individual records with the aid of computers;
- To help form Internet communities with similar health issues areas of interests.

The first significant publications in the addressed area appeared almost 20 years ago. The authors used Bayesian classifiers for processing encounter notes [Aronov, 1995a;

Aronov, 1995b]. An interesting and comprehensive work was published in 2006. It demonstrated that in spite of the use of advanced algorithms of classification, such as the SVM, the results prove to be not so good [Rost, 2006]. We assume that this can be explained by a weak application of lexical resources to the documents under consideration. In a recent publication [Zhang, 2010] the authors use structural patterns in encounter notes, which allowed to improve the results. A short review concerning classification of free text clinical narratives was published last year [Kaurova, 2011]. It contains a description of some medical corpora, methods and software tools. The results described in this review led us to find and test new algorithms in order to improve existing results without the need for additional extraordinary efforts.

In the paper we study possibilities of the document classifier RUBRYX to process such specific documents as primary medical records. In the experiments we use the last version of the mentioned program. The version 2.2 is free shareware and can be easy downloaded [Rubrix, http]. The document set includes 55 documents related to 6 stomach diseases. It allows for analyzing the results of experiments in detail.

The RUBRYX algorithm uses patterns in the form of one-word terms, bigrams and trigrams and takes into account their joint position in a document. Currently are not aware of any publications describing medical records classification by RUBRYX. The above mentioned circumstances define the objectives of our work.

1.2 State of the art

Classification procedures are traditionally included into the technologies of Machine Learning and Data Mining. Well-known resources [Mitchell,1997; Bishop, 2006] provide good theoretical basis for the area. Document classification is covered in the books [Baeza,1999; Manning, 1999]. Here special attention is given to document indexing – the transformation of free text documents into their numerical form. A recent example of a text book containing many algorithms of document classification is [Manning, 2009].

There are many software packages on the market related to Machine Learning and Data Mining, for example: Weka [Weka, http], Rapid Miner [RapidMiner, http], CLUTO [CLUTO, http], R [R, http]. Some of these have ad-hocs for working with textual documents [WekaText, http; RText, http]. These ad-hocs use very simple procedures of text indexing that can not, and do not, give satisfactory results.

The program RUBRYX was developed in 2000s [Polyakov, 2001; Polyakov, 2003]. This program proved to be very friendly for end-users because of its simplicity in training and tuning. RUBRYX demonstrated its advantages on the famous set of Reuter news. Namely it provided the *F*-measure of 86% with only 5 representatives from each of 10 categories used for training. Other algorithms could reach the levels of 75%-92% of *F*-measure using dozens of documents for training. One should mention here an excellent work where these results are shown [Stein, 2003a].

In section 2 we describe lexical resources and classification algorithm of RUBRYX. In section 3 we present the corpus used in our study. In section 4 we present the results of experiments. Short discussion is provided in section 5. Section 6 contains conclusions.

RUBRYX description

2.1 Training (preprocessing)

We present RUBRYX description because we could not find it in literature. Hereinafter we use the following terminology. By 'mini-vocabulary' we mean a vocabulary related with a concrete category. These mini-vocabularies are created during the training stage. By 'terminological vocabulary' we mean a vocabulary created for a given domain by external experts. This vocabulary reflects a common terminology for all categories in a document corpus. Terminological vocabulary is not obligatory for RUBRYX functionality. RUBRYX can create its mini-vocabularies with the support of terminological vocabulary as well as without it.

Both mini-vocabulary of a given category and a common terminological vocabulary contain 3 lists:

- one-word terms
- two-word terms (bigrams)
- three-word terms (trigrams)

To create mini-vocabularies a user selects several of the most representative documents from each category. Let us have *M* documents related with a certain category. The procedure consists in the following:

- All stop terms are eliminated
- All common one-word terms form the first list in the file WordList
- All common bigrams form the second list in the file WordLst2
- All common trigrams form the third list in the file WordLst3

Speaking 'common' we mean terms which occur at least in m documents, here $m \le M$. In our experiments we set m=M. The terminological vocabulary is an additional filter for term selection. Namely, RUBRYX selects those terms from WordList, WordLst2, and WordLst3, which occur also in the terminological dictionary.

