EPiC Series in Language and Linguistics
Volume 4, 2019, Pages 61–69
Proceedings of Third Workshop "Computational linguistics and language science"

Defining discourse formulae: computational approach*

Ekaterina Gerasimenko[1], Svetlana Puzhaeva[1], Elena Zakharova[1], and Ekaterina Rakhilina[1,2]

[1] National Research University Higher School of Economics, Moscow, Russia
katgerasimenko@gmail.com, syupuzhaeva@gmail.com,
1583253@gmail.com, rakhilina@gmail.com
[2] Vinogradov Institute of Russian Language, Russian Academy of Sciences, Moscow, Russia

Abstract

In this paper, we address the problem of automatic extraction of discourse formulae. By discourse formulae (DF) we mean a special type of constructions at the discourse level, which have a fixed form and serve as a typical response in the dialogue. Unlike traditional constructions [4, 5, 6], they do not contain variables within the sequence; their slots can be found in the left-hand or right-hand statements of the speech act. We have developed the system that extracts DF from drama texts. We have compared token-based and clause-based approaches and found the latter performing better. The clause-based model involves a uniform weight vote of four classifiers and currently shows the precision of 0.30 and the recall of 0.73 (F1-score 0.42).The created module was used to extract a list of DF from 420 drama texts of XIX-XXI centuries [1, 7]. The final list contains 3000 DF, 1800 of which are unique. Further development of the project includes enhancing the module by extracting left context features and applying other models, as well as exploring what DF concept looks like in other languages.

Keywords: Construction Grammar, discourse formulae, machine learning, natural language processing, entity extraction.

1 Introduction

In the 20th century, a new integrated approach called discourse analysis [2, 3] appeared. It implied studying the language during communication in a specific social situation. The increasing interest in colloquial form of discourse resulted in the fact that conversational analysis became a separate field of knowledge. Scientists started to pay attention to the structure of dialogues and conversations. Sacks, Schegloff and Jefferson [14] put forward the idea that the dialogue consists of adjacency pairs whose elements depend on each other. In other words, utterance A causes the emergence of utterance B following A. Subsequently, the dialogue structure turned out to be much more complicated than it had been thought to be initially, but the idea that there is an inventory of fixed utterances used in conversations was appealing, and these fixed

*The study was supported by grant of the Russian Science Foundation (project No. 16-18-02071)

G. Wohlgenannt, R. von Waldenfels, S. Toldova, E. Rakhilina, D. Paperno, O. Lyashevskaya, N. Loukachevitch, S.O. Kuznetsov, O. Kultepina, D. Ilvovsky, B. Galitsky, E. Artemova and E. Bolshakova (eds.), CLLS 2018 (EPiC Series in Language and Linguistics, vol. 4), pp. 61–69

utterances were called formulaic sequences. The range of formulaic sequences is extremely wide, and Jackendoff [9] shows that formulaic language is almost equal in size to the lexicon of separate words.

We have to say that the diversity of formulaic utterances caused the increase in the number of terminological expressions, adding the new ones such as *chunks*, *collocations*, *formulaic speech*, *multiword units*, *ready-made utterances*, etc. [15]. Several types of such utterances, such as idioms and sayings, have been thoroughly described earlier, but others still need to be defined.

In this paper, we propose a new view on the origin of a particular type of formulaic sequences - discourse formulae. We define such utterances as a special type of constructions, which, unlike the traditional ones [4, 5, 6], do not have variables within them — their slots can be found in left-side or right-side statements of the speech act: *Ničego sebe!* (lit.: 'Nothing to oneself'), *Nu i čto?* (lit.: 'Come on and what?'), *Esče by!* (lit. 'more'), etc.

DF have a fixed form and serve as typical responses in the dialogue. We propose that discourse formulae are constructions at the discourse level. The inner form of this term can be explained by the discourse function of providing coherence in the dialogue, on the one hand, and by its fixed formulaic nature, on the other.

Although we can find the description of the cognate items in other approaches, we have observed that the class of DF is restricted and does not have a comparable equivalent in other linguistic theories. To describe DF, we need a representational list of such utterances for Russian language. The main features of DF are frequency of occurrence, fixed intonation pattern, and non-compositionality. DF predominantly include conjunctions, particles, and counterwords, and they represent a particular step in grammaticalization [8]. Their dependence on the previous speech act implies that they appear mostly in the beginning of the conversational turn. All these factors make DF available for automatic extraction.

