Benchmarks for Unsupervised Discourse Change Detection

Quan Duong, Lidia Pivovarova and Elaine Zosa

University of Helsinki, Finland

Abstract

The main motivation for this work lies in the need to track discourse dynamics in historical corpora. However, in many real use cases ground truth is not available and annotating discourses on a corpus-level is hardly possible. We propose a novel procedure to generate synthetic datasets for this task, a novel evaluation framework and a set of benchmarking models. Finally, we run large-scale experiments using these synthetic datasets and demonstrate that a model trained on such a dataset can obtain meaningful results when applied to a real dataset, without any adjustments of the model.

Keywords

Discourse dynamic, News monitoring, Synthetic data, Quantitative evaluation, Neural network, Unsupervised, Pattern detection, Pivots detection, Sequence to Sequence

1. Introduction

Various computational methods, from keyword extraction to topic modelling, have been established to facilitate discourse analysis. However, studying *discourse dynamics*—the change in prevalence of certain topics, opinions, and attitudes over time—is a novel and challenging research area yet to be developed.

The term ”discourse” has many definitions across humanities and social disciplines; it could be understood either as a property of a corpus as a whole or a property of a single text and its structure. In this paper we treat discourse as a *corpus property*. A fine-grained structure of particular documents is irrelevant for our research question and ignored in the experiments. Discourse change can only be found in a *diachronic corpus*, i.e. corpus that contains data from several consecutive time periods.

Thus input for our methods is a collection of texts, split into multiple time periods. The task breaks up into three following **sub-tasks**:

1. to detect, whether a certain discourse in this collection is *non-stable*, e.g. increases or decreases;
2. to find *a subset of documents* that belong to this discourse;
3. to find *pivot point* in the timeseries, i.e. time points where non-stable behaviour of the discourse starts and ends.

HistoInformatics 2021 – 6th International Workshop on Computational History, September 30, 2021, online

quan.duong@helsinki.fi (Q. Duong); lidia.pivovarova@helsinki.fi (L. Pivovarova); elaine.zosa@helsinki.fi (E. Zosa)

0000-0001-6566-368X (Q. Duong); 0000-0002-0026-9902 (L. Pivovarova); 0000-0003-2482-0663 (E. Zosa)

Historical research questions are generally complex and involve a lot of uncertainty, thus the ground truth needed for quantitative evaluation is usually unavailable. Quite often research deals with a specific use case, focusing on a single non-annotated dataset without a proper split into training and test subsets. Thus, finding training and evaluation data for this task is currently not possible. As far as we know, there does not yet exist a diachronic corpora annotated with discourses.

To overcome this difficulty, we propose an evaluation framework using multiple synthetic datasets. The idea is to exploit manually assigned article categories, available in many news corpora. Distinct periods and spikes in the data could be mimicked by sampling from a certain label according to a certain pattern, while all other categories are sampled randomly. Synthetic datasets allow for training and evaluation models able to find a subset of documents that are related to the same theme and follow the pattern, without looking at the manually assigned labels. The source code for this study is available on Github, which is freely accessible for further development.[1]

2. Background

Discourse dynamics has been a topic of several multidisciplinary studies that apply NLP to historical or social science research questions. Quite often these studies lean on topic modelling [1, 2, 3, 4], though others use techniques, such as language models and clustering [5, 6]. Each of these studies deal with a complex research question, such as ”immigration discourse” or ”nation building”, and the suitability of the applied methods is assessed only qualitatively, using close reading or background knowledge of the field.

There were several attempts within the NLP field to model discourse change, by the means of unsupervised topic models, such as dynamic topic models [7, 8, 9]. However, these models are often evaluated qualitatively and as a result, the applicability of the models remains unclear especially for research questions that go beyond localizing well-known historical events in time. Any model has certain limitations, that are rarely articulated [10]; and quite often a basic LDA model is preferred to more sophisticated models [11].

Another task relevant to diachronic change is lexical semantic change detection [12, 13, 14]. In this task, manual data annotation is extremely challenging [15] and synthetic datasets are commonly used [16, 17, 18, 19].

