Project Number: **770299**

NewsEye:

A Digital Investigator for Historical Newspapers

Research and Innovation Action
Call H2020-SC-CULT-COOP-2016-2017

D4.6: Comparative analysis of data between contexts (b) (final)

Due date of deliverable: M45 (31 January 2022)

Actual submission date: 7 January 2022

Start date of project: 1 May 2018 **Duration:** 45 months

Partner organization name in charge of deliverable: UH-CS

Project co-funded by the European Commission within Horizon 2020		
Dissemination Level		
PU	Public	PU
PP	Restricted to other programme participants (including the Commission Services)	-
RE	Restricted to a group specified by the Consortium (including the Commission Services)	-
CO	Confidential, only for members of the Consortium (including the Commission Services)	-

Revision History

Document administrative information		
Project acronym:	NewsEye	
Project number:	770299	
Deliverable number:	D4.6	
Deliverable full title:	Comparative analysis of data between contexts (b) (final)	
Deliverable short title:	Comparative analysis of data between contexts	
Document identifier:	NewsEye-T42-D46-ComparativeAnalysis-b-Submitted-v6.0	
Lead partner short name:	UH-CS	
Report version:	V6.0	
Report preparation date:	7.1.2022	
Dissemination level:	PU	
Nature:	Report	
Lead author:	Elaine Zosa (UH-CS)	
Co-authors:	Lidia Pivovarova (UH-CS), Mark Granroth-Wilding (UH-CS)	
Internal reviewers:	Emanuela Boros (ULR), Sarah Oberbichler (UIBK-ICH)	
Status:		Draft
		Final
	x	Submitted

The NewsEye Consortium partner responsible for this deliverable has addressed all comments received, making changes as necessary. Changes to this document are detailed in the change log table below.

Change Log

Date	Version	Editor	Summary of changes made
19/3/2021	0.1	Elaine Zosa (UH-CS)	Initial draft
25/3/2021	0.2	Mark Granroth-Wilding (UH-CS)	Small changes
29/3/2021	0.3	Lidia Pivovarova (UH-CS)	Added section on detection of discourse change
8/4/2021	1.0	Mark Granroth-Wilding (UH-CS)	Final changes for final draft
12/4/2021	1.1	Elaine Zosa (UH-CS)	Updates following internal reviews
20/4/2021	2.0	Mark Granroth-Wilding (UH-CS)	Final changes before quality management (QM)
26/4/2021	2.1	Mark Granroth-Wilding (UH-CS)	Further updates following QM
30/4/2021	3.0	Antoine Doucet (ULR)	Minor adjustments and submission
17/12/2021	4.0	Elaine Zosa, Lidia Pivovarova, Mikko Lipsanen (UH-CS)	Draft update detailing works performed during the project extension
5/1/2022	5.0	Elaine Zosa, Lidia Pivovarova, Mikko Lipsanen (UH-CS)	Final update following internal reviews and submission to QM
7/1/2022	6.0	Antoine Doucet (ULR)	Minor adjustments and submission

Executive summary

The NewsEye project addressed challenges relating to the exploration of historical news corpora. It made contributions in text recognition, text analysis, natural language processing (NLP) and generation (NLG); in digital newspaper research; in digital humanities; and in history, in terms of analyzing historical assets with new methods.

WP4 aimed to develop and implement methods for *contextualized* and *contrastive content analysis*, carried out *dynamically*, both for use directly by a *Demonstrator* component, which allowed the user to access the tools and collections, and by an *Investigator* component, which performed autonomous analysis and presents its results to the user. In this task, we developed methods and tools for performing this analysis, primarily using *topic models* (TMs).

We report on a collection of tools used for comparative analysis and articles and article collections and their integration into the NewsEye pipeline, taking input from WPs 2 and 3 and producing tools for use in the Demonstrator and the Investigator. These tools are available both for the end-users and for the automated *Personal Research Assistant*.

These methods characterized document sets according to the topics they express and made comparisons *between* document sets using these same topics. Due to the time span covered by the NewsEye collection, we also investigated methods to compare topics and topic prevalence between time slices. Furthermore, we also developed methods for detecting changes in word meaning over time.

