Random at SemEval-2020 Task 1: Gaussian Mixtures Cross Temporal Similarity Clustering

Pierluigi Cassotti
University of Bari, Italy
pierluigi.cassotti@uniba.it

Annalina Caputo
ADAPT Centre, Dublin City University, Ireland
annalina.caputo@dcu.ie

Marco Polignano
University of Bari, Italy
marco.polignano@uniba.it

Pierpaolo Basile
University of Bari, Italy
pierpaolo.basile@uniba.it

Abstract

This paper describes the system proposed by the Random team for SemEval-2020 Task 1: Unsupervised Lexical Semantic Change Detection. We focus our approach on the detection problem. Given the semantics of words captured by temporal word embeddings in different time periods, we investigate the use of unsupervised methods to detect when the target word has gained or lost senses. To this end, we define a new algorithm based on Gaussian Mixture Models to cluster the target similarities computed over the two periods. We compare the proposed approach with a number of similarity-based thresholds. We found that, although the performance of the detection methods varies across the word embedding algorithms, the combination of Gaussian Mixture with Temporal Referencing resulted in our best system.

1 Introduction

The recent development in word embeddings, and their increasing capability to capture lexical semantics has inspired the application of these methods to new tasks and introduced new challenges. The diachronic analysis of language is one of these linguistic tasks which has benefited from the advantages of these new methods, i.e. the capability to build semantic representations of words by skimming through large corpora spanning multiple time periods. SemEval 2020 Task 1 (Schlechtweg et al., 2020) addresses the current lack of a systematic approach for the evaluation of automatic methods for diachronic analysis by proposing a common evaluation framework that comprises two tasks and covers corpora written in four different languages, namely German (Zeitung, 2018; Textarchiv, 2017), English (Alatrash et al., 2020), Latin (McGillivray and Kilgarriff, 2013), and Swedish (Borin et al., 2012). Given two corpora C_1 and C_2 for two periods t_1 and t_2, Subtask 1 requires participants to classify a set of target words in two categories: words that have lost or gained senses from t_1 to t_2 and words that did not, while Subtask 2 requires participants to rank the target words according to their degree of lexical semantic change between the two periods. We tackle the problem of automatically detecting lexical semantic changes with approaches that rely on temporal word embeddings. These approaches create a word vector representation for each time period by exploiting a shared semantic space. Similarity measures can then be used to capture the extent of a word semantic change between two periods. Some temporal word embedding techniques adopt a two-step approach, where they first learn separate word embeddings for each time period and then align the word vectors across multiple time periods (Hamilton et al., 2016). Other *dynamic* approaches incorporate the alignment directly into the learning stage via the optimisation function (Tahmasebi et al., 2018). Dynamic word embeddings can be further categorised according to the constraint imposed on the alignment. The *explicit* alignment adopts a conservative approach to the semantic drift that a word can undergo by posing a limit to the distance between the word vectors belonging to the two temporal spaces. In the *implicit* alignment, there is no need for explicit constraint since the alignment is automatically performed by sharing the same word context vectors across all the time lapses.

In this work, we focus on dynamic word embeddings by exploring methods based on both explicit, such as Dynamic Word2Vec (Yao et al., 2018), and implicit alignment, namely Temporal Random Indexing

This work is licensed under a Creative Commons Attribution 4.0 International License. License details: http://creativecommons.org/licenses/by/4.0/.

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 74–80
Barcelona, Spain (Online), December 12, 2020.

(Basile et al., 2015) and Temporal Referencing (Dubossarsky et al., 2019). We analyse the use of different similarity measures to determine the extent of a word semantic change and compare the cosine similarity with Pearson Correlation and the neighborhood similarity (Shoemark et al., 2019). While these similarity measures can be directly employed to generate a ranked list of words for Subtask 2, their adoption in Subtask 1 requires further manipulation. We introduce a new method to classify changing vs. stable words by clustering the target similarity distributions via Gaussian Mixture Models. We describe the embedding models and the clustering algorithm in Section 2, while Section 3 provides details about the hyper-parameter selection. Section 4 reports the results of the task evaluation followed by some concluding remarks in Section 5.