This paper is positioned in between the aforementioned fields. The research question, automatic discourse change detection, is motivated by the needs of humanities scholars but the point of view is methodological: we propose an evaluation framework rather than investigate any particular use case. The evaluation procedure is based on extensive experiments on multiple synthetic datasets, an approach adopted from the closely related task of lexical semantic shift detection. We are unaware of any work approaching discourse dynamics from this angle and run experiments similar to ours, either in NLP or digital humanities literature.

[1] https://github.com/ruathudo/detangling-discourses

3. Synthetic Datasets

3.1. Yle News Corpus

The synthetic datasets are created from a corpus of news articles published from 2011 to 2018 by the Finnish broadcasting company Yle. The corpus is distributed through Finnish Language Bank (Kielipankki)[2] and is freely available for research use[3].

Each article belongs to one major category and one or more sub-categories. To create the synthetic dataset, we take articles that belong to well-separated major categories. We found 12 categories in the corpus that are suitable for this purpose: *autot* (cars), *musiikki* (music), *luonto* (nature), *vaalit* (elections), *taudit* (diseases), *työllisyys* (employment), *jääkiekko* (hockey), *kulttuuri* (culture), *rikokset* (crimes), *koulut* (schools), *tulipalot* (fires) and *ruoat* (food). These categories have a relatively balanced number of articles and cover distinct subjects, which is appropriate for creating a clean dataset for evaluation. However, a single article may cover several themes–this introduces additional noise in the synthetic datasets and thus a desirable property. After limiting our data to these 12 categories, we end up with a reduced corpus of 207,881 articles.

3.2. Discourse Change Patterns

The datasets for our experiments are sampled to simulate pre-defined patterns of discourse change. Each dataset consists of 100 artificial time points. For each time point, we randomly sample documents from several categories in such a way that one category follows a non-stable pattern—for example, increases over time—while all others remain stable, i.e. randomly oscillating.

We define six possible patterns of discourse behaviour across time, which are illustrated in Figure 2:

- **Up**: The number of articles belonging to a discourse starts increasing at certain time point, and grows until some later point, when it becomes stable.
- **Down**: The number of articles decreases between two time points, then becomes stable.
- **Up - Down**: The number of articles increases, then decreases, then becomes stable.
- **Down - Up**: The number of articles decreases, then increases, then becomes stable.
- **Spike Up**: The trend behaves similar to the Up-Down pattern but spikes are more steep and could appear several times
- **Spike Down**: The trend behaves similar to the previous one but in reversed way.

In addition we use a **Stable** pattern, with no significant change in discourse prevalence over time.

We randomly select one target category and then for this category randomly select one of the six non-stable patterns. For the target category, in each time point *t*, we sample a number of articles *n* so that the timeline follows a randomly selected pattern. We use 100 time points. While generating these sequences, we also randomly assign the pivot points when the non-stable

[2] http://urn.fi/urn:nbn:fi:lb-2017070501

[3] According to the license we cannot redistribute datasets derived from these data. Upon acceptance we will publish our code, which ensures reproducibility of our experiments, including dataset generation.

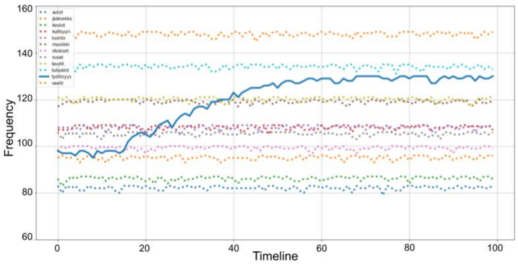

Figure 1: A sample experiment with 1 increasing category (Up) and 11 stable categories.

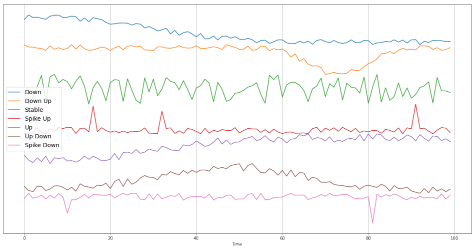

Figure 2: Seven patterns used to emulate discourse dynamics in the synthetic datasets.

pattern starts and ends, which is necessary for sub-task 3. Before and after start and end point the timeseries follows a stable pattern. Then we sample data from the remaining 11 categories, which all follow the stable pattern.