2 Automatic DF extraction

In order to get a comprehensive list of DF, we implemented a system for automatic DF extraction. In the present paper, we compare two approaches to automatic DF extraction: clause-based and token-based models.

The quality is measured with the following metrics: precision, recall, and F1-score (a harmonic mean between precision and recall). In this case, recall is more important because it is required that the compiled list contain as many potential DF as possible, whereas false positives can be filtered out manually. We do not utilize accuracy because of a considerable class disbalance (only around 2% of tokens and pseudoclauses belong to DF class).

We trained both systems on 34 drama texts and tested them on 3 drama texts. For grammatical annotation of texts, we used package pymorphy2 for Python [11]. For classification, we used the classifiers implemented in the scikit-learn package for Python [13].

All the texts were manually annotated: one part of the corpus was annotated by a group of annotators and then validated by the main annotator, and the other part of the texts was annotated solely by the main annotator. We considered as DF only those utterances that satisfy the following conditions:

1. The remark is a response to the verbal stimulus;
2. The isolated usage of the utterance is possible (as checked in RNC);
3. The utterance is not syntactically related to the rest of the statement;
4. The utterance does not function as a linking expression or as a reaction to one's own statement;

5. The remark predominantly consists of functional words, which increases the relevance of its pragmatic function;
6. The remark does not contain notional words that are semantically close to the words in the left context;
7. The sequence of remarks, separated by the punctuation mark other than final, is considered to be the connection of several DF if requirements 1-4 are met;
8. A remark can be classified as a DF if it is not located at the absolute beginning of the conversational turn but satisfies the requirements 1-4.

2.1 Token-based model

In the token-based model, the DF is extracted word by word by predicting the tag for each token. Here we include punctuation marks as tokens, since we believe that punctuation both in the context and in a DF may be significant. We use BILOU tags for marking the tokens, therefore, the task is a multiclass classification.

The size of the training set is 426803 tokens, 7684 of which belong to DF, and the test set consists of 57354 tokens, 1509 of which belong to DF.

In this approach, we use few features but employ the context. For each target token, we use:

1. token text;
2. token POS tag;
3. FastText Skip-Gram 300-dimension vector of token lemma.[1]

We do not use FastText as a classifier, since we believe that using the explicit word context and POS tags is more effective in this case than word vectors only.

We also include text and a POS tag for five tokens to the left and five tokens to the right. Moreover, we include five previous target tags as features. The total number of features after one-hot encoding of the tokens is 482614, which exceeds even the number of objects. Therefore, the selection of relevant features may improve the performance. We conduct feature selection using Logistic Regression with L1 penalty and balanced class weight, which reduces the number of features to 5774. For prediction, we use Logistic Regression with L2 penalty and balanced class weight.

We have compared the performance of models trained on different feature sets. The results are presented in Table 1. Text+POS+Target stands for features obtained from one-hot encoding of target word and context words and their POS tags. In +FastText feature set, FastText vector of target word is added. One more feature set is +Selection, which is a filtered version of +FastText set obtained via above-mentioned feature selection model. Maximum metrics values for each class are marked in bold.

Although these three models show similar prediction quality, the last one shows the best recall for all classes, which in our view is preferable. Therefore, further in this paper we use the model trained on selected features. As there is a certain pattern in the predictions (B I I L), and the DF sequence cannot be very long, model predictions should be corrected by rules. We have elaborated the following rules:

1. Change all ‘O’ tags to ‘I’ between ‘B’ and ‘L’ tags if the distance between ‘B’ and ‘L’ does not exceed 5. Otherwise, if ‘B’ and ‘L’ tags are separated with a long ‘O’ tag sequence, change ‘B’ and ‘L’ tags to ‘U’.

[1] The pre-trained model is taken from *RusVectōrēs* website [12].

