Genre-shift detection using Functional Text Dimensions

Tagir Gadzhiev [1], Serge Sharoff [2]

[1] Russian State University for the Humanities

[2] University of Leeds

Abstract

In this article we present an approach to detecting genre shifts. A genre shift is when a passage of text of one genre gives place to a passage of another genre. The proposed genre shift detection method (GSD-method) is based on the system of functional text dimensions (FTDs) — functional categories that allow us to characterise genre of a text based on the purpose of the text. The algorithm for genre shift detection is fully implemented and is shown to produce sensible segmentation with the WinPR based f-score metric of 0.61 on our test corpus of synthetic texts. The results of this work may have practical relevance to information retrieval, machine translation, corpus linguistics and other fields of computational linguistics where it is important have a more fine-grained information about genre structure of an individual text.

Keywords: genre shift, genre classification, functional dimensions.

Определение жанровых переходов на базе системы функциональных текстовых размерностей

Тагир Гаджиев[1], Сергей Шаров[2]

[1] Российский государственный гуманитарный университет

[2] University of Leeds

В статье предлагается метод автоматического определения жанровых переходов в жанрово-неоднородных текстах. Жанровым переходом мы называем смену одного текстового жанра другим в рамках единого текста. Предлагаемый метод базируется на системе функциональных текстовых размерностей (ФТР) — наборе функциональных категорий, позволяющих охарактеризовать жанр отдельного текста с точки зрения функции, на которую нацелен текст. Ф-мера качества сегментации, полученная на базе метрики WinPR, демонстрирует значение 0.61 на корпусе автоматически сгенерированных синтетических текстов с априорной сегметацией. Результаты работы могут найти практическое применение в области информационного поиска, машинного перевода, а также значимы для корпусной лингвистики, поскольку могут быть применены для выявления жанровой неоднородности отдельных текстов в корпусах и автоматической разметки жанровых переходов.

Ключевые слова: жанровые переходы, жанровая неоднородность, жанровая классификация, функциональные размерности.

1 Introduction

A sudden change in genre of an individual text we call a **genre shift**. This phenomena is frequently encountered in texts (in particular, on the Internet), for example in research articles that cite resources of different genre, in advertisements that contain instructions ("How To"), news items containing strongly argumentative interviews etc.

Detecting changes in genre of an individual text is a kind of a text segmentation task. Text segmentation deals with subdividing a text into homogeneous units on various grounds. One example concerns the topic-based segmentation, which has been developing since the early 1990s (Jane and Hirst, 1991; Kozima, 1993; Reynar, 1994; Salton, 1996; Hearst, 1997). However, this task is very different from segmenting a text into homogeneous units with respect to their genres.

Availability of methods for genre shift identification is of great importance for corpus linguistics as it would allow for more fine-grained comparison between corpora by virtue of revealing genre structure of individual texts.

One of the approaches to the problem of describing genres is the system of Functional Text Dimensions or FTDs (Sharoff, 2018). Each dimension is a functional category which characterizes the purpose of a text . As experiments on clasterization show (Lagutin et al., 2015), some stable combinations of FTDs correspond to particular text-genres in the common sense of the notion of genre.

In the FTD approach each text sample is assessed in 18 functional categories (i.e. along 18 functional dimensions). Each dimension is characterized by a code (e.g. A4), label (e.g. `fictive`), test question and prototypical genres.

A particular dimension (purpose) in a text is better identified by answering its test question. The whole set of questions can be reduced to the following template: "Is this text written for [a purpose]?". For instance for the dimension A7, which has a corresponding label `instruct`, a test question is as follows: *To what extent does the text aim at teaching the reader how something works?* Answering a test question a human assessor evaluates a text on a customised Likert scale: 0 none or hardly at all; 0.5 slightly; 1 somewhat or partly; 2 strongly or very much so (Sharoff, 2018). The value of 0 ('None') is the default value. It means that the dimension is not present in a text assessed and the text does not resemble any prototypical genre texts for this dimension. The value of 2 means that the dimension is strongly present in the text assessed and at the same time the text is close to one of the prototypical texts for this dimension.