The procedure presented above is implemented for all n categories. Therefore if we have n categories then the result of preprocessing will be 3n lists of terms.

Stop terms have their own vocabulary. This vocabulary consists of 3 lists with one-word terms, bigrams and trigrams respectively. The titles of files are fixed as: BlackList, BlackList2, BlackList3. Unlike the mentioned mini-vocabularies and terminological vocabulary the black lists can use so-called 'regular expressions' [Expressions, http]. For example, '?' and '??' mean all words with one or two letters. The expression '*[0-9]*' means all words, which contain at least one number. Etc.

2.2 Classification (processing)

Algorithm

RUBRYX uses the well known lineal algorithm: it calculates contribution of each category to a given document as a linear combination of category indexes [Baeza, 1999; Manning, 2009]. In our case the indexes are terms from the mini-vocabularies. Then the category having the largest contribution is announced to be a winner. Here is the short description

Let j be the number of category; $\{L_{j1}, L_{j2}, L_{j3}\}$ be the numbers of terms from all three lists in a given document; $\{N_{j1}, N_{j2}, N_{j3}\}$ be the numbers of all one-word terms, bigrams and three-grams in a given document. The contribution of j-category is:

$$C_j = K_1 (L_{j1}/N_{j1}) + K_2 (L_{j2}/N_{j2}) + K_3 (L_{j3}/N_{j3})$$

where K_1+K_2+K_3 =1. Obviously, that C_j = [0, 1] for all categories

RUBRYX developers set the following values for K_i : K_1=(0.2)/3, K_2= (1.3)/3, and K_3= (1.5)/3. It is easy to see that $\sum_i K_i = 1$. These values were determined empirically on the basis of numerous experiments of the authors with different document sets. We use the same values in our research

Modifications

1) Thresholds for category selection

Traditional algorithm uses the following rule for decision making in case of hard classification:

the category j is a winner if $C_j=max_i (C_i)$.

RUBRYX uses a more complex rule, which takes into account the results of training. Namely, let T_j be a threshold for j-category. That is:

- a document belongs to j-category if $C_j \geq T_j$
- a document does not belong to j-category if $C_j < T_j$

But what to do when we have more than one satisfied condition, let for example, $C_1 \geq T_1$, $C_2 \geq T_2$. In this case the rule of decision making considers the values $\lambda_1= C_1 - T_1$, $\lambda_1= C_2 - T_2$. The category having the maximum λ-value will be the winner: $\lambda_j=max_i (\lambda_i)$

If all thresholds are too high and we have not even one satisfied condition $C_j \geq T_j$ then a given document can not be classified.

The thresholds { $T_1, T_2, \ldots T_n$ } are calculated during the training stage as a result of optimization problem solution. Namely, the best values of T_j provide the minimum number of errors when we classify the training document set.

2) Taking into account the term positions in a document

The developers suppose that terms better support their category when they are located together. For this reason the developers increase the weights of close terms in a document on a certain value *p.* Speaking of 'weights' we mean coefficients K_1, K_2, and K_3 introduced above. Speaking of 'close terms' we mean the simultaneous term occurrences in a given window. The developers fixed the window size S=10. Parameter p is an algorithm parameter, we set p=0.3. It is the advice of the developers. This value can be easy changed by a user.

2.3 Tuning

Having obtained the initial results of classification a user can change parameters of the algorithm to improve these results. The principal parameters to be changed are the thresholds T_j. Let we deal with *j*-category and let '*j*-documents' mean the documents from this category selected by RUBRYX.

Case 1: There are a *small* number of *j*-documents and a *small* number of alien documents.