2 System description

We model the problem of automatic detection of semantic change by exploiting temporal word embeddings $E_i : w \to \mathbb{R}^d$ that project each word w in the vocabulary V into a d-dimensional semantic space. Given two different time periods t_1 and t_2, we create two embeddings E_1 and E_2. We investigate several models to compute temporal word embeddings:

Dynamic Word2Vec (DW2V) (Yao et al., 2018) simultaneously learns time-aware embeddings by aligning and reducing the dimensionality of time-binned Positive Point-wise Mutual Information matrices.

Temporal Random Indexing (TRI) (Basile et al., 2015) implicitly aligns co-occurrence matrices by using the same random projection for all the temporal bins.

Collocations extracts for each word and each time period the set of relevant collocations through the Dice score. As similarity function, we measure the cosine similarity between the sets of collocations belonging to the two different time periods. More details are reported in Basile et al. (2019).

Temporal Referencing (TR) (Dubossarsky et al., 2019) used only in the post-evaluation, it consists in a modified version of Word2Vec Skipgram that adds a temporal referencing to target vectors, keeping context vectors unchanged.

A similarity measure between vectors in the two temporal spaces is adopted to compute the extent of the semantic drift of the target words. We explored several similarity measures:

Cosine similarity (CS) is the cosine of the angle between two vectors.

Pearson correlation (PC) measures the linear correlation between two variables, in case of centred vectors (with zero means) is equivalent to the cosine similarity.

Neighborhood similarity (NS) computes two k-neighbour sets $nbrs_k(E_1(w))$ and $nbrs_k(E_2(w))$ and the union set $\mathcal{U} = nbrs_k(E_1(w)) \cup nbrs_k(E_2(w))$. Two second-order vectors, one for each word representation u_j, are created. The components of u_i are the cosine similarity between the vector v_j[1] and the i-th element of $\mathcal{U}$: $u_{j_i} = cos(v_j, \mathcal{U}(i))$. The Neighborhood similarity is the cosine similarity between the second-order vectors. In all the experiments we set $k = 25$.

2.1 Subtask 2

In Subtask 2, we use one of the three similarity measures (CS, PC, NS) to compute the set of target similarities $\mathcal{S} = \{sim(E_1(w), E_2(w)) \mid w \in T\}$. Then, we rank the target words according to the distance, computed as: $1- \mid sim(E_1(w), E_2(w)) \mid$.

[1] Where v_j is the vector representation for the word generated by E_j and j is the time period.

2.2 Subtask 1: Gaussian Mixture Clustering

Subtask 1 requires a further step: given $\mathcal{S}$, the set of target similarities, we need to predict the target labels. The aim is to assign either of the two classes, 0 (stable) or 1 (change), to each target word of a given language. Once we compute the set of target similarities $\mathcal{S}$, we want to find a way to assign the corresponding label. We assume that low similarities suggest changing words and high similarities indicate stable words.

Gaussian Mixture Models (GMMs) allow us to build probabilistic models for representing the Gaussian distribution of stable and changing targets. We use GMMs[2] to model the density of the distributions of the similarities of targets as a weighted sum of two Gaussian densities (Huang et al., 2017):

$$f(\mathcal{S}) = \sum_{m=0}^{M} \pi_m \phi(\mathcal{S}|\mu_m, \Sigma_m) \tag{1}$$

where M is the number of mixture components, $\phi(\mathcal{S}|\mu_m, \Sigma_m)$ is the Gaussian density with mean vector μ_m and covariance matrix Σ_m, and π_m is the prior probability for the m-th component. Additional constraints can be applied to the covariance matrix in Eq. 1. In our experiments, we allow each component to have its own covariance matrix.

For our purpose, we speculate that the distribution of target similarities is a mixture of two densities, i.e. representing the stable and changing words. Consequently, we fix the number of the mixture components in the GMMs to two. We initially randomly assign a label (stable/changing) to each density distribution. Let μ_0 and μ_1 be the means of the two Gaussians associated with the "stable" and "changing" labels respectively. If $\mu_0 < \mu_1$ (i.e. the similarity mean of the distribution labelled as "stable" is lower than the mean of distribution labelled as "changing"), we invert the labels. Alg. 1 can be used to properly label each word of the target vocabulary.