We call a text sample **mono-functional** when its FTD vector contains only one Strongly-present FTD and all other FTDs (text functions) are absent. Such a vector means that a text is homogeneous in the terms of genre and in the multidimensional space of functional dimensions such a text is placed near the prototype for its FTD. For instance, a physics journal article (with no uncommon inclusions) is characterised by an FTD vector that contains the only value of 2 — for the dimension A14.

We say that a text sample is **poly-functional** when it serves several functions at the same time, thus it has several Strongly-present FTDs. For instance, a news item might be argumentative (A1) and newswire-like (A8) at the same time.

Finally, there is a case of **mixed genres** when a text contains several mono-functional segments. For example, a philology research article containing some citations from a work of fiction. This is the type of texts that the proposed method for genre shift detection concerns.

This paper is structured as follows:

- (Section 2) gives a short description of the related work;
- (Section 3) gives details on datasets sources and structure, preprocessing applied to the data, the classifier configuration; it describes metrics used to evaluate the segmentation quality and genre shift detection scheme;
- (Section 4) discusses the results;

2 Related Work

Two important aspects of the investigation in the field of automatic genre classification in the last few decades can be noticed which are of high relevancy for the task of genre shift detection (GSD). The first is the search for the general framework that would allow to describe any text in terms of its genre and produce a representation of the text that incorporates its genre related features and would be flexible enough to suit needs of researchers in different genre related tasks (Biber, 1995; Kessler et al. 1997, Wachsmuth and Bujna, 2011; Sharoff, 2018).

The second aspect is investigating of different methods intended to handle those representations. These methods among others include discriminant analysis (Biber, 1995), Logistic Regression (Kessler et al. 1997), Multiple Regression (Stamatatos et al., 2001), Support Vector Machines (Freund et al. 2006, Wachsmuth and Bujna, 2011), Principal Component Analysis (Feldman et al., 2009) and Likelihood Estimation (Zampieri and Gebre, 2012, 2014).

In the last decade neural network architectures made a huge progress and have shown to be effective in solving a wide range of linguistic tasks such as document topic recognition (Johnson, Zhang, 2015), language modeling (Mikolov et al., 2010, 2011) and sequence tagging (Xu et al., 2015). The effectiveness of a neural network for a particular task depends on its architecture. Recurrent neural networks have shown their effectiveness in tasks where it is important to preserve information about structure of the data (Sutskever, 2014) which is important in the case of genre identification.

3 Methodology

3.1 Data

3.1.1 Training Set

The training corpus in its main part comes from the gold standard presented in Lagutin et al. (2015) which included 618 texts annotated using FTDs. The gold standard corpus was extended by adding 435 annotated texts from the social network ’vkontakte.ru’ and a popular blog platform ’blog.mail.ru’.

3.1.2 Test Set

Different approaches to assessing the text segmentation quality are proposed in literature. One among the others is comparing segmentation against automated segmentation strategies (Hearst, 1997). We are not aware of any open-source corpora containing genre-segmentation markup so in order to test the GSD-method proposed we generated a dataset of 50 synthetic texts with a priory known manually set segmentation. The general procedure we followed generating the dataset is to a large extent similar to that of Manchanda et al. (2018).

We separated part of mono-functional texts from the initial corpus and used them as a source for the dataset of synthetic texts. Each synthetic text in the dataset is composed of segments of two different functions (in terms of Functional Text Dimensions). Each segment of a particular function is picked randomly from the subset of that function.

For the experiments present we chose the functional dimension A14 ('scitech') as a main dimension and the functional dimension A4 ('fictive') as an auxiliary dimension. The choice is motivated by the availability of texts with this combination of dimensions (philology journal articles which cite some works of fiction) for the comparison with our modeled case (this is the prospect of further work). All the samples in a dataset are produced the same way.