> We decrease the threshold T_j that allows to increase the number of *j*-documents.
>
> The number of alien documents is expected not to increase in the same proportion.

Case 2: There are a *small* number of *j*-documents and a *large* number of alien documents.

> The situation is undefined. One should 'play' with the thresholds including the threshold T_j

Case 3: There are a *large* number of *j*-documents and a *small* number of alien documents.

> It is just what we want and we do nothing

Case 4: There are a *large* number of *j*-documents and a *large* number of alien documents.

> We increase the threshold T_j that allows to decrease the number of alien documents.
>
> The number of *j*-documents is expected not to decrease in the same proportion.

We use these rules in our experiments to tune RUBRYX

Experimental material

3.1 Corpus under consideration and its lexical resources

The corpus for this study is a collection of 55 anonymous primary medical records from one Clinical Hospital. The records are related to gastrointestinal diseases. Each of these records contains a short description of chief complaint, past medical history (including major illnesses, any previous surgery/operations any current ongoing illness, e.g. diabetes)., family history, medications, allergies, objective status and finally two diagnoses, the principal one and the concomitant one (morbidity and co-morbidity). Texts belong to 6 classes – diseases. The corpus is described in Table 3.1. Appendix presents an example of a primary medical record.

Table 3.1 Categories presented at the corpus

Class	***Disease***	***Number of texts***	***Number of words in all texts***	***Number of different words in all texts***
1	gallbladder disease	12	2849	428
2	mechanical jaundice	8	2076	458
3	stomach cancer	11	2873	572
4	acute appendicitis	6	1339	245
5	gastrointestinal bleeding	7	1525	373
6	inguinal hernia	11	2396	243
Total		55	12828	1269

3.2 Terminological vocabulary

Terminological vocabulary is not the obligatory element for RUBRYX work. But in many cases it can improve the quality of mini-vocabularies and as a consequence of the quality of classification.

The terminological vocabulary was constructed by external expert. It was the surgery related with the corpus of medical records described above. To construct the vocabulary we used the following technology:

<u>Step 1</u>.

All specific one-word terms were extracted from the whole corpus and presented to the expert. The expert selected the most interesting terms from this list. Here we used the program LexisTerm [Lopez, 2011].

<u>Step 2</u>.

All collocations (the left and right ones) with the selected terms were extracted from the corpus and presented to the expert again. He corrected the list and formed the final list of one-word terms, bigrams and trigrams. Here we used an auxiliary program.

For reasons of clarity we include following a brief comment concerning the program LexisTerm we used on the Step 1. For this we give two definitions [Lopez, 2011]:

Definition 1. The general lexis is a frequency word list based on a certain corpus of texts.

The 'certain corpus' means here any standard document set reflecting the lexical richness of a given language. Generally such a corpus contains in a certain proportion the documents taken from newspapers, scientific publications related with various domains, novels and stories. In our case it was the British National corpus.

Definition 2. The level of specificity of a given word **w** in a given document corpus C is a number $K \geq 1$, which shows how much its frequency in the document corpus $f_C(\mathbf{w})$ exceeds its frequency in the general lexis $f_L(\mathbf{w})$:

$K = f_C(\mathbf{w}) / f_L(\mathbf{w})$

Obviously, the more *K* is, the less words appear in the resulting list. We tested LexisTerm with K= {5,10, 20, 50,100}. From the expert point of view the value K=10 proved to be the best one.

Experiments

4.1 Measures for results evaluation

To evaluate the quality of classification we use several well-known measures being popular in Information Retrieval [Baeza, 1999]. Here is a short review of these measures:

Let a classifier selects *l* documents from the existing *m* documents to be selected, and let there are *k* really corrected documents between these *n.* In this case we can calculate Precision (*P*) , Recall (*R*), and *F*-measure using the formulae: $P= k/l$, $R= k/m$, $F=2PR / (P+R)$

Obviously these formulae refer to a binary classification. When we deal with several categories one should use the combined *F*-measure proposed in [Stein, 2003b]. We can use here also the traditional Accuracy: $A= n/N$, where *n* is a number of all correct classified documents, *N* is a number of all documents. Good survey of measures used in classification problems is presented in [Pinto, 2008].