Two functions are used as basic components for discourse change: *sigmoid* or *Gaussian*. The sigmoid function is used to sample the **Up** and **Down** patterns: we assume that a novel discourse slightly increases or decreases at the beginning, then speeds up in the middle and then gradually slows up before becoming stable again, which is exactly how the sigmoid function behaves. Thus, the discourse change forms a S-curve, which is a natural shape in many language-change processes [20].

More concretely, a number of articles in Up and Down patterns follows the formula:

$$S_i = N + \frac{1}{1 + e^{-k \times (T_i - (T_{end} - T_{start})/2)}} \times N \times R$$

where T_{start} and T_{end} are the time points where the pattern starts and ends, respectively; S_i is the number of articles at time point $T_i \in [T_{start}, T_{end}]$; N is the number of articles before the starting point, R is the change rate for the pattern, arbitrarily selected between 0.3 and 0.8, and k is the parameter that defines how the change is distributed along the time. With a large k the S-curve is steep, with a slow change at two ends of the range, and a rapid change in the middle. We set $k = 0.1$ to form a gradual change.

The Gaussian function is used for the **Up - Down** and **Down - Up** patterns which have a bell shape. By modifying the mean and standard deviation of the Gaussian, we produce different forms of the bell shape, depending on the amount of data and the number of time points. We sample the bell pattern using the following formulas:

$$S = \Phi_{\mu,\sigma^2}(X) \quad \mu = (T_{end} - T_{start})/2 \quad \sigma = (T_{end} - T_{start})/k$$

$$S_i = N + \frac{T_i - \min(S)}{\max(S) - \min(S)} \times N \times R$$

The S is a set of values drawn from Probability Density Function Φ of Gaussian distribution to form a bell curve. The Gaussian distribution has μ is the middle point in time range, and σ depends on a parameter k in the equation. A large k will create a shape with a sharp peak in the middle . From our experiments, we found that $k = 5$ gives a smooth changing pattern. After having S sampled in the bell shape, we can calculate the number of articles for each time point, however, S needs to be rescaled to have a consistent input in range $[0, 1]$ using min-max scaling as in the last equation.

Another pattern that uses the Gaussian distribution is *multiple periods* up or down spikes. This pattern will have a very short range of beginning and ending time points which is similar to a pine shape.

Figure 1 shows an example dataset: the Up pattern is used. As can be seen in the figure, random noise is added to all patterns, so small spikes are visible for all categories, including stable ones. The input to our trend-detection model are raw texts, while categories are hidden. In this way we try to emulate a realistic situation where many themes are oscillating in the news at the same time and only a few of them display a certain increasing or decreasing trend.

4. Method

In all our experiments we use two major steps: (i) building a timeseries from textual data; (ii) analysing the timeseries to classify them as either stable or unstable and finding pivot points.

We split a document collection into clusters using either k-means or LDA and then build a separate timeseries for each cluster. Then each timeseries is processed separately to detect whether it is stable or non-stable. For this step we use a sequence-to-sequence neural network, which is trained to jointly predict non-stable trends and pivot points. For comparison, we use linear regression as a baseline.

4.1. Building Timeseries

Clustering We use doc2vec model [21] to obtain document representations. The inferred document vectors are then clustered using k-means. Clustering is run independently for each of the 1000 datasets, so each dataset simulates a single independent use case. We set the number of clusters to 20 for all our datasets. Thus, we do not use our prior knowledge about the number of categories used. Moreover, perfect clustering is not possible with this setting since the number of clusters is bigger than the number of categories used to generate a dataset. The rationale behind this is that when working with real data we would not know the number of discourses in the collection. The method we propose does not aim at perfect clustering, only on detection of non-stable trends.

Clustering is done jointly for all time points in the dataset. Then we built a timeline for each cluster, by counting the number of documents from each cluster at each time point. Timelines are scaled to [0,1] interval so that the biggest value for each timeline is always 1.

LDA We use topic modelling as an alternative to k-means. We train a separate LDA model for each synthetic dataset and train with 20 topics to align with k-means.