Dataset parameters: chunk size — **30** tokens, n_chunks per segment of the main dimension (function) — **10**, n_chunks per segment of the auxiliary dimension — **5**, n_segments of main dimension — **7**, n_segments of auxiliary dimension — **6**, resulting synthetic text length — **3000** tokens .

3.2 Data Preprocessing

Length of samples in the initial corpus used for training and testing the classifier is restricted with **1000** tokens. As lexical cues are mentioned among the factors that degrade the performance of the genre classifier we replace rare words (with rang less than **3000**) with the most common morphological tags for those words. Dataset samples are tokenized and split into chunks of the length of **30** tokens which seems to be a reasonable approximation for the average sentence length.

3.3 Classification Model

The bidirectional GRU (gated recurrent unit) recurrent neural network architecture is used for the multi-class, multi-label classification in the experiments present (the original source-repository on GIT[1]).

Model: Embedding —> Dropout —> GRU —> Dropout —> FC

The classifier is used ’out of the box’ with no parameter tuning. Default parameters are as follows: batch size **128**, starting learning rate **0.005**, weight decay **1e-4**, number of layers **2**, hidden size **128**, clip **0.25**, number of training epochs **10**. The learning rate was decayed by 10 every **8** epochs. Cross-entropy Loss + Adam optimizer. The word-2-vec embeddings of dimension **300** pre-trained on RuWac corpus by Serge Sharoff are used[2]. The cross entropy loss output values in range of the upper 3 quarters are considered to be the true values.

3.4 WinPR Metric to Evaluate Segmentation

The main metrics used for evaluation in tasks related to text segmentation are P_k (Beeferman et al., 1997, 1999), WindowDiff (Pevzner and Hearst, 2002), and WinPR (Scaiano, Inkpen, 2012). To assess the quality of segmentation in this paper we use the **WinPR** metric[3] as it allows for producing precision, recall, and f-score measures (due to distinguishing between false positive and false negative errors). The metric is based on the comparison of the number of boundaries in a probe window in the computed segmentation against the reference segmentation. Additionally the metric is insensitive to window size and allows to customise sensitivity to *near miss* errors wich is very important in tasks related to text segmentation. A near miss error occurs when a calculated (or predicted) boundary does not exactly match a reference boundary but is shifted from it for some extent.

3.5 Genre shift detection

General approach taken in this work follows that proposed in (Hearst, 1997) where the author introduces several methods for 'subtopic shift' detection. Following that work we introduce a notion of 'genre shift' as an analogy to 'subtopic shift'.

To find the boundaries between consecutive passages of different genres we use an approach derived from the block comparison algorithm by Hearst (1997). As Hearst states "block comparison, compares adjacent blocks of text to see how similar they are according to how many words the adjacent blocks have in common." Similarly, to measure the similarity between adjacent blocks of text-chunks we assess the output of the neural network looking how similar the adjacent blocks of labels are.

3.5.1 Block comparison algorithm adopted for genre shift detection

First, the `block size` is set which determines how many units (e.g. text chunks or sentences) each block contains. Blocks on both sides of a gap (e.g. block A and block B at *gap 0* in Fig. 1) contain the same `block size` number of labels. For the sake of convenience we add a padding vector of `block`

1 https://github.com/keishinkickback/Pytorch-RNN-text-classification
2 http://corpus.leeds.ac.uk/serge/wikis/w2v/ruwac-300-fast.vec.xz
3 https://pan.webis.de/clef19/pan19-web/style-change-detection.html

`size` length to both ends of the vector of labels. The most common label among the first `block size` labels in the vector of labels is chosen as a filler for the padding vector on the left (the label *7* is chosen in the depicted case). The padding vector on the right is filled the same way. This is done as a way to handle border conditions and models a situation when the conditions at the beginning and the end of the text remain unaltered.