4.2 Sensitivity to thresholds

Hereinafter we will use the following terminology. By 'class' we mean the Gold Standard. 'Category' is a result of classification with RUBRYX.

Table 4.1 Classification, *K* are calculated automatically

Class	***Training set***	***Test set***	***K=***	***1***	***2***	***3***	***4***	***5***	***6***	***Correct docs***
1	5	7	27	**1**	5			1		1
2	5	3	24		**3**					3
3	5	6	23		1	**5**				5
4	5	1	37				**1**			1
5	5	2	27					**2**		2
6	5	6	35						**6**	6
Total	30	25		1	9	5	1	3	6	18

Table 4.1 shows that Category 1 practically does not contain the documents from Class 1. According the rules described in the section 2.3 we decrease the threshold K_1 by 25%. Now it is K_1 =21. The results are presented in Table 4.2. We have here *Accuracy* = (18/25)*100% = 75%

Table 4.2 Classification, K_1 = 21 (25% decreased)

Class	***Training set***	***Test set***	***K=***	***1***	***2***	***3***	***4***	***5***	***6***	***Correct docs***
1	5	7	21	**7**						7
2	5	3	24		**3**					3
3	5	6	23		1	**5**				5
4	5	1	37				**1**			1
5	5	2	27					**2**		2
6	5	6	35	6					**0**	0
Total	30	25		13	4	5	1	2	0	18

In this experiment we use mini-vocabularies constructed on the basis of 5 samples *without a terminological vocabulary*. The results are presented in Table 4.1. Table rows contain the distribution of documents from a given class between categories. Table columns contain the distribution of all documents assigned to a given category between classes.

The accuracy is calculated as a ratio of the number of correct cases to the total number of cases. It is easy to see that *Accuracy* = (18/25)*100% = 75%

Table 4.2 shows that Category 6 is empty and Category 1 contains all documents of Classes 1 and 6. We have to go back to the original value of K_1=27 and to increase the threshold K_2 by 25%. Now it is K_2 =30. The latter was done to filter the documents of Class 1, which were put to Category 2 with K_1=27, see Table 4.1. The results of experiment are given in Table 4.3. We have now *Accuracy* = (23/25)*100% = 92%.

Table 4.3 Classification, K_2 = 30 (25% increased)

Class	***Training set***	***Test set***	***K=***	***1***	***2***	***3***	***4***	***5***	***6***	***Correct docs***
1	5	7	27	**5**		1		1		5
2	5	3	30		**3**					3
3	5	6	23			**6**				6
4	5	1	37				**1**			1
5	5	2	27					**2**		2
6	5	6	35						**6**	6
Total	30	25		5	3	7	1	3	6	23

As the result of the last experiment was successful we decide to complete the last experiment with K_1=24 and K_2=30. The results are given in Table 4.4. For this case *Accuracy* = (23/25)*100% = 92%.

Table 4.4 Classification, K_1=24 (10% decreased), K_2=30 (25% increased)

Class	***Training set***	***Test set***	***K=***	***1***	***2***	***3***	***4***	***5***	***6***	***Correct docs***
1	5	7	24	**6**				1		6
2	5	3	30		**3**					3
3	5	6	23			**6**				6
4	5	1	37				**1**			1
5	5	2	27					**2**		2
6	5	6	35	1					**5**	5
Total	30	25		7	3	6	1	3	5	23

4.3 Sensitivity to terminological vocabulary

Here we test the sensitivity of results to application of terminological vocabulary. Basically, we repeat the first experiment of the previous series of experiments, but now we use the vocabulary on the stage of training. The results are presented in Table 4.5. Here we have *Accuracy* = (21/25)*100% = 84%