The timeline on top of LDA is built using soft clustering, since an article can have more than one topic. To count the number of documents that belong to a certain topic, we use all documents where the topic probability is higher than 0.25. If no topic has a probability above the threshold, we assign the document to the topic with the highest probability. Similar to k-means, topic timelines are scaled to [0, 1] range.

Training Data The cluster-based timeseries, described above, are used only to construct the validation set. To train a neural network, we directly sample the patterns with noise to mimic the sequence of frequency in the clustered set. Stable and non-stable timeseries are sampled equally for the training.

The input for our models is a sequence of frequencies. The model produces two outputs: a binary prediction of whether a timeseries is stable or non-stable and a sequence, where the value at each time point is the probability that the time point belongs to a non-stable pattern. In the training data, we set to 1 all values between pattern start and end, while all other values are set to 0. If the timeseries is stable, all values in the output sequence are zeros, which corresponds to zero value for the first output.

4.2. Sequence-to-sequence model

Recurrent Neural Network The model structure is presented in Figure 3. The input to this model is a matrix with the shape $(N, 100)$ where N is the batch size and 100 is the timeseries length. Each example is a sequence of numbers in the range $[0, 1]$.

We use an RNN variant—bidirectional Long Short Term Memory (bi-LSTM)— stacked with one fully connected (FC) layer. The bi-LSTM layer has 256 hidden units.

The following FC layer takes all outputs from the LSTM layer and flatten them as input. Dropout layer is introduced to reduce overfitting. The FC layer is connected to two output layers: one to predict the probability that the input is non-stable and the other to predict a sequence of non-stable point probabilities. Both output layers use the sigmoid activation function to get probability values.

Convolutional Neural Network The CNN is intended for capturing local features for image recognition [22]. Our idea is to use this ability to detect patterns in sequence data. The CNN model is shown in Figure 4. The input and output is the same as one described for RNN . Because our sequence data only has one dimension, the 1D CNN layers are used for feature extraction. We use two stacked convolutional layers with a kernel size of 3. The first layer has 8 output channels while the second one expands to 16 channels. We also have max pooling layers after each convolutional layer. Finally, the output features are flattened and passed to the FC layer, and the rest of the model is organized identically to the RNN model.

Combined Model While RNN is good at handling sequence information, CNN is more suitable for local pattern detection. We leverage the strengths of both models to produce a combined model that might be more robust at pivot point detection. The architecture of the combined model (which we further denote as RCNN) is presented in Figure 5. CNN and bi-LSTM layers are identical to those used in the separate models. Then the hidden state output of the bi-LSTM layer is concatenated with the output of the last convolutional layer, flattened, and passed to the FC layer. After concatenating RNN and CNN outputs, the rest of the model is organized identical to the previous cases.

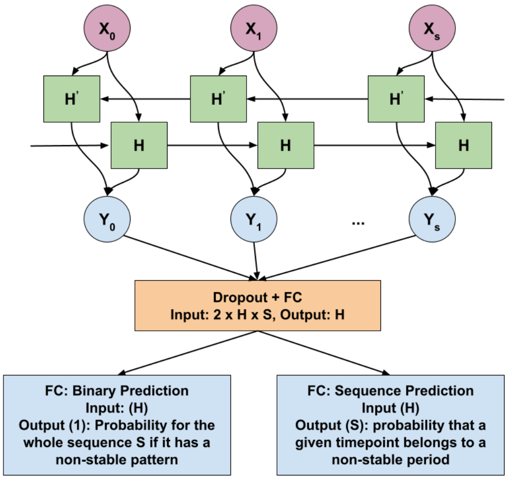
X0
X1
Xs
H'
H'
H'
H
H
H
Y0
Y1
...
Ys
Dropout + FC
Input: 2 x H x S, Output: H
FC: Binary Prediction
Input: (H)
Output (1): Probability for the whole sequence S if it has a non-stable pattern
FC: Sequence Prediction
Input (H)
Output (S): probability that a given timepoint belongs to a non-stable period

Figure 3: RNN network architecture with biLSTM layer. The prediction Y from all timesteps are used for the FC layer.