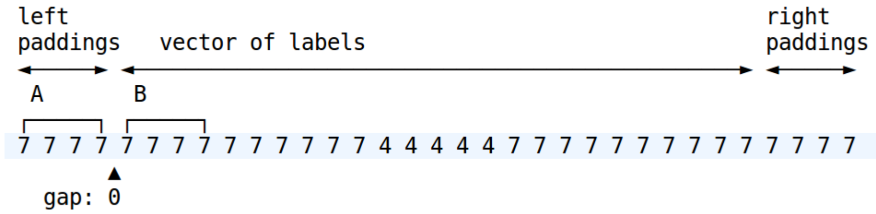

Fig. 1. The resulting vector of labels for a particular text sample.

At each gap in the vector of labels (ends are included, gaps within the padding additions are excluded) we calculate a *measure of similarity* between two consecutive blocks of labels (block A and block B if Fig. 1) which is defined as a simple dot product of one-hot-encoded vector representations of the blocks, for instance [0 0 0 0 0 0 0 4 0 0 0] • [0 0 0 0 0 0 0 4 0 0 0] = **16** at gap 0 in Fig. 1.

Fig. 2 depicts the general scheme for genre shift detection. (**a**) shows the segmentation which consists of 3 segments. The first and the third segments contain 10 labels each and correspond to the dimension *A14* (label *7*). The second segment is considered to be an inclusion of genre that differs from the genre of the surrounding text and corresponds to the dimensino A9 (label 4). (**b**) is a vector of predicted labels. Each label corresponds to a chunk of text. An 'ideal' case is depicted in Fig. 2 when each label in a segment is predicted correctly so all the labels for an individual segment are the same. The first and the last four labels in (**b**) are paddings. The gaps within padding vectors are not considered so are eliminated. (**c**) shows gaps between labels — these are the positions where a genre shift may potentially occur. (**d**) represents indices corresponding to gaps.

To identify boundaries between consecutive passages of different genres, the similarity scores calculated at each gap are combined into a vector of similarity scores (**e**), which is then analysed for local minima and maxima [4].

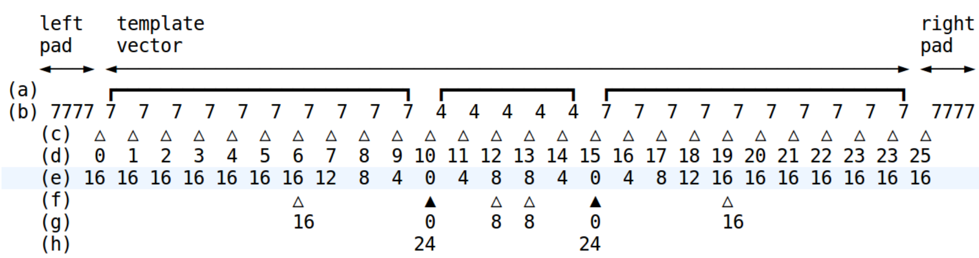

Fig. 2. Scheme of genre shift detection based on the use of similarity scores at gaps between consecutive blocks of text chunks. (a) — *segmentation*; (b) — vector of *labels*; (c) — gaps (*potential segment boundary positions*); (d) — *indices* of gaps; (e) — vector of *similarity scores*; (f) — *positions* of local extrema; (g) — *similarity scores* in local extrema positions; (h) — *depth scores* in local extrema valleys.

This results in a vector (**g**) of local extrema which are found at positions (**f**). The vector of local extrema (**g**) is analysed then for local minima to finally determine positions of genre shifts. Local minima positions are shown with the black triangles in (**f**) and local maxima are shown with the white triangles.

4 To find extrema we apply the 'detect_peaks' library by Marcos Duarte: https://github.com/demotu/BMC

The valleys between two maxima in the vector of similarity scores (**e**) correspond to the positions where a genre shift may potentially happen.

The **depth score** (**h**) is calculated with the subtraction of the similarity score found at a *minimum* from each of the scores at gaps of *neighbouring maxima* and summing up the result. For instance, at gap 10 in Fig. 2 the depth score is calculated as follows: (16 – 0) + (8 – 0) = **24**. The depth score can be used to set the sensitivity of the GSD-system by placing a restriction on the potential boundary positions (gaps) that are not to be considered shift positions as the value of the depth score at these gaps is not high enough.