Table 4.5 Classification with terminological vocabulary, *K* are calculated automatically

Class	*Training set*	*Test set*	*K=*	*1*	*2*	*3*	*4*	*5*	*6*	*Correct docs*
1	5	7	22	**5**	2					5
2	5	3	16		**3**					3
3	5	6	10			**5**		1		5
4	5	1	23				**0**	1		0
5	5	2	5					**2**		2
6	5	6	33						**6**	6
Total	30	25		5	5	5	0	4	6	21

Table 4.5 shows that the accuracies for Categories 2 and 5 are low, namely 60% for Category 2 and 50% for Category 5. So, we increase the thresholds for these categories by 25% and round it up in a higher number. We have now K_2= 20 and K_5 = 7. The results are given in Table 4.6. It is seen that *Accuracy* = (24/25)*100% = 96%.

Table 4.6 Classification with terminological vocabulary, K_2=20 and K_5=7 (both increased by 25%)

Class	*Training set*	*Test set*	*K=*	*1*	*2*	*3*	*4*	*5*	*6*	*Correct docs*
1	5	7	22	**7**						7
2	5	3	20		**3**					3
3	5	6	10			**6**				6
4	5	1	23				**0**	1		0
5	5	2	7					**2**		2
6	5	6	33						**6**	6
Total	30	25		7	3	6	0	3	6	24

4.4 Sensitivity to size of training set

For this series of experiments we use the most numerous classes. According to the Table 3.1 it is classes 1, 3 and 6. With these classes we can train and test RUBRYX on the largest number of samples. In the first experiment we use 3 documents for training from each class and in the second experiment we use 6 documents from each class. Naturally, the test set is the same in both cases. Terminological vocabulary is not used. *K*-values are calculated automatically. The results of classification are presented in Table 4.7 and 4.8 respectively.

To evaluate the quality of these results we use two measures: accuracy and combined *F*-measure. For the first experiment we have *Accuracy* = 75%, *F-measure* = 73%. For the second experiment we have *Accuracy* = 88%, *F-measure*=87%

Table 4.7 Classification with training set of 3 samples

Class	*Training set*	*Test set*	*K*=	*1*	*3*	*6*	*Correct docs*
1	3	6	32	**2**	4		2
3	3	5	30		**5**		5
6	3	5	42			**5**	5
Total	9	16		2	9	5	12

Table 4.8 Classification with training set of 6 samples

Class	*Training set*	*Test set*	*K*=	*1*	*3*	*6*	*Correct*
1	6	6	25	**4**	2		4
3	6	5	22		**5**		5
6	6	5	34			**5**	5
Total	18	16		4	7	5	14

Discussion

The potential possibilities of classification - any classification - are defined by relations between categories. The closer their characteristics are, the lower level of results we obtain. In case of document classification the closeness between categories is mainly defined by the intersection of lexis related with each category. When this intersection is absent the quality of classification is the highest one: we can avoid any errors. But when lexical resources of categories are similar then we can expect many errors. These extreme cases present so-called wide domain and narrow domain with respect to categories, which compose this domain. The problem of classification of narrow domain corpora was considered in detail in the dissertation [Pinto, 2008].

In the present paper we deal with a relatively narrow domain: the intersection of lexis between some categories is 16%-25%. The results of measurements are presented in Table 5.1.

Table 5.1 Intersection of lexis

	Categories 1-2	*Categories 2-3*	*Categories 3-4*	*Categories 4-5*	*Categories 5-6*
Common words	23%	23%	16%	25%	19%

The experiments show that RUBRYX can easily cope with the difficulties caused by the mentioned intersection of lexis. The results we obtained are enough good and they exceed those obtained early in [Catena, 2008].

Conclusions

In the paper we tested the document classifier RUBRYX on a limited set of primary medical records. This set can be considered as a relatively narrow domain collection. We studied the sensitivity of classification results to threshold variations, use of terminological vocabulary and size of training set.