	Text+POS+Target			+FastText			+Selection		
	P	R	F1	P	R	F1	P	R	F1
B	**0.37**	0.33	0.35	**0.37**	0.36	**0.36**	0.35	**0.37**	**0.36**
I	**0.24**	**0.22**	**0.23**	0.23	**0.22**	**0.23**	0.19	**0.22**	0.20
L	**0.35**	0.31	0.33	**0.35**	0.33	**0.34**	0.32	**0.35**	0.33
O	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98
U	**0.07**	0.40	0.11	0.03	0.20	0.06	**0.07**	**0.60**	**0.13**

Table 1: Token-based model: different feature sets

	Precision	Recall	F1-score	Support
B	0.35	0.37	0.36	356
I	0.37	0.21	0.27	792
L	0.33	0.35	0.34	356
O	0.98	0.99	0.98	55845
U	0.07	0.60	0.13	5

Table 2: Prediction quality of token-based model after rules application

2. Correct boundaries:
   (a) Change ‘I’ to ‘L’ if the tag is preceded by ‘I’ or ‘B’ and followed by ‘O’.
   (b) Change ‘I’ to ‘B’ if the tag is preceded by ‘O’ or ‘L’ and followed by ‘I’ or ‘L’.
   (c) Change everything except for ‘O’ to ‘U’ if the tag is preceded and followed by ‘O’.
3. Delete all non-‘O’ sequences whose length exceeds 10.

The prediction quality obtained after the application of rules is shown in Table 2.

In this case, not all rules were applied. Rule 1 was not applied, Rule 2 was applied once. There was an ‘I L I I’ pattern in a predicted sequence, which was corrected to ‘I L B I’, thus correcting the split between two DF. However, the resulting two DF were too long and were correctly eliminated by Rule 3. Rule 3 was generally applied to more than 500 tokens, thus having a major impact on the result. (1) illustrates the Rule 3 case, where the whole sentence was incorrectly tagged as a DF probably due to its beginning.

(1) *Nu chto, esli by u nas v Rossii bylo pobol’she takih, kotorye by tak mudro rassuzhdali?*
‘Well, if we had in Russia more of those who would reason so wisely?’

The benefit of applying rules in this case varies for different tags but the main improvement is the precision of ‘I’ prediction, which rose from 0.19 to 0.37.

2.2 Clause-based model

In this approach, the text is split into the elements that we call ‘pseudoclauses’. By ‘pseudoclauses’ we mean segments between punctuation marks which roughly correspond to a clause. (2) is the example of a line separated into pseudoclauses (the DF pseudoclause is in bold):

(2) *[**Pochemu by i net?**] [Bud’ eshhe raz angelom-hranitelem,] [provedi raz"jasnitel’nuju rabotu sredi Petrovyh i Sidorovyh. . .]*
‘Why not? Be a guardian angel again, conduct explanatory work among Petrovs and Sidorovs...’

Model	Hyperparameters
Random Forest	class weight: 1:30; 0:1, number of trees: 300, criterion: entropy, minimum samples in a leaf: 5
Logistic Regression	class weight: 1:30; 0:1
Ridge Classifier	class weight: 1:30; 0:1, alpha: 40
SVC	class weight: 1:30; 0:1, C: 0.05, kernel: linear (LinearSVC class was used)

Table 3: Model hyperparameters

This approach may generate incorrect splits, e.g. it splits coordinated NPs, but it is quite suitable for our purpose. We do not expect punctuation marks to occur inside DF, while DF itself is separated from the rest of the speech act with a punctuation mark. Therefore, with our 'pseudoclause' concept, DF remain intact and are separated as a distinct item at the same time.

The training set consists of 100002 pseudoclauses, 2515 of which are DF, and the size of the test set is 13931 pseudoclauses, 337 of which are DF.

For each pseudoclause, we extract a number of features that correspond to a theoretical notion of DF. These features are:

1. pseudoclause text, which is further used for generation of two feature sets:
   (a) count of separate words and word bigrams in the pseudoclause text;
   (b) count of character 3- and 4-grams in the pseudoclause text;
2. length of a pseudoclause in words;
3. presence of an exclamation mark at the end of a pseudoclause;
4. presence of a question mark at the end of a pseudoclause;
5. presence of a predicate, by which we mean any token that was given a 'VERB' tag by pymorphy2 (only verbs in a personal form, not including infinitives, participles and gerunds);
6. presence of a verb in the imperative form;
7. presence of a subject, which is here defined as a noun in Nominative case agreeing with a verb in number, person, or gender;
8. presence of an object, which is here defined as a noun in Accusative case not preceeded by a preposition in a clause with a transitive verb;
9. position of a pseudoclause among the first three pseudoclauses in a conversational turn;
10. count of all parts of speech in the pseudoclase.