4 Results and Discussion

4.1 Classifier Performance

The task of genre shift detection requires a model capable to identify genre of very short pieces of text. For these preliminary tests we trained a bi-GRU classification model using the chunk size of **30** tokens, i.e. all the samples in the training and test sets were divided into chunks of the length 30, which is close to the average sentence length.

Each sample is assessed in 11 functional categories (i.e. along 11 functional dimensions). Each category has a label which is given in brackets: A1 (argum), A4 (fictive), A7 (instruct), A8 (hardnews), A9 (legal), A11 (personal), A12 (compuff), A14 (scitech), A16 (info), A17 (eval), A20 (appell). See Sharoff (2018) for more detail.

Table 1: Precision, recall, f-measure on the test set. Classifier ***bi-GRU***, chunk size ***30***, epoch ***10***.

	A1	A4	A7	A8	A9	A11	A12	A14	A16	A17	A20	weighted
precision	0.675	0.682	0.878	0.500	0.874	0.618	0.800	0.697	0.748	0.425	0.571	**0.702**
recall	0.457	0.514	0.607	0.275	0.832	0.417	0.254	0.587	0.634	0.202	0.143	**0.534**
f1-score	0.545	0.586	0.718	0.354	0.852	0.498	0.386	0.637	0.687	0.274	0.229	**0.600**

4.2 Genre Shift Detection Performance

The *depth scores* are assigned at gaps which are considered to be potential boundaries between segments of different genres. The score results from the block comparison algorithm which compares two blocks of labels on both sides of each gap. Each label corresponds to a chunk of text and encodes a predicted dimension for the chunk. Setting a *threshold* for the depth score allows discarding boundaries with the value of depth score under the threshold. This allows to adjust the sensitivity of the GSD-system. *Near-miss sensitivity* of the WinPR metric is set the way that a calculated boundary shifted from the reference boundary by one unit (i.e. one chunk in our case) is considered to be a *near-miss error* and contributes partly to the final metric.

Note that the `block size` used to calculate *depth scores* and the `window size` used to calculate the WinPR metrics — these are two separate parameters. Although they may be set different values, in the experiments present they are always set the same value. In the further discussion of the WinPR metrics we will only mention `window size` as it relies to the metrics calculations.

In Fig. 3, 4, 5 the dependence of the WinPR metrics (f-score, precision and recall) on the *depth score threshold* is shown for three most informative values of the `window size`: 4, 5 and 6. It can be recognised from the central box-plot in Fig. 3, which corresponds to the `window-size` of **5**, that the best performance of the GSD-system is achieved with this `window size` and results in the median f-score of **0.61** at the depth threshold value of 2 (see Table 2).

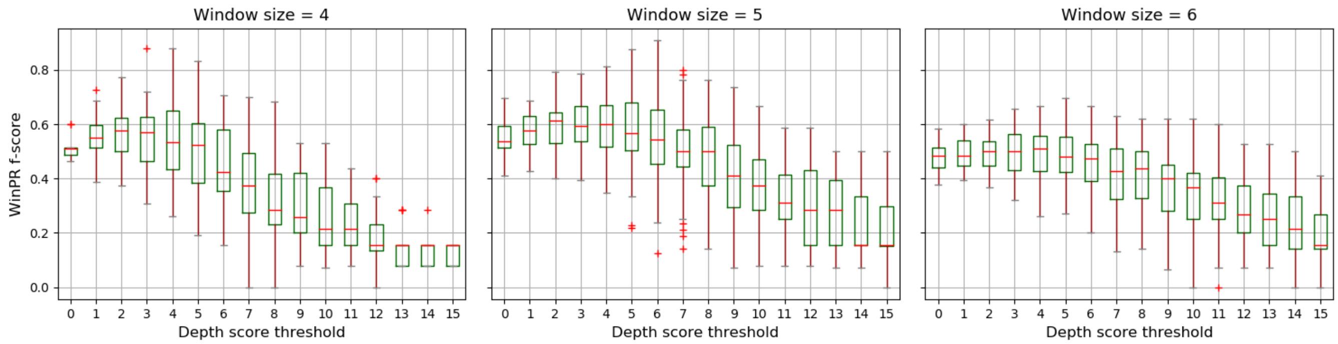

Fig. 3. **WinPR f-score — depth score threshold** dependence for different values of **window size**.