The experiments show that

- RUBRYX is easy tuned automatically and manually on a given corpus that allows to reach high results
- one word terms, bigrams and trigrams taken together and also taking into account their mutual position in a document allow to process narrow domain collections

In future we plan to combine the pre-processing procedure of RUBRYX with other classifiers such as Naïve Bayes, SVM, etc.

Acknowledgement

The authors are very appreciated to Mr. Vladimir Sinitsyn, one of RUBRYX developers, for his numerous consultations and additional software tools he offered for our work

Bibliography

[Aronow, 1995a] D.B.Aronow, J.R.Cooley, S.Soderland. Automated identification of episodes of asthma exacerbation for quality measurement in a computer-based medical record. In: Proc. of Annual Symposium on Computer Applications in Medical Care. 309-13, USA, 1995.

[Aronow, 1995b] D.B.Aronow, et.al. Automated classification of encounter notes in a computer based medical record. In: Proc. of MEDINFO '95 8th World Congress on Medical Informatics, Medinfo, Canada, p. 8-12, 1995.

[Baeza, 1999] R. Baeza-Yates, B. Ribeiro-Neta. Modern Information Retrieval. Addison Wesley, 1999

[Bishop, 2006] C. Bishop. Pattern Recognition and Machine Learning, Springer, 2006

[Catena, 2008] A.Catena, M.Alexandrov, B.Alexandrov, M.Demenkova. NLP-Tools Try To Make Medical Diagnosis. In: Proc. of the 1-st Intern. Workshop on Social Networking (SoNet-2008), Skalica, Slovakia, 2008.

[CLUTO, http] CLUTO: http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview

[Expressions, http] Expressions: http://en.wikipedia.org/wiki/Regular_expression

[Kaurova, 2011] E. Kaurova, M. Alexandrov, X. Blanco. Classification of free text clinical narratives (short review). In: Scient.Book “Information Science and Computing”, Publ. House ITHEA, 2011, 12 pp.

[Lopez, 2011] R.Lopez, M.Alexandrov, D,Barreda, J.Tejada. Proc. of the 4-th Intern. Conf.on Intelligent Information and and Engineering Systems (INFOS-2011), Publ. House ITHEA, Poland, 2011, 8 pp.

[Manning, 1999] C.D.Manning, H.Schutze, Foundations of statistical natural language processing. MIT Press, 1999

[Manning, 2009] C.D.Manning, H.Schutze, Introduction to Information Retrieval. Cambridge, 2009

[Mitchell, 1997] T.Mitchell. Machine Learning, McGrow Hill, 1997

[Pinto, 2008] D. Pinto, On Clustering and Evaluation of Narrow Domain Short-Text Corpora. Doctoral Dissertation, Polytechnic University of Valencia, Spain, 2008

[Polyakov, 2001] V. Polyakov, V. Sinitsyn. Method for automatic classification of web-resource by patterns in text processing and cognitive technologies. In: Text Collection, No.6, Publ. House Otechestvo, p. 120-126, 2001 (rus.)

[Polyakov, 2003] V.Polyakov, V. Sinitsyn. RUBRYX: technology of text classification using lexical meaning based approach. In: Proc. of Intern. Conf. Speech and Computing (SPECOM-2003), Moscow, MSLU, p. 137-143, 2003

[R, http] R-project: http:// www.r-project.org

[RText, http] R-project: http://cran.es.r-project.org/web/views/NaturalLanguageProcessing.html

[RapidMiner, http] RapidMiner: http://rapid-i.com

[Røst, 2006] T.B.Røst, O.Nytro, A.Grimsmo. Classifying Encouter Notes in the Primary Care Patient Record. In: Proc. of the 3-rd Intern. Workshop on Text-Based Information Retrieval (TIR-06). Univ. Press, p. 5-9, 2006.