The task is binary classification as to whether a pseudoclause belongs to a DF class (1) or not (0). For prediction we use a uniform weight vote of four classifiers: Random Forest Classifier, Logistic Regression, Ridge Classifier, Support Vector Classifier. These four classifiers are chosen due to the presence of a class weight parameter and the performance quality. The hyperparameters are listed in Table 3.

The vote gives a small improvement in prediction quality compared to the classifiers outside the ensemble (Table 4).

	Precision	Recall	F1-score
Random Forest	0.27	0.73	0.39
Logistic Regression	0.28	0.73	0.40
Ridge Classifier	0.26	**0.80**	0.39
SVC	0.28	0.75	0.41
Vote	**0.30**	0.73	**0.42**

Table 4: Prediction quality of four classifiers and their uniform weight vote

	Precision	Recall	F1-score
Baseline	0.18	0.25	0.21
NoText	0.12	**0.86**	0.20
NoSelection	**0.30**	0.73	**0.42**
Selection	0.29	0.74	0.41

Table 5: Prediction quality of baseline and models trained on different feature sets

We also compared models trained on three feature subsets, with the one based on the whole feature set performing slightly better. The comparison is presented in Table 5. ‘Baseline’ is a rule-based baseline, according to which each pseudoclause that belongs to the first three pseudoclauses in the conversational turn contains conjunctions or particles and ends with an exclamation or a question mark is a DF. ‘NoText’ refers to all features except text-based features (word and character n-grams), ‘NoSelection’ is a full feature set, and ‘Selection’ is the feature set selected from all features by Logistic Regression with L1 penalty.

2.3 Model comparison

The quality obtained for test drama texts shows that clause-based model outperforms token-based model significantly. In order to compare the two models, we brought BILOU prediction notation to a binary one, with B, I, L and U tags transformed to tag ‘1’. Table 6 shows the metrics values for a DF class. While precision is comparable, recall is much higher for the clause-based model. It may be accounted for by the presence of clausal features, such as presence of subject, object and predicate, POS tags count and pseudoclause length.

2.4 Discussion

2.4.1 False positives

Both token-based and clause-based algorithms erroneously assign DF label (1 in case of clause-based model and a sequence of tokens with BIL-pattern or a token with U label in case of token-based model) in cases of homonymy of a non-DF pseudoclause with DF. In the example below both models labeled the pseudoclause in bold as DF although in this case it is an introductory word:

Table 6: Prediction quality of token-based and clause-based models

	Precision	Recall	F1-score
Token-based → binary	**0.37**	0.29	0.33
Clause-based	0.30	**0.80**	**0.42**

(3) *Da? U menja, mozhet byt', drugie plany.*
'Really? Maybe I have other plans'.

The algorithms make type I error when the features of semantic and syntactic context of a clause but not the form of a clause itself determines whether the clause belongs to the DF class. We suppose that adding features which characterize the presence and the type of syntactic relations between the target clause and its context will resolve the problem of homonymy and improve the quality of classification. For clause-based model, improving the clause segmentation algorithm could also be helpful.

2.4.2 False negatives

Lists of DF that were labeled as non-DF are mostly the same for token-based and clause based approach. Both models make type II error in cases of non-prototypical DF, which have features not generally typical of DF, such as:

- appearance at the end part of a conversational turn,
- increased length (4 tokens and more),
- complete syntax structure (subject and/or finite predicate),
- mostly tokens of notional parts of speech.

The DF in the example (4), labeled as non-DF by both algorithms (O for all tokens in case of token-based one), consists of 5 tokens, has subject and finite predicate and has tokens of notional parts of speech:

(4) *Eto vy zrja tak dumaete.*
'You should not think like so'.

The fact that both models make mostly identical mistakes means that the feature set used for training is insufficient and needs to be extended.

3 Creation of the list

The created clause-based module was used to extract a list of DF from 420 drama texts of XIX-XXI centuries [1, 7]. Almost 10000 potential DF were extracted and then manually filtered. The final list contains 3000 DF, 1800 of which are unique. The list can be expanded by adding the normalized varieties of DF, which are created according to the regularities revealed in our research; the list then reaches 4000 units.