Algorithm 1: Assign labels

```
input  : $\mathcal{S}$
output : labels
$\mathcal{N}(\mu_0, \sigma_0), \mathcal{N}(\mu_1, \sigma_1), labels \longleftarrow GaussianMixtures(\mathcal{S})$;
if $\mu_0 < \mu_1$ then
    $labels \longleftarrow 1 - labels$;
end
```

In order to set the best parameters for each language and model, we rely on the GMMs log likelihood, which is generally used for estimating the clusters quality:

$$\ell(\theta \mid \mathcal{S}) = log \sum_{m=0}^{M} \pi_m \phi(\mathcal{S} \mid \mu_m, \Sigma_m) \tag{2}$$

where θ are the parameters of the GMM. For each language, we select the best model configuration to submit at the challenge using the GMMs log likelihood $\ell(\theta \mid \mathcal{S})$. This means that hyper-parameters across different languages are tuned using GMMs log likelihood. We improperly use this approach for choosing parameters across different models (different sets of similarities $\mathcal{S}$), as we do not have validation set for tuning the parameters. We will investigate this limitation as future work. The selected models and hyper-parameters are reported in Tab. 1. In particular, we use cosine similarity, Pearson correlation and Neighborhood similarity for computing the targets similarities in $Overall_{CS}$, $Overall_{PC}$ and $Overall_{NS}$ runs, respectively. In $DW2V$ and TRI runs we use always cosine similarity.

[2] https://scikit-learn.org/stable/modules/generated/sklearn.mixture.GaussianMixture.html

3 Experimental Setup

In all the runs, we do not pre-process data and we use a context window size of 5 while analyzing sentences. The TR model[3] has been adopted into its original implementation[4], as the TRI[5] approach and $DW2V$[6] one. For runs involving TRI, we experimented with a varying vector size from 200 to $1,000$. Moreover, we investigated (1) the initialization of the count matrix at time j with the matrix at time $j-1$, (2) the contribution of positive-only projections, and (3) the application of PPMI weights, as explained in QasemiZadeh and Kallmeyer (2016). For $DW2V$, we use the parameter setting proposed in Yao et al. (2018). We set $\lambda = 10$, $\tau = 50$, $\gamma = 100$, $\rho = 50$ and experimented with a number of iterations from one to five. As vocabulary, we kept the top 50,000 most frequent tokens for both TRI and $DW2V$. In the TR runs, we set the vector size to 100, and we experimented eight iterations for English and Latin, and four for German and Swedish. We use 20 negative samples, keeping only the tokens that occur at least 10 times. All the other parameters used for configuring the models are reported in Tab. 1.

Run	Configuration	English	German	Latin	Swedish
$Overall_{CS}$	Model Parameters	DW2V it=3	Collocation -	DW2V it=3	DW2V it=4
$Overall_{PC}$	Model Parameters	DW2V it=3	DW2V it=4	DW2V it=3	DW2V it=4
$Overall_{NS}$	Model Parameters	DW2V it=3	DW2V it=1	DW2V it=3	DW2V it=4
TRI	Parameters	k= 400 pw=False	k=1000 pw=True	k=1000 pw=True	k=1000 pw=True
$DW2V$	Parameters	it=3	it=4	it=3	it=4

Table 1: Hyper-parameters and models selected for each run. *it* is the number of iterations, *k* is the embedding size, *pw* the use of PPMI weights

4 Results

SemEval 2020 Task 1 provide three baselines, namely Freq. Baseline, which uses the absolute difference of the normalized frequency in the two corpora as a measure of change; Count Baseline, which implements the model described in (Schlechtweg et al., 2019); and Maj. Baseline that always predicts the majority class. Tab. 2 reports the main results obtained by the different models. It shows the results obtained from the official submissions at the challenge and the results obtained by the TR approach performed during the post-evaluation phase. The results obtained for Subtask 1 are reported using the accuracy metric, while for Subtask 2, the Spearman's rank-order correlation coefficients are used.

