Computational Linguistics and Intellectual Technologies:
Proceedings of the International Conference “Dialogue 2017”

Moscow, May 31—June 3, 2017

DOMAIN-INDEPENDENT CLASSIFICATION OF AUTOMATIC SPEECH RECOGNITION TEXTS

Mescheryakova E. I. (e-meshch@yandex.ru),
Nesterenko L. V. (lyu.klimenchenko@gmail.com)

National Research University Higher School of Economics; DC-Systems, Moscow, Russia

Call centers receive large amounts of incoming calls. The calls are being regularly processed by the analytical system, which helps people automatically inspect all the data. Such system demands a classification module that can determine the topic of conversation for each call. Due to high costs of manual annotation, the input for this module is the automatically transcribed calls. Hence, the texts (=automatic transcription) used for classification contain ill-transcribed words which can probably influence the classification process. Another important point is that this module also has special requirements: it should be domain-independent and easy to setup. Document classification task always requires an annotated data set for classifier training, but it seems to be too costly to make an annotated training set for each domain manually. In this paper, we propose an approach to automatic speech recognition texts classification that allows the user avoiding full manual annotation and at the same time to control its quality.

Key words: document classification, document clustering, automatic speech recognition, noisy texts processing

ТЕМАТИЧЕСКАЯ КЛАССИФИКАЦИЯ АВТОМАТИЧЕСКИ ТРАНСКРИБИРОВАННЫХ ТЕКСТОВ ЛЮБОЙ ПРЕДМЕТНОЙ ОБЛАСТИ

Мещерякова Е. И. (e-meshch@yandex.ru),
Нестеренко Л. В. (lyu.klimenchenko@gmail.com)

Национальный исследовательский университет Высшая школа Экономики; DC-Systems, Москва, Россия

1. Introduction

Customer contact centers or call centers have to deal with a large amount of calls every day, and it appears to be difficult to manage this data and do the analytics manually. The supervisors and managers of call centers are interested in getting detailed analytic reports on a daily basis, which means it should not demand much human involvement and should be done quickly, in other words, it should be done automatically.

Getting the information about popular topics of incoming calls requires an analytic system to have an option of text classification/clustering (in our case the automatically transcribed calls recordings). Some solutions for this task had been proposed in [Agarwal et. al. 2007], [Popova et al. 2014], but the problem of domain-independent classification remains open. Here we propose an approach to domain-independent automatic speech recognition (ASR) texts classification. Our approach to handling noisy Russian data (texts with ASR mistakes) outdoes the one proposed in [Popova et al. 2014] and the use of clustering for semi-automatic training set annotation seems to be a solution to domain-independent classification.

The paper is organized as follows. In section 2 we make an overview of some papers devoted to clustering and classification of short ASR texts. Next, in section 3 we describe special characteristics of the data—the ASR texts. In sections 4 and 5 we describe the pipeline we implemented for domain-independent ASR texts classification and present the results of evaluation. Section 6 contains the conclusions of our work.

2. Related Work

While short noisy texts like social media content are a hot topic in NLP nowadays [Subramaniam 2009], ASR texts do not receive much attention. The ASR systems performance is stated to be high; however, when applied to call-center data, ASR quality decreases because of the system's sensitivity to loud environment and low-quality equipment. Obviously, it results in more errors for the languages with rich morphology like Russian.

In [Agarwal et. al. 2007], besides an overview of types of noise in textual data and related NLP tasks, one can find a number of experiments describing how ASR mistakes affect the supervised classification results (SVM vs. multinomial naive Bayes, English Reuters texts). The observations are optimistic: with word error rate up to 40%, a classifier accuracy does not decrease significantly. In [Popova et al. 2014] authors compare manual text transcripts and automatically recognized texts (word error rate about 20–35%) clustering and make the same conclusions as suggested in [Agarwal et. al. 2007] about the effect of error rate on the clustering results. It is also claimed that stop words (manually gathered domain-specific list) removal and Latent Semantic Indexing improve clustering results (best-averaged result stated is F1-score=0.47 on k-means with a stop-list when LSI is not used). Another work [Popova, Krivosheeva, Korenevsky 2014] proposes a more sophisticated approach to automatic stop words list generation: a word is included in the stop list if its removal from all the documents increases the dissimilarity between documents related to different clusters and also decreases the dissimilarity between documents within the same cluster. The best averaged F-measure achieved is 0.57.

ASR texts processing is usually done via supervised classification or clustering into a fixed number of clusters. The former approach demands a large manually annotated collection and the latter usually requires determining the number of clusters by a human or a robust procedure of finding one. A possible workaround is a two-stage clustering method [Wang, Wu, Shao 2014] where hierarchical clustering is performed in a sliding window and the clusters are iteratively merged using the information gain measure.