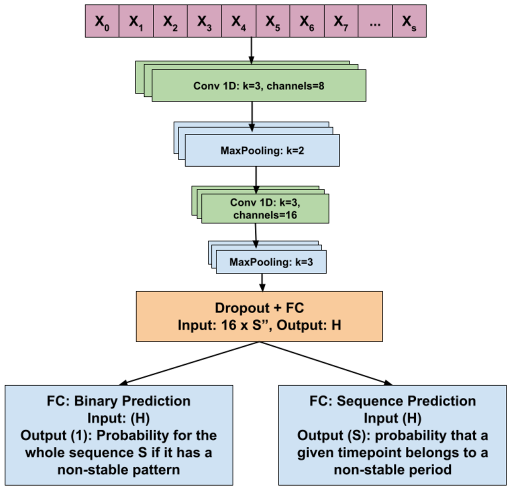
X0
X1
X2
X3
X4
X5
X6
X7
...
Xs
Conv 1D: k=3, channels=8
MaxPooling: k=2
Conv 1D: k=3, channels=16
MaxPooling: k=3
Dropout + FC
Input: 16 x S'', Output: H
FC: Binary Prediction
Input: (H)
Output (1): Probability for the whole sequence S if it has a non-stable pattern
FC: Sequence Prediction
Input (H)
Output (S): probability that a given timepoint belongs to a non-stable period

Figure 4: CNN network architecture. Where k is the kernel size, H is the hidden size, and S" is the length of sequence after going through the convolutional layers.

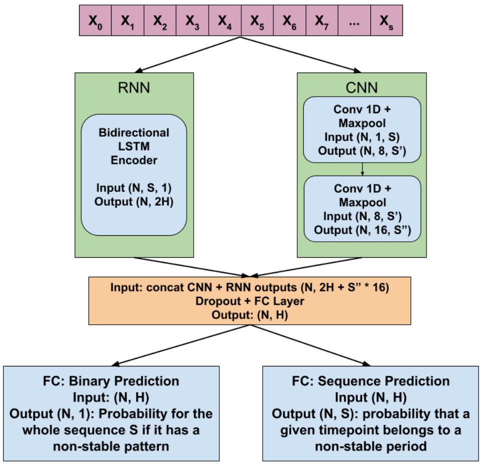

Figure 5: Combined (RCNN) network architecture. Where N is the batch size, H is the hidden size. S is the length of input sequence, S" is the length of CNN output. The hidden states from LSTM are used for the next layer instead of the predicted outputs.

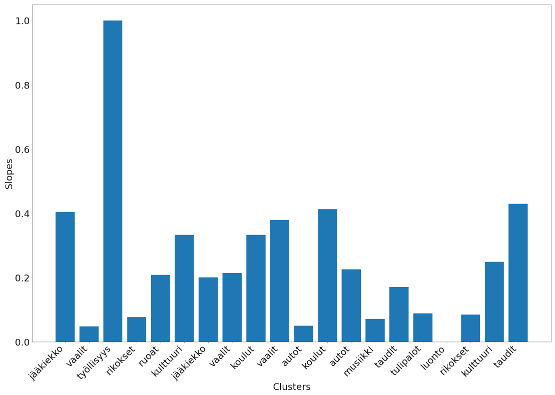

Figure 6: An example dataset, where each cluster, obtained from k-means, is fitted with linear regressions. The normalized slopes are shown in the histogram, with one pattern having significantly higher slope than the others, which indicates non-stable discourse dynamic. Bars are labelled with the major category of the articles within the cluster.

4.3. Baseline

Unlike neural models, our baseline is not independent for each cluster within a dataset. We fit a linear regression model to each of the 20 clusters obtained for the dataset. The absolute slope value of the linear function is normalized to a [0,1] scale, so that the largest normalized slope is equal to 1. A timeseries with a slope above a certain threshold is then classified as non-stable. After preliminary experiments we set this threshold to 0.8 for all datasets.

As an example in Figure 6 we show an output for the dataset presented in Figure 1. In the histogram each bar is a cluster labeled with its major category, i.e. the most frequent category for the clustered articles. The y-axis is the normalized slope value. We see that the category for the biggest bar—*työllisyys*, employment—is the same as one used to build the increasing pattern in Figure 1.