Considering the results of the evaluation phase, the models show inconsistent behaviors. TRI showed the best performance when considering "all the languages" for both Subtasks, although in Subtask 1 it is not able to overcome *Count Baseline* and *Maj. Baseline*. Focusing on Subtask 1, if we consider each language in isolation, we see that $DW2V$ gives the best results for English[7] while $Overall_{PC}$ is our best system for German language, although it is not able to overcome *Count Baseline*. $Collocation$ is the best system for Latin (although outperformed by *Freq. Baseline*) while TRI is our best system for Sweden language. In Subtask 2, the best English score was reported by $Overall_{NS}$. $Overall_{CS}$ ($Collocation$) performed the best in German language. For Latin and Sweden, TRI provided the best results, and interestingly, it is one of the few systems that did not generate a negative correlation, although outperformed by $CountBaseline$ in Latin language.

At the end of the challenge, when the labelled test set was released, we performed more experiments reported in the *post-evaluation* row. In this phase, we run an additional system, TR, which outperformed

[3] We add this model during the post-evaluation.

[4] https://github.com/Garrafao/TemporalReferencing

[5] https://github.com/pippokill/tri

[6] https://github.com/yifan0sun/DynamicWord2Vec

[7] Please, note that for EN, LA and SW $Overall_{CS}$ and $DW2V$ coincide

all the previous reported approaches, including all baselines. The only exception is for Latin, in which for Subtask 1 *Freq. Baseline* achieves 0.650 accuracy in comparison to 0.525 of TR. Comparing TR and TRI, which are both based on implicit alignment, the former is a prediction-based model while the latter is a count-based one. Moreover, TR creates a temporal word embedding only for the target words rather than for the whole vocabulary. Consequently, this results in better word embeddings for all the words in the vocabulary that do not have a temporal reference, because they are represented by using all occurrences in C_1 and C_2. We suppose that these differences allow TR to achieve better results than the other models.

	Subtask 1					**Subtask 2**				
System	**All Lang.**	**EN**	**DE**	**LA**	**SV**	**All Lang.**	**EN**	**DE**	**LA**	**SV**
$Freq.Baseline$	0.439	0.432	0.417	***0.650***	0.258	-0.083	-0.217	0.014	0.020	-0.150
$CountBaseline$	*0.613*	0.595	*0.688*	0.525	0.645	0.144	0.022	0.216	*0.359*	-0.022
$Maj.Baseline$	0.576	0.568	0.646	0.350	0.742	NaN	NaN	NaN	NaN	NaN
$Overall_{CS}$	0.509	*0.622*	0.500	0.400	0.516	0.111	0.252	*0.415*	-0.183	0.041
$Overall_{PC}$	0.533	0.595	0.646	0.375	0.516	0.056	0.272	0.168	-0.135	-0.080
$Overall_{NS}$	0.508	0.568	0.542	0.375	0.548	0.035	*0.298*	-0.059	-0.179	0.078
$Collocation$	0.513	0.486	0.500	0.550	0.516	0.273	0.144	*0.415*	0.194	0.340
$DW2V$	0.541	*0.622*	0.625	0.400	0.516	0.098	0.252	0.366	-0.183	-0.041
TRI^*	0.554	0.486	0.479	0.475	*0.774*	0.296	0.211	0.337	0.253	*0.385*
TR *(post-eval.)*	**0.704**	**0.703**	**0.812**	0.525	**0.774**	**0.496**	**0.304**	**0.722**	**0.395**	**0.562**

Table 2: Results obtained by our models during the official competition and during the post-evaluation phase. For the Subtask 1 the results represent the accuracy score. Spearman's rank-order correlation coefficients are used for the Subtask 2. TRI^* is the official submission in the evaluation phase since it obtained the best score in the Subtask1.

	Subtask 1					**Subtask 2**				
System	**All Lang.**	**EN**	**DE**	**LA**	**SV**	**All Lang.**	**EN**	**DE**	**LA**	**SV**
TRI	.55	.49	.48	.47	**.77**	.30	.21	.34	.25	.38
NLPCR	.58	**.73**	.54	.45	.61	.29	**.44**	.45	.15	.11
UWB	**.69**	.62	**.75**	.70	.68	.48	.37	**.70**	.25	**.60**
Jiaxin & Jinan	.66	.65	.73	**.70**	.58	**.52**	.32	.72	.44	.59
Life-Language	.68	.70	**.75**	.55	.74	.22	.30	.21	-.02	.39
RPI-Trust	.66	.65	**.75**	.50	.74	.43	.23	.52	**.46**	.50

Table 3: Best results obtained in Subtask 1 for each language: TRI is compared with results submitted by all participants to the SemEval-2020 Task 1.