The approach we are proposing here is both simple and effective. Clustering allows us to avoid full manual annotation and at the same time to control the annotation quality.

3. Data

For the experiments, we used the dataset of 1,370 automatically transcribed calls of an airlines call center (all texts are in Russian). In the following, we refer to automatically transcribed calls as *texts* or *documents*. The dataset was manually annotated according to 5 topics, which are *luggage, booking change, ticket return, flight status,* and *flight information* (see Table 1 for the distribution of the topics).

Table 1. Topics distribution in the collection

Topic	Documents
luggage	653
booking	288
return	257
status	74
flight info	98
Total	1,370

These texts are typically short (min=18, max=1,439, median=170 words) and contain mistakes of the ASR system. The proportion of incorrectly transcribed word forms in ASR results is typically about 10–40% depending on audio quality. Below we refer to the words that were incorrectly transcribed by the ASR system as *noise*. Table 2 shows some examples of noisy sentences.

Table 2. ASR transcription errors. The erroneous words are in **bold**

	ASR transcription examples	Correct transcription
1	spasibo za **nogti konja** (thanks for **the horse nails**)	spasibo za zvonok vsego dobrogo do svidanija (thanks for the call and good bye)
2	davajte **kot** bronirovanija vam nazovu let me tell you the booking **cat**	davajte **kod** bronirovanija vam nazovu let me tell you the booking **code**
3	nazovite nomer **brone** pozhalujsta tell me your **bookings** code please	nazovite nomer **broni** pozhalujsta (tell me your **booking** code please)

	ASR transcription examples	Correct transcription
4	**drova zazhiganija** firewood ignition	**spasibo za ozhidanie** (thanks for waiting)
5	broni **junosheskogo truda** skazhet booking **teenage labour** say	broni ? ? skazhet/skazhite booking ? ? say
6	mne by na popozzhe **rjabina** for me a bit later **ashberry**	mne by na popozzhe ? for me a bit later ?

As one can see, there are two major problems here. The first one is words deletion, and we do not deal with it in this paper. The second one is incorrect word transcription, and we see it useful to distinguish between two types of such mistakes. First, the mistakes (like examples 4–6 in Table 2), for which one can hardly name the correct word form or find any regularity in its appearance in texts. The other type of ASR mistakes (like examples 1–4 in Table 2) are the words that seem to be very similar to the correct transcription and they always stand for the same original words or, to put it differently, they are regular. In Section 4.2 we discuss how different ways of text vectorization allow us to cope with such noise.

Another salient characteristic of our data is its dynamics. A typical call-center that we are dealing with gets thousands of calls every day, and all of them have to be categorized. The distribution of topics can change in time depending on many external factors, with new topics appearing and some of the old ones vanishing. That means, we can not train a classifier once and be satisfied with the result. The fact that the data we deal with can change significantly obliges us to keep our classifier up-to-date and retrain it when needed.

When we apply the approach described below to the data we get for a new call-center project, at the starting point we do not know whether it is related to bank industry, telecommunications or any other domain. Quick setup for a new call-center project is also desirable and it should not involve time-consuming gathering or/and editing keywords lists.

To sum up, all the characteristics of data determine requirements to our classification module: resistance to ASR errors, timely model re-training, domain independence, and quick setup.

4. Implementation: From Clustering to Classification

4.1. Texts preprocessing and vectorization

For the purposes of quick setup, our pipeline demands minimal preprocessing of the ASR results. Firstly, the texts are lemmatized[1] and the stop words are deleted. We use a standard stop words list consisting of highly frequent Russian words (such as functional words and pronouns) and an additional wordlist containing words

[1] We used Mystem morphological analyzer [Segalovich 2003].

typical to call-centers (e.g. *talk, speak, please, hello* etc.; we found that for our purposes 60 words are enough). This list does not include domain-specific words and can be applied to various contact centers; this reduces customization costs. We also do not try to filter or correct ASR errors as most of them are being filtered automatically during the document vectorization procedure.

Normalized texts are then vectorized via one of the usual NLP techniques: tf-idf [Pedregosa et al. 2011] or doc2vec [Mikolov et al. 2013]. In order to compare different ways of vectorization, we performed classification using Random Forest Classifier (RFC, [Breiman 2011]), Logistic Regression [Hosmer et al. 2013] and SVM Classifier [Steinwart, Christmann 2008] on the same dataset vectorized by tf-idf, doc2vec distributed memory and doc2vec distributed bag-of-words models (Table 3). During the cross-validation procedure, training and test document sets were vectorized by tf-idf separately from each other on each iteration. When building a tf-idf collection matrix, the following the document frequency thresholds appeared to be the optimal: a word was not included in the tf-idf vocabulary if it was found in less than 20 documents or more than 50% of the collection. As for the optimal doc2vec model parameters, we finally set the vector size to 400 and the word frequency threshold to 3, i.e. a word that is ignored if it occurs less than 3 times in the collection.