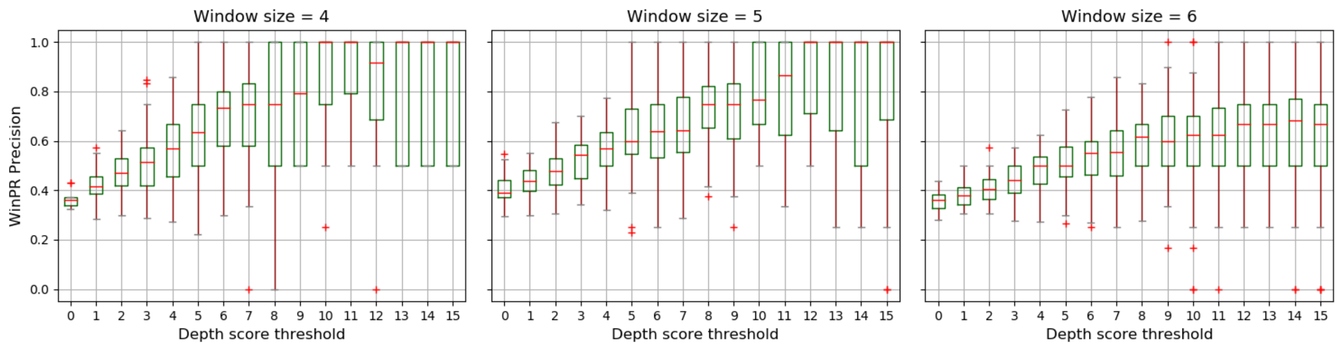

Fig. 4. **WinPR precision — depth score threshold** dependence for different values of **window size**.

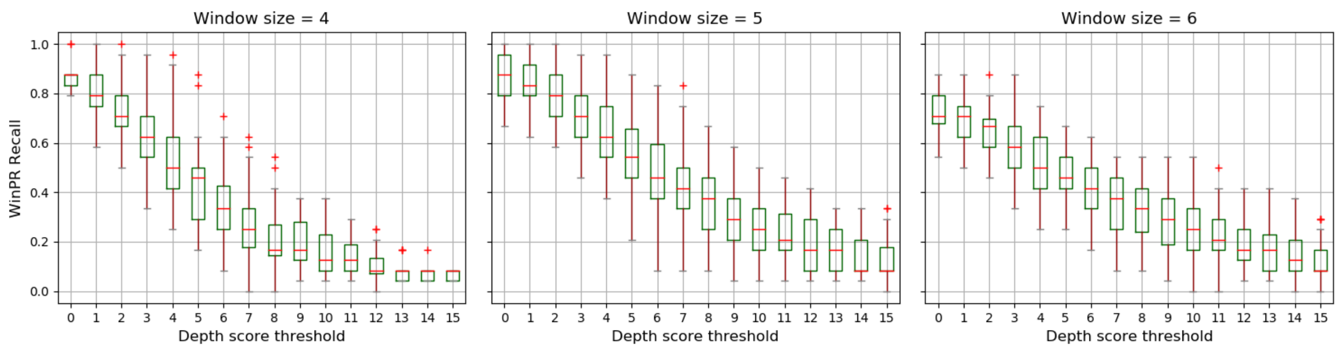

Fig. 5. **WinPR recall — depth score threshold** dependence for different values of **window size**.