Timeseries identified as non-stable in the previous step are processed using the sliding-window segmentation method to identify pivot points.[4]

5. Evaluation

Category-level Evaluation Category-level accuracy measures how well a model can detect a non-stable category. For each cluster classified as non-stable we define a *major category*, i.e. a category that has a highest count in this cluster. If this major category is the same as the target category used for the dataset generation, then prediction is considered to be correct. For each dataset we calculate a ratio of correct non-stable clusters to all non-stable clusters. If a model does not find any non-stable cluster for the dataset, the accuracy is 0.

Document level Precision, recall and F-measure are used to measure how ”clean” are subsets of documents that form non-stable patterns. For this evaluation, we use all clusters that are predicted to be non-stable, even if their major category is incorrect.

For each non-stable cluster, precision is calculated as a proportion of documents from the target category in this cluster, and recall as the proportion of documents from a non-stable cluster in a target category. The dataset recall and precision are the means of all non-stable cluster measures, and F-measure is computed as the harmonic mean of recall and precision. If all clusters are predicted to be stable then precision, recall and F-measure are set to zero. Then three measures are averaged across datasets.

Time-point level For each cluster that is classified as non-stable, a model must output time points where the non-stable pattern starts and ends. These pivot points segment a timeline into several periods. Then each pair of time points could belong either to the same or to different time periods. RandIndex [23] is computed as a proportion of time-point pairs correctly put either in the same or in the different periods. Shifting a pivot point by 1-2 positions from the true point slightly decreases RandIndex. Radical misplacement or finding an incorrect number of pivot points, however, results in a large drop.

RandIndex is averaged for all non-stable clusters in the dataset. If all clusters are classified as stable, RandIndex is zero. This measure is then averaged across all datasets. Note that this evaluation is orthogonal to the document-level measures, since it is possible to place pivot points to correct positions even if a cluster is noisy or incomplete.

[4] For more detail see the Rupture documentation: https://centre-borelli.github.io/ruptures-docs/user-guide/detection/window/

6. Experiments

Synthetic datasets Table 1 shows results obtained on the synthetic datasets with aforementioned measures. One of the most important results for us is the diversity of the model performance: this means that synthetic datasets are adequately complex and allows for method comparison.

The best performing model is the proposed combination of RNN and CNN (RCNN), which gives the highest results in combination with both k-means and LDA. The best performance is obtained by applying the combined model on top of the k-means output. On top of LDA the combination also yields the highest performance. CNN is better than RNN at non-stable pattern detection. However, RNN yields a much higher RandIndex, which means better at pivot point detection.

The lower performance of LDA compared to k-means needs to be investigated further. Obviously, LDA is much more than just a clustering technique: LDA is a Bayesian model, which outputs topic distribution over documents. In our experiments, this distribution is converted into hard labels and used only indirectly. It is likely that a higher performance could be achieved by other ways of combining topic modelling with neural networks. There is another difficulty when it comes to a rich morphological language like Finnish, where words have many variants and compounds are frequently used [24].

Experiments with real data For a qualitative assessment, we use another Finnish corpus: The Finnish News Agency (STT) Archive [5]. We limit our experiments to the data from years 2007-2008, so does not overlap in time with the YLE dataset. We split the two year data into weeks, excluding the first and the last two weeks, which gives us 100 weeks. Then we can directly apply models trained on synthetic data. The dataset consists of 250,000 documents.

We use our best model for this experiment, i.e. combined RCNN applied on top of k-means with 20 clusters. Out of those 20 clusters, 6 were classified as unstable. We manually scanned the documents within these clusters and found a couple of clusters for which we could find an interpretation. For example, Figure 7 shows a cluster associated with party politics. The date of the Finish parliamentary elections, shown with the green vertical line, is positioned between two automatically determined pivot points—it seems natural that elections are actively discussed some time before and after the event.

We use this experiment to demonstrate that a model trained on synthetic datasets, generated using the proposed procedure, is able to extract meaningful results from real-world data. Whether these results would be relevant for digital humanities or computational social science research is yet to be found in collaboration with domain specialists. Previous collaborations[10] indicate that there is a need for a model able to track discourse change in textual data.