Table 3. Different classifiers performance with tf-idf and doc2vec vectorization

Classifier, vectorization	F1-score
RFC (100 trees), tf-idf(max_df=0.5, min_df=20)	0.85
Logistic Regression (C=1), tf-idf(max_df=0.5, min_df=20)	0.86
SVM (C=1), tf-idf(max_df=0.5, min_df=20)	0.84
RFC (100 trees), doc2vec (size=400, min_count=3, distributed memory)	0.65
RFC (100 trees), doc2vec (size=400, min_count=3, bag of words)	0.62
Logistic Regression (C=1), doc2vec (size=400, min_count=3, distributed memory)	0.43
Logistic Regression (C=1), doc2vec (size=400, min_count=3, bag of words)	0.31
SVM (C=1), doc2vec (size=400, min_count=3, distributed memory)	0.30
SVM (C=1), doc2vec (size=400, min_count=3, bag of words)	0.31

The experiments had shown that tf-idf approach is preferred over the doc2vec models. We chose tf-idf model for the sake of its good performance, simplicity, and interpretability. Firstly, this helps to ignore most ASR mistakes during the classification as their document frequency is usually below the threshold (examples of the filtered words are given in Table 4); secondly, ASR mistakes defined above (see Section 3) as regular mistakes, if frequent enough to be in tf-idf vocabulary, are supposed to improve clustering to some extent.

Table 4. Stop words filtered by their document frequencies. These are obviously non-regular ASR errors

Stop words	English translation
file, veselit, razdelno, globus, izlechit, travmaticheskij, programmka, paluba, arest, lishaj	fillet, to amuse, separately, globe, to cure, traumatic, programme, deck, arrest, shingles

4.2. Clustering and clusters merging

Despite the fact that nowadays one has a large number of well-known clustering methods to choose from, the main challenge is still to determine the optimal number of clusters for a dataset. The problem is usually solved by optimization techniques such as elbow method, silhouette method, etc. However, we can not stick to one criterion as we try to make a domain-independent classification module. On the one hand, we want to avoid human involvement when possible, on the other hand, however, it is desirable to have an option that allows to edit clustering results if necessary. We solve this problem in the following way: the documents are clustered into deliberately larger, than it presumably is, number of clusters, so that their homogeneity is certainly high, and then the clusters are merged according to their lexical similarity. The merging procedure can be done or supervised by a human.

After the K-means clustering procedure (k-means++ initialization, 15 clusters), the average cluster homogeneity was 0.61, which we found acceptable. Adjusted rand index, on the other hand, was only 0.18, and completeness = 0.31.

Every cluster can be described by a list of most frequent lemmas bigrams (Table 5). Clusters merging procedure is therefore quite trivial and stands on these lists pairwise similarity. We refer to the named sets of clusters that this procedure results in as calls topics. The calls topics were named taking into account the manual annotation labels set. We make this assumption in order to perform the final evaluation in terms of the classification problem.

As shown in Table 5, the lists of the bigrams do not include much noise. Because of the high quality of these wordlists, it becomes possible for a person to adjust the clustering results and/or to name the calls topics if necessary.

Table 5. Most frequent lemmas bigrams of clusters

cluster id	bigrams	calls topic
# 0	*Russian* salon samolet, summa izmerenie, sem'desyat santimetr, santimetr summa, damskij sumochka, ruchnoj klast', damskij sumka, sumka noutbuk, sto pyatdesyat, dopolnitel'nyj plata, bagazh kilogramm *English translation* plane cabin, summ dimension, seventy centimeter, centimeter summ, lady's bag, hand luggage, women's bag, luggage kilograms	luggage
# 1	*Russian* moskva ekaterinburg, nol' nol', vylet nol', predstavitel' aviakompanija, izmenit sorok, dar'ja predstavitel', nol' izmenit', ekaterinburg vylet, tridcat' utro, nol' utro, moskva vylet *English translation* moscow ekaterinburg, zero zero, departure zero, airline representative, change forty, darja (name) representative, zero change, ekaterinburg departure, thirty morning, zero morning, moscow departure	flight status
# 2	*Russian* bagazhnyj otdelenije, salon samoljot, bagazhnyj otsek, bagazh salon, bagazh kilogramm, sto pyatdesyat, summa izmerenie, provoz bagazh, sdat' bagazh, besplatno norma, pyatdesyat santimetr *English translation* luggage section, plane cabin, luggage place, luggage cabin, luggage kilogram, hundred fifty, sum dimension, carriage luggage, claim luggage, free norm, fifty centimeter	luggage
# 5	*Russian* data vylet, izmenit' data, kupit' bilet, tysjacha rubl', familija passazhir, vylet vylet, bilet izmenit', vylet napravlenie, annulirovat' bilet, novyj bilet, denezhnyj sredstvo, nevozvratnyj tarif *English translation* departure date, change date, buy ticket, thousand ruble, surname passenger, departure departure, booking change, departure direction, cancel booking, new booking, money, economy class	booking change