Table 2. Statistics for **WinPR f-score** for the **window size** of **5**.

	count	mean	std	min	25%	50%	75%	max
0	40.00000	0.54848	0.06304	0.41026	0.51318	0.53752	0.59459	0.69697
1	50.00000	0.57505	0.06615	0.42857	0.52668	0.57576	0.62909	0.68750
2	50.00000	0.59335	0.07886	0.40000	0.53039	0.61290	0.64516	0.79310
3	49.00000	0.59642	0.08660	0.39286	0.53571	0.59375	0.66667	0.78571
4	48.00000	0.59321	0.10781	0.34615	0.51792	0.60000	0.66964	0.81481
5	50.00000	0.57107	0.13117	0.21739	0.50543	0.56832	0.67964	0.87500
6	48.00000	0.54170	0.15644	0.12500	0.45455	0.54423	0.65259	0.90909
7	49.00000	0.49642	0.14921	0.14286	0.44444	0.50000	0.57895	0.80000
8	47.00000	0.46866	0.14903	0.14286	0.37500	0.50000	0.58947	0.76190
9	45.00000	0.41230	0.15226	0.07143	0.29412	0.41176	0.52381	0.73684
10	46.00000	0.36897	0.12836	0.07692	0.28571	0.37500	0.47059	0.66667
11	46.00000	0.33680	0.13307	0.07692	0.25000	0.31250	0.41579	0.58824
12	42.00000	0.30662	0.14764	0.07692	0.15385	0.28571	0.43107	0.58824
13	42.00000	0.26550	0.13423	0.07143	0.15385	0.28571	0.39375	0.50000
14	41.00000	0.23379	0.12921	0.07143	0.15385	0.15385	0.33333	0.50000
15	40.00000	0.20836	0.13162	0.00000	0.15110	0.15385	0.29762	0.50000

This is an expected behaviour as the `window size` of 5 corresponds to the size of inclusions of auxiliary genre in our synthetic texts. It suggests that for real texts the `window size` corresponding to the *mean length* of the auxiliary genre inclusions should perform best.

The box-plots in the Fig. 5 show that the best performing `depth score threshold` values are in range of 2 to 4. This means that discarding shallow boundaries with depth scores 0 and 1 benefits the performance of the GSD-system but further restriction discards too many true boundaries and that harms the performance. The latter fact can be seen in Fig. 5 which reveals that the WinPR *recall* falls down when more restriction is placed on the `depth score threshold`.

Fig. 4 shows that the *precision* increases for higher values of `depth score threshold` which means that there remain few boundaries and the percentage of the true boundaries among them is high.

Thus the setting of `depth score threshold` allows to shift balance to higher precision or recall depending on needs of a researcher.

5 Conclusion

The proposed method for genre shift detection based on the general framework of Functional Text Dimensions demonstrates promising results. For the synthetic texts with rather many inclusions of auxiliary genre (6 inclusions of 150 tokens each) separated by segments of the main genre (of 300 tokens length each) it produces the WinPR based f-score of **0.61** for the segmentation quality.

The method is shown to be able to identify boundaries between segments of different functions which in this case correspond to genres as the source texts are mono-functional. The method allows for adjusting the balance towards precision or recall via setting an appropriate depth score threshold.

Potential for further improvement may be obtained by using more stable classification scheme.

6 References

1. Beeferman, Douglas, Adam Berger, and John Lafferty. (1997). Text segmentation using exponential models. In Proceedings of the Second Conference on Empirical Methods in Natural Language Processing, pages 35–46.
2. Beeferman, Douglas, Adam Berger, and John Lafferty. (1999). Statistical models of text segmentation. Machine Learning, 34(1-3), February
3. Biber. D. (1995). Dimensions of register variation: A cross-linguistic comparison. Cambridge University Press, Cambridge, UK, 1995.
4. Feldman, S., M. A. Marin, M. Ostendorf, and M. R. Gupta. (2009). Part-of-speech histograms for genre classification of text. In Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, pages 4781–4784, Washington, DC.
5. Hearst M. A. (1994). Multi-paragraph segmentation of expository text. In Proceedings of the 32nd Meeting, pages 9-16, Las Cruces, NM, June. Association for Computational Linguistics.
6. Hearst M. A. (1997). Texttiling: Segmenting text into multiparagraph subtopic passages, Computational linguistics, vol. 23, no. 1, pp. 33–64
7. Jane M. and Hirst G. (1991). Lexical cohesion computed by thesaural relations as an indicator of the structure of text. Computational Linguistics, 17(1):21-48.
8. Johnson, R., & Zhang, T. (2015). Effective Use of Word Order for Text Categorization with Convolutional Neural Networks. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado. Association for Computational Linguistics.