Conclusion

We presented the novel task of automatic detection of discourse change in text collections, which is relevant for historical research and digital humanities in general. However, computational

[5] http://urn.fi/urn:nbn:fi:lb-2019041501

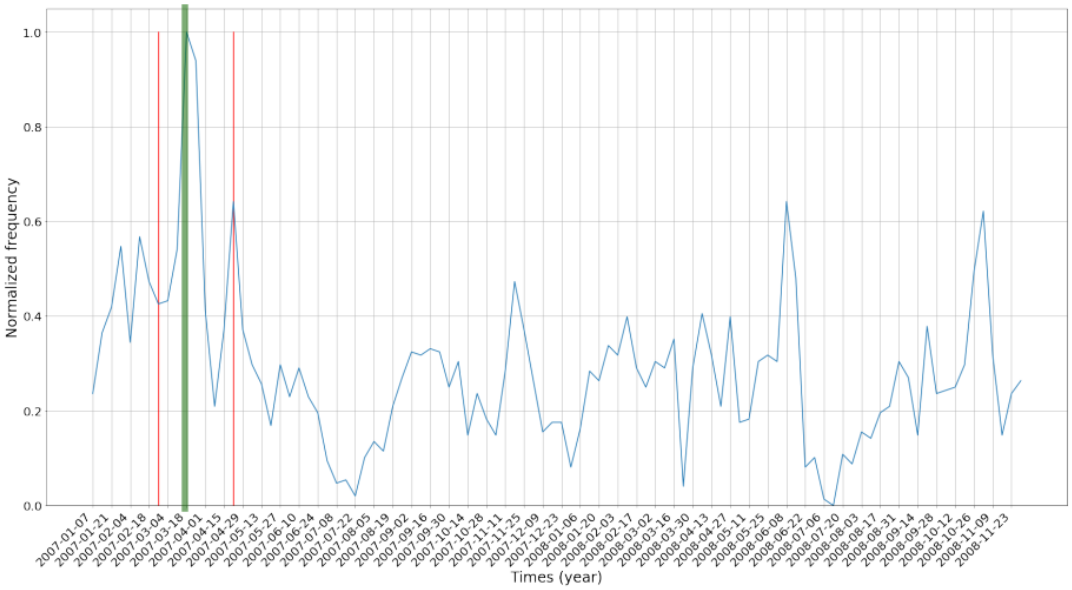

Figure 7: A non-stable cluster obtained on the STT data. Red lines: detected pivot points. Green line: the Finnish Parliamentary elections.

Method		**Category**	**Precision**	**Recall**	**F1**	**Rand**
STEP 1	STEP2	**accuracy**				**index**
k-means	Regression	52.78	43.98	34.73	37.04	42.52
	RNN	73.63	60.55	46.33	50.43	73.17
	CNN	75.17	61.46	46.56	51.49	67.79
	RCNN	**78.43**	**63.77**	**51.69**	**55.22**	**73.26**
LDA	Regression	41.88	31.56	31.26	27.14	41.04
	RNN	38.65	30.48	31.84	27.53	65.04
	CNN	47.73	36.41	33.26	31.87	53.27
	RCNN	51.46	37.22	43.94	36.03	60.43

Table 1: Result obtained on 1000 synthetic datasets

methods to tackle this type of problems are not yet established. One of the main obstacles is the lack of training data and fundamental difficulty to annotate corpus-level phenomena. To overcome this issue we proposed a methodological framework that leans to discourse-change simulation with synthetic data, which allows us to train supervised models. The procedure which we proposed in this paper generates sufficiently complex datasets so that the problem cannot be solved by simple methods, such as regression. This allows for evaluation, comparison, and improvement of the methods, impossible on most typical use cases where ground truth is not accessible.

Acknowledgments

We are grateful to the anonymous reviewers for their valuable comments regarding the positioning of this work and historical discourse studies in general. Though we were unable to address these issues in this paper we take them seriously and will take into consideration in our future work.

This work has been partly supported by the European Union's Horizon 2020 research and innovation programme under grant 770299 (NewsEye) and 825153 (EMBEDDIA).

References

[1] L. Viola, J. Verheul, Mining ethnicity: Discourse-driven topic modelling of immigrant discourses in the USA, 1898–1920, Digital Scholarship in the Humanities 35 (2020) 921–943.