4.3. Classification of the new documents

The procedure described above yields a large decently annotated collection of documents that can be used as a training set for the further classification. The classifier (best results were shown by Logistic Regression with C = 25)[2] is trained on the clustering results and predicts cluster ids for the new documents. Then these labels are mapped to the calls topic names according to the clusters naming done at the previous stage, and, finally, these results are compared to the manual annotation (Table 1). We evaluated the classifier's performance on the same dataset (Table 1) by dividing it into the training set, which was used for clustering, and the test set.

Table 6 shows the best classifier performance for each topic. The overall result—weighted average precision, recall and F-measure—is given in Table 7. The weighting was performed according to the proportion of 'true' instances of the particular topic class.

Table 6. Logistic Regression evaluation

Topic	Precision	Recall	F1-score	Number of documents
luggage	0.96	0.90	0.93	125
booking change	0.83	0.37	0.51	65
ticket return	0.48	0.90	0.63	52
flight status	0.58	0.69	0.63	16
flight information	0.73	0.50	0.59	16

Table 7. Weighted performance measures

Weighted Precision	Weighted Recall	Weighted F1
0.80	0.74	0.74

As shown in Table 6, the largest calls class ('luggage') was classified very well. We explain this by low lexical similarity of these documents with the others. On the other hand, 'flight status' and 'flight information' are rather often confused, and we see their closeness as the reason for high FP rate of the former and FN rate of the latter. The overall results seem satisfactory given that we did not edit the results of clustering. In comparison to the supervised classification results (Table 3 for the tf-idf vectorization), where F1 achieved 0.86, the results of the classifier trained on semi-automatically annotated dataset are slightly lower but still adequate.

5. Conclusion

In this paper, we observed the problem of domain-independent classification of automatic speech recognition texts and proposed a solution that allows to avoid fully manual annotation of the documents collection. Our results show that using clustering techniques as an automatic training set annotation tool does not worsen the classification results greatly. We regard the described pipeline as an acceptable solution for the case when one cannot afford manual annotation of a large training set.

---

[2] We used TPOT Python module [Olson et al. 2016] to chose the optimal classifier configuration.

References

1. *Agarwal, Sumeet, et al.* (2007), How much noise is too much: A study in automatic text classification. Data Mining ICDM 2007. Seventh IEEE International Conference on. IEEE.
2. *Breiman, Leo* (2001), Random Forests. Machine Learning. 45 (1), pp. 5–32.
3. *Halko, Nathan, Per-Gunnar Martinsson, and Joel A. Tropp.* (2011), Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. SIAM review 53.2, pp. 217–288.
4. *Hosmer Jr., David W., Stanley Lemeshow, and Rodney X. Sturdivant* (2013), Applied logistic regression. Vol. 398. John Wiley & Sons.
5. *Mikolov, Tomas, et al.* (2013), Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems.
6. *Olson R. S., Urbanowicz R. J., Andrews P. S., Lavender N. A., La Creis Kidd, and Jason H. Moore* (2016), Automating biomedical data science through tree-based pipeline optimization. Applications of Evolutionary Computation, pages 123–137.
7. *Pedregosa et al.* (2011), Scikit-learn: Machine Learning in Python. JMLR 12, pp. 2825–2830.
8. *Popova S. et al.* (2014), Automatic speech recognition texts clustering. International Conference on Text, Speech, and Dialogue. — Springer International Publishing, pp. 489–498.
9. *Popova S., Krivosheeva T., Korenevsky M.* (2014), Automatic Stop List Generation for Clustering Recognition Results of Call Center Recordings. International Conference on Speech and Computer. Springer International Publishing, pp. 137–144.
10. *Rosenberg, Andrew, and Julia Hirschberg* (2007), V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure. EMNLP-CoNLL. Vol. 7.
11. *Steinwart, Ingo, and Andreas Christmann.* (2008), Support vector machines. Springer Science & Business Media.
12. *Subramaniam L. V. et al.* (2009), A survey of types of text noise and techniques to handle noisy text. Proceedings of The Third Workshop on Analytics for Noisy Unstructured Text Data. — ACM, pp. 115–122.
13. *Wang Y., Wu L., Shao H.* (2014), Clusters merging method for short texts clustering. Open Journal of Social Sciences. Vol. 2. — № 09, p. 186.
14. *Segalovich, Ilya.* (2003), A Fast Morphological Algorithm with Unknown Word Guessing Induced by a Dictionary for a Web Search Engine.