9. Kessler, B., Nunberg, G. & Schuetze, H. (1997). Automatic Detection of Text Genre. 35th Ann. Meeting ACL 32–38.

10. Kozima, Hideki. (1993). Text segmentation based on similarity between words. In Proceedings of the 31th Annual Meeting (Student Session), pages 286-288, Columbus, OH. Association for Computational Linguistics.

11. Lagutin M. B., Katinskaya A. Y., Selegey V. P., Sharoff S., Sorokin A. A. (2015). Automatic Classification of Web Texts Using Functional Text Dimensions. Moscow, 2015

12. Luanne F., Charles L. A. Clarke, and Elaine G. Toms. (2006). Towards genre classification for IR in the workplace. In Proceedings of the 1st International Conference on Information Interaction in Context, pages 30–36, New York, NY

13. Manchanda S., Karypis G. (2018). Text segmentation on multilabel documents: A distant-supervised approach, 2018 IEEE International Conference on Data Mining

14. Mikolov, T., Karafiat, M., Burget, L., Cernocky, J., & Khudanpur, S. (2010). Recurrent neural network based language model.. In INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010, pp. 1045-1048.

15. Mikolov, T., Kombrink, S., Lukas Burget, Cernocky, J. H., & Khudanpur, S. (2011). Extensions of recurrent neural network language model. In Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on, pp. 5528-5531. IEEE.

16. Pevzner L. and Hearst M. (2002). A critique and improvement of an evaluation metric for text segmentation. Computational Linguistics, Jan.

17. Reynar, Jeffrey C. (1994). An automatic method of finding topic boundaries. In Proceedings of the 32nd Annual Meeting (Student Session), pages 331-333, Las Cruces, NM. Association for Computational Linguistics.

18. Salton, Gerard, Amit Singhal, Chris Buckley, and Mandar Mitra. (1996). Automatic text decomposition using text segmentation and text themes. In Proceedings of Hypertext '96, Seventh ACM Conference on Hypertext, pages 53-65, Washington, D.C.

19. Scaiano M, Inkpen D. (2012) Getting More from Segmentation Evaluation. 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 362–366, Montreal, Canada, June 3-8, 2012

20. Sharoff, S. (2018). Functional Text Dimensions for the annotation of Web corpora. Corpora, 31:1, 2018.

21. Stamatatos E., Fakotakisy N., Kokkinakisz G. (2001). Automatic Text Categorization in Terms of Genre and Author, Association for Computational Linguistics

22. Sutskever, I., Vinyals, O., & Le, Q. V. V. (2014). Sequence to Sequence Learning with Neural Networks. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., & Weinberger, K. Q. (Eds.), Advances in Neural Information Processing Systems 27, pp. 3104{3112. Curran Associates, Inc.

23. Wachsmuth H, Bujna K. (2011). Back to the Roots of Genres: Text Classification by Language Function. Proceedings of IJCNLP ’11 5th International Joint Conference on Natural Language Processing, Chiang Mai, Thailand, November 8-13

24. Xu, W., Auli, M., & Clark, S. (2015). CCG Supertagging with a Recurrent Neural Network. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pp. 250{255, Beijing, China. Association for Computational Linguistics.

25. Zampieri, M. and Gebre, B. G. (2012). Automatic identification of language varieties: The case of Portuguese. In Proceedings of KONVENS2012, pages 233–237, Vienna, Austria.

26. Zampieri, M. and Gebre, B. G. (2014). Varclass: An open source language identification tool for language varieties. In Proceedings of Language Resources and Evaluation (LREC), Reykjavik, Iceland.