Tab 3 reports the best results for each language among all participants to Task 1. UWB obtains the best result for German language, tied with Life-Language and RPI-Trust, and the best average result over all languages. Our official submission TRI gives the best result in the Swedish language, whereas Jiaxin & Jian results first for Latin and NLPCR for English language. In Subtask 2, NLPCR and UWB obtain the best results for English and German languages respectively, confirming results obtained in Subtask 1. Concerning the Latin language, also Jiaxin & Jian confirm results obtained in Subtask 1, outperformed only by RPI-Trust, while in Swedish UWB obtain the best result. In general, each system achieved the best performance in one language while performing differently on the remaining others.

During the post-evaluation, we decided to investigate also the role of GMMs for class labeling (Sec. 2). We compared GMMs with semi-manual thresholds $\mu_{\mathcal{S}}$, $\mu_{\mathcal{S}} - \sigma_{\mathcal{S}}$, $\mu_{\mathcal{S}} + \sigma_{\mathcal{S}}$ and Winsorizing (Kokic and Bell, 1994) computing $\mu_{\mathcal{S}}$ and $\sigma_{\mathcal{S}}$ on data provided for Subtask 1, where $\mu_{\mathcal{S}}$ and $\sigma_{\mathcal{S}}$ are the mean and the standard deviation computed on the similarity set $\mathcal{S}$. Figure 1 reports the different accuracy scores obtained by the five methods for the TRI, $Collocation$, $DW2V$, TR approaches. The scores for the GMMs strategy are close to those obtained by $\mu_{\mathcal{S}}$ for TRI and Collocation. While GMMs outperforms $\mu_{\mathcal{S}} + \sigma_{\mathcal{S}}$ in every run, $\mu_{\mathcal{S}} - \sigma_{\mathcal{S}}$ seems to work better than GMMs except that in TR. Winsorizing works better than GMMs in TRI and $Collocation$. GMMs outperforms Winsorizing in $DW2V$ and TR. These results are not clear enough to advocate for a specific threshold. Consequently, further analysis will be part of future work in order to understand what is the better threshold that could be included in the GMMs process.

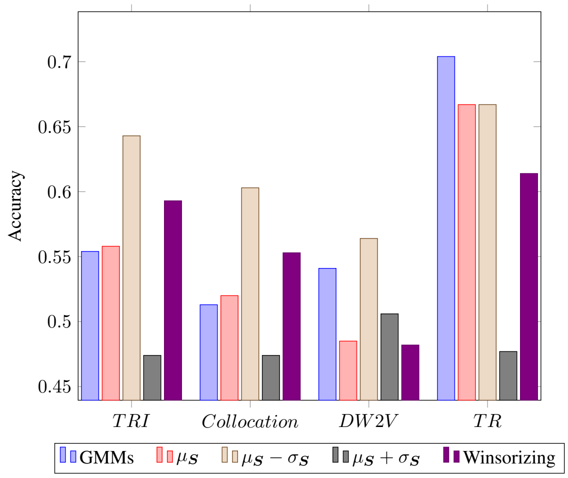

Figure 1: Accuracy scores in Subtask 1 using different class labeling strategies: GMMs, $\mu_{\mathcal{S}}$, $\mu_{\mathcal{S}} - \sigma_{\mathcal{S}}$, $\mu_{\mathcal{S}} + \sigma_{\mathcal{S}}$ and Winsorizing using mean and standard deviation.

5 Conclusions

We described the runs we submitted to the SemEval-2020 Task 1: Unsupervised Lexical Semantic Change Detection. This paper has two main contributions. We reported a comparison of some of the most recent approaches to model lexical semantic change with temporal word embeddings, and we experimented with an automatic unsupervised procedure to classify changing and stable words. Results show that implicit alignment works generally better in modelling the lexical semantic change. In future works we plan to carry out an analysis on unlemmatised corpora and gauge a better understanding of the impact of Gaussian Mixture Clustering for unsupervised lexical semantic change detection.