The resulting DF can be included into the Constructicon database, which describes Russian constructions in a structured way [10]. At this stage, DF description will involve the research on DF pragmatic functions and semantic constraint on the left context. These features may contribute to the theoretical DF classification, which is the principal aim of the project.

4 Conclusion

The task of automatic DF extraction may be considered relatively complicated because of the need to formalize the theoretical features of the constructions under study, and this cannot be done in a straightforward way. Therefore, further development of the tool is mostly connected with feature extraction. The clause-based model could be enhanced by adding features based on distributional semantics as well as by including information about left context, as it contains a stimulus for the DF. The token-based model can probably be improved by the dependency parser output. Both models can probably benefit from employing neural networks, which require much larger training data and more computational resources.

Although the list of Russian DF has been compiled and manually processed, the question how to improve the models remains open, as further development of the project includes exploring DF concept in a typological perspective, which in its turn implies adjusting the automatic extraction tool to other languages.

References

[1] The Lubimovka Young Russian Playwrights Festival. http://lubimovka.ru. Accessed: 2018-01-25.

[2] Gillian Brown and George Yule. *Discourse analysis*. Cambridge university press, Cambridge, 1983.

[3] Teun van Dijk. Introduction: Discourse analysis as a new cross-discipline. In Teun van Dijk, editor, *Handbook of discourse analysis*, chapter 1. Academic Press, New York, 1985.

[4] Charles J Fillmore. The mechanisms of "construction grammar". In *Annual Meeting of the Berkeley Linguistics Society*, volume 14, pages 35–55, 1988.

[5] Charles J Fillmore. Grammatical construction theory and the familiar dichotomies. In *North-Holland Linguistic Series: Linguistic Variations*, volume 54, pages 17–38. Elsevier, 1989.

[6] Charles John Fillmore and Paul Kay. *Construction Grammar Course Book*. University of California, Berkeley, 1993.

[7] Frank Fischer, Tatyana Orlova, Daniil Skorinkin, German Palchikov, and Natasha Tyshkevich. Introducing RusDraCor, a TEI-encoded Russian drama corpus for the digital literary studies. In *International scientific conference "Corpus linguistics–2017"*, St. Petersburg, June 27-30 2017.

[8] Paul J Hopper and Elizabeth Closs Traugott. *Grammaticalization*. Cambridge University Press, 2003.

[9] Ray Jackendoff. The boundaries of the lexicon. *Idioms: Structural and psychological perspectives*, pages 133–165, 1995.

[10] Laura A Janda, Olga Lyashevskaya, Tore Nesset, Ekaterina V Rakhilina, and Francis M Tyers. A Constructicon for Russian: Filling in the Gaps. In Benjamin Lyngfelt, Lars Borin, Kyoko Ohara, and Tiago Timponi Torrent, editors, *Constructicography: Constructicon development across languages*, pages 165–182. John Benjamins, Amsterdam, 2018.

[11] Mikhail Korobov. Morphological Analyzer and Generator for Russian and Ukrainian Languages. In Mikhail Yu. Khachay, Natalia Konstantinova, Alexander Panchenko, Dmitry I. Ignatov, and Valeri G. Labunets, editors, *Analysis of Images, Social Networks and Texts*, volume 542 of *Communications in Computer and Information Science*, pages 320–332. Springer International Publishing, 2015.

[12] Andrey Kutuzov and Elizaveta Kuzmenko. WebVectors: A Toolkit for Building Web Interfaces for Vector Semantic Models. In Dmitry I. Ignatov, Mikhail Yu. Khachay, Valeri G. Labunets, Natalia Loukachevitch, Sergey I. Nikolenko, Alexander Panchenko, Andrey V. Savchenko, and Konstantin Vorontsov, editors, *Analysis of Images, Social Networks and Texts: 5th International Conference, AIST 2016, Yekaterinburg, Russia, April 7-9, 2016, Revised Selected Papers*, pages 155–161. Springer International Publishing, Cham, 2017.

[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[14] Harvey Sacks, Emanuel A Schegloff, and Gail Jefferson. A simplest systematics for the organization of turn taking for conversation. In Jim Schenkein, editor, *Studies in the organization of conversational interaction*, pages 7–55. Elsevier, 1978.

[15] Alison Wray. *Formulaic language and the lexicon.* Cambridge University Press, Cambridge, 2002.