[2] R. Light, J. Cunningham, Oracles of peace: Topic modeling, cultural opportunity, and the Nobel peace prize, 1902–2012, Mobilization: An International Quarterly 21 (2016) 43–64.

[3] T.-I. Yang, A. Torget, R. Mihalcea, Topic modeling on historical newspapers, in: Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities, 2011, pp. 96–104.

[4] D. J. Newman, S. Block, Probabilistic topic decomposition of an eighteenth-century American newspaper, Journal of the American Society for Information Science and Technology 57 (2006) 753–767.

[5] S. Hengchen, R. Ros, J. Marjanen, A data-driven approach to the changing vocabulary of the nation in English, Dutch, Swedish and Finnish newspapers, 1750-1950, in: Proceedings of the Digital Humanities (DH) conference, 2019.

[6] M. Kestemont, F. Karsdorp, M. During, Mining the twentieth century's history from the Time magazine corpus, in: Abstract book of EACL 2014: the 14th Conference of the European Chapter of the Association for Computational Linguistics, 2014, p. 62.

[7] D. M. Blei, J. D. Lafferty, Dynamic topic models, in: Proceedings of the 23rd international conference on Machine learning, 2006, pp. 113–120.

[8] A. B. Dieng, F. J. Ruiz, D. M. Blei, The dynamic embedded topic model, arXiv preprint arXiv:1907.05545 (2019).

[9] X. Wang, A. McCallum, Topics over time: a non-markov continuous-time model of topical trends, in: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, 2006, pp. 424–433.

[10] J. Marjanen, E. Zosa, S. Hengchen, L. Pivovarova, M. Tolonen, Topic modelling discourse dynamics in historical newspapers (2021).

[11] D. Hall, D. Jurafsky, C. D. Manning, Studying the history of ideas using topic models, in: Proceedings of the 2008 conference on empirical methods in natural language processing, 2008, pp. 363–371.

[12] N. Tahmasebi, L. Borin, A. Jatowt, Survey of computational approaches to lexical semantic change, in: Preprint at ArXiv 2018., 2018. URL: https://arxiv.org/abs/1811.06278.

[13] A. Kutuzov, L. Øvrelid, T. Szymanski, E. Velldal, Diachronic word embeddings and semantic

shifts: a survey, in: Proceedings of the 27th International Conference on Computational Linguistics, 2018, pp. 1384–1397.

[14] X. Tang, A state-of-the-art of semantic change computation, Natural Language Engineering 24 (2018) 649–676.

[15] D. Schlechtweg, S. S. im Walde, S. Eckmann, Diachronic usage relatedness (durel): A framework for the annotation of lexical semantic change, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), 2018, pp. 169–174.

[16] A. Tsakalidis, M. Liakata, Sequential modelling of the evolution of word representations for semantic change detection, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 8485–8497.

[17] D. Schlechtweg, S. S. i. Walde, Simulating lexical semantic change from sense-annotated data, in: The Evolution of Language: Proceedings of the 13th International Conference (EvoLang13)., 2020.

[18] P. Shoemark, F. F. Liza, D. Nguyen, S. Hale, B. McGillivray, 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), 2019, pp. 66–76.

[19] A. Rosenfeld, K. Erk, Deep neural models of semantic shift, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), 2018, pp. 474–484.

[20] R. A. Blythe, W. Croft, S-curves and the mechanisms of propagation in language change, Language (2012) 269–304.

[21] Q. Le, T. Mikolov, Distributed representations of sentences and documents, in: International conference on machine learning, 2014, pp. 1188–1196.

[22] J. Schmidhuber, Deep learning in neural networks: An overview, Neural Networks 61 (2015) 85–117. URL: http://dx.doi.org/10.1016/j.neunet.2014.09.003. doi:10.1016/j.neunet.2014.09.003.

[23] C. Truong, L. Oudre, N. Vayatis, Selective review of offline change point detection methods, Signal Processing 167 (2020) 107299.

[24] Q. Duong, M. Hämäläinen, S. Hengchen, An unsupervised method for ocr post-correction and spelling normalisation for finnish, 2020. arXiv:2011.03502.