6 Acknowledgements

This research was partially conducted with the financial support of Science Foundation Ireland under Grant Agreement No. 13/RC/2106 at the ADAPT SFI Research Centre at Dublin City University. The ADAPT SFI Centre for Digital Media Technology is funded by Science Foundation Ireland through the SFI Research Centres Programme and is co-funded under the European Regional Development Fund (ERDF) through Grant # 13/RC/2106.

References

Reem Alatrash, Dominik Schlechtweg, Jonas Kuhn, and Sabine Schulte im Walde. 2020. Ccoha: Clean corpus of historical american english. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 6958–6966.

Pierpaolo Basile, Annalina Caputo, and Giovanni Semeraro. 2015. Temporal random indexing: A system for analysing word meaning over time. *Italian Journal of Computational Linguistics*, 1(1):55 – 68.

Pierpaolo Basile, Giovanni Semeraro, and Annalina Caputo. 2019. Kronos-it: A dataset for the Italian semantic change detection task. In *CEUR Workshop Proceedings*, volume 2481.

Lars Borin, Markus Forsberg, and Johan Roxendal. 2012. Korp-the corpus infrastructure of spräkbanken. In *LREC 2012*, pages 474–478.

Haim Dubossarsky, Simon Hengchen, Nina Tahmasebi, and Dominik Schlechtweg. 2019. Time-Out: Temporal Referencing for Robust Modeling of Lexical Semantic Change. In *57th Annual Meeting of the Association for Computational Linguistics*, pages 457–470. Association for Computational Linguistics (ACL), sep.

William L. Hamilton, Jure Leskovec, and Dan Jurafsky. 2016. Diachronic word embeddings reveal statistical laws of semantic change. In *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*, volume 3, pages 1489–1501, may.

Tao Huang, Heng Peng, and Kun Zhang. 2017. Model selection for gaussian mixture models. *Statistica Sinica*, pages 147–169.

PN Kokic and PA Bell. 1994. Optimal winsorizing cutoffs for a stratified finite population estimator. *Journal of Official Statistics*, 10(4):419.

Barbara McGillivray and Adam Kilgarriff. 2013. Tools for historical corpus research, and a corpus of latin. *In Paul Bennett, Martin Durrell, Silke Scheible, and Richard J. Whitt, editors, New Methods in Historical Corpus Linguistics, Tübingen. Narr.*

Behrang QasemiZadeh and Laura Kallmeyer. 2016. Random positive-only projections: PPMI-enabled incremental semantic space construction. In **SEM 2016 - 5th Joint Conference on Lexical and Computational Semantics, Proceedings*, pages 189–198.

Dominik Schlechtweg, Anna Hätty, Marco Del Tredici, and Sabine Schulte im Walde. 2019. A wind of change: Detecting and evaluating lexical semantic change across times and domains. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 732–746.

Dominik Schlechtweg, Barbara McGillivray, Simon Hengchen, Haim Dubossarsky, and Nina Tahmasebi. 2020. SemEval-2020 Task 1: Unsupervised Lexical Semantic Change Detection. In *To appear in Proceedings of the 14th International Workshop on Semantic Evaluation*, Barcelona, Spain. Association for Computational Linguistics.

Philippa Shoemark, Farhana Ferdousi Liza, Dong Nguyen, Scott Hale, and Barbara McGillivray. 2019. Room to Glo: A Systematic Comparison of Semantic Change Detection Approaches with Word Embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 66–76. Association for Computational Linguistics.

Nina Tahmasebi, Lars Borin, and Adam Jatowt. 2018. Survey of computational approaches to lexical semantic change. *arXiv preprint arXiv:1811.06278.*

Deutsches Textarchiv. 2017. Grundlage für ein referenzkorpus der neuhochdeutschen sprache. herausgegeben von der berlin-brandenburgischen akademie der wissenschaften. `http://www.deutschestextarchiv.de/`.

Zijun Yao, Yifan Sun, Weicong Ding, Nikhil Rao, and Hui Xiong. 2018. Dynamic word embeddings for evolving semantic discovery. In *WSDM 2018 - Proceedings of the 11th ACM International Conference on Web Search and Data Mining*, volume 2018-Febua, pages 673–681.

Berliner Zeitung. 2018. Diachronic newspaper corpus published by staatsbibliothek zu berlin. `http://zefys.staatsbibliothek-berlin.de/index.php?id=155`.