[Rubryx, http] Rubryx: http://www.sowsoft.com/rubryx/rubryx2.zip

[Stein, 2003a] B.Stein,S.M.Eissen. AISearch: Category Formation of Web Search, www.aisearch.de, 2003

[Stein, 2003b] B.Stein, S.M.Eissen, F. Wissbrock. On Cluster Validity and the Information Need of Users. In: Proceedings of the 3rd IASTED International Conference on Artificial Intelligence and Applications (AIA'03), Benalmadena, Spain, Acta Press, pp.216-221, 2003

[Weka, http] Weka: http://www.cs.waikato.ac.nz/ml/weka/

[WekaText, http] Weka: http://stackoverflow.com/questions/7213125/building-running-a-streaming-weka-text-classifer-in-java

[Zhang, 2010] J.Zhang, Y.Gu, W.Liu, T.Zhao, X.Mu, W.Hu. Automatic Patient Search for Breast Cancer Clinical Trials Using Free-Text Medical Reports'. In: Proc. of the 1-st ACM International Health Informatics Symposium. USA., 2010.

Appendix

An example of primary medical record (in Spanish)

Quejas:dolores permanentes sordos en la zona iliaca.

Anamnesis de la enfermedad: El paciente se sintio enfermo hacia las 15 horas 10.02.07, cuando surgieron los dolores sordos vagos en el mesogastrio, nauseas, escalofrios. Despues de algunas horas los dolores se extendieron a la zona iliaca derecha. La ausencia de mejora hizo que llamara para pedir ayuda medica. Se traslado con sospecha de apendicitis aguda al hospital KB¹119 por el servicio de ambulancias.

Anamnesis de vida. Ha crecido y se ha desarrollado con normalidad. No hay enfermedades heredadas.

Enfermedades sufridas: No habia traumas u operaciones. Rechaza la anamnesis ulcerosa y cardial.

Anamnesis de alergia: No es relevante.

Diagnostico objetivo: estado de gravedad media. La epidermis es de color normal, humeda. La temperatura del cuerpo es de 37.2 C. La hemodinamica es estable. Pulso - 84. Presion arterial – 130/80 mm. La respiracion se realiza llenando por completo los pulmones. Frecuencia respiratoria 16 por minuto. La lengua es seca, con la placa blanca. El vientre es suave, indoloro en la zona iliaca derecha. Los sintomas de Rovzing, Sytkovskiy, Bartomie-Mihelson son positivos. No hay sintomatologia de peritonitis. La palpacion en la zona de cintura no produce dolor. No hay dificultades en la evacuacion urinaria y defecacion.

Se prescribe hospitalizacion en el area quirurgica e intervencion quirurgica urgente.

Authors' Information

Olga Kaurova *– Saint Petersburg State University (Department of Theoretical and Applied Linguistics - graduated in 2009); Autonomous University of Barcelona (International Master in "Natural Language Processing & Human Language Technology" - graduated in 2010; PhD program "Lenguas y Culturas Románicas" - current), 08193 Bellaterra (Barcelona), Spain;*

e-mail: kaurovskiy@gmail.com

Major Fields of Scientific Research: document classification, sentiment analysis

Mikhail Alexandrov *– Professor of the Academy of national economy and civil service under the President of Russia; Prosp. Vernadskogo 82, bld. 1, Moscow, 119571, Russia; research fellow of the fLexSem Research Group, Autonomous University of Barcelona, 08193 Bellaterra (Barcelona), Spain;*

e-mail: MAlexandrov@mail.ru

Major Fields of Scientific Research: data mining, text mining, mathematical modelling

Ales Bourek *– Senior lecturer, Masaryk University, Brno, Czech Republic; Head of Center for Healthcare Quality, Masaryk University. Kamenice 126/3, 62500 Brno, CZ. e-mail: bourek@med.muni.cz*

Major Fields of Interest: reproductive medicine – gynecology, health informatics, healthcare quality improvement, health systems