The tools provided by WP4 tasks have been made available to both the Demonstrator and the Investigator.

We also report on work we have done with digital humanities collaborators from the universities of Helsinki and Innsbruck, using methods we developed to explore their research questions and investigate the use of unsupervised data-driven text analysis methods for historical research.

Contents

1. Introduction

In this section, we set the work of WP4 in the broader context of NewsEye, describe the goals of Task 4.2 and summarise the work carried out in this task.

1.1. Context within NewsEye

The NewsEye project addressed a number of challenges relating to the exploration of historical news corpora. These involve contributions in several directions:

- in text recognition, text analysis, natural language processing, computational creativity, and natural language generation, particularly with regard to historical newspapers;
- in digital newspaper research, addressing a number of editorial issues like optical character recognition (OCR) noise and article separation;
- in digital humanities, dealing with huge amounts of text material, availability of useful tools and possibilities of searching and browsing; and
- in history, in terms of analyzing historical assets with new methods across different languages corpora.

Central to the project are the *Demonstrator*, a means for a user to explore large collections, and the *Personal Research Assistant* (PRA), a tool to perform autonomous exploratory search of collections to help a user identify content of interest. The PRA consists of the *Investigator*, carrying out autonomous analysis, the *Reporter*, delivering reports on the results to the user, and the *Explainer*, explaining how the results were arrived at and why they may be of interest. The interactions between these components are described by Figure 1.

At the heart of both, the Demonstrator and the Investigator components of the PRA, lies a collection of tools for analysing historical newspaper data, made available in textual form by WP2 (Text Recognition and Article Separation) and enhanced with semantic annotations by WP3 (Semantic Text Enrichment). WP4 provided a set of tools for broad-scale analysis of the collection and analysis of smaller groups of articles in the context of the whole collection. These tools are used both by the user directly (through the Demonstrator) and by the autonomous Investigator.

1.2. Work package 4: Dynamic text analysis

The main objective of WP4 was to develop and implement methods for contextualized and contrastive content analysis, carried out dynamically. In this task, we developed methods and tools for the contrastive aspect of this analysis, primarily using *topic models* (TMs). In this deliverable, we report on the methods we developed for T4.2 concerning the comparative analysis of data between contexts. To this end, we developed methods for ranking top topics in a set of documents, extracting common topics from two document sets and extracting distinctive topics between two document sets. We also investigated methods to track topic evolution in diachronic corpora (corpora that is changing over time) and demonstrated their application on parts of the NewsEye collection.

The goals of these methods were:

- to provide detailed analysis of textual content in a given context in contrast with another context or the background context;

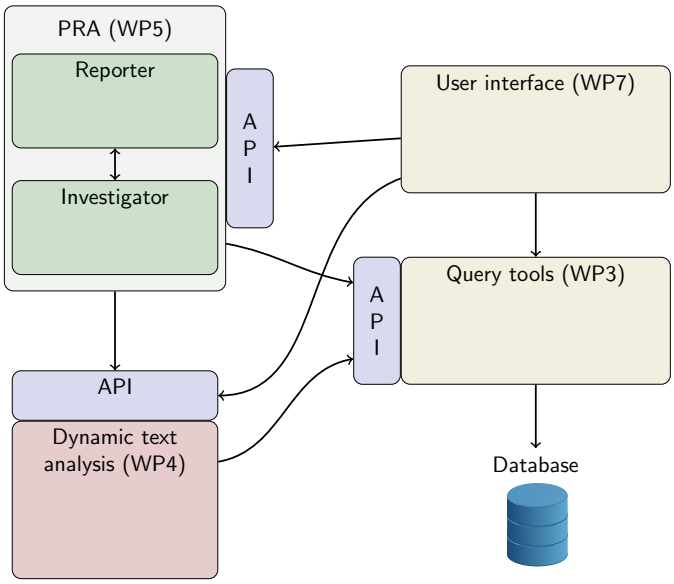


Figure 1: High-level architecture of component systems of NewsEye, showing how WP4 will interact with other WPs to acquire data and provide analyses.

- to support interactive analysis of the content by discovering patterns, topics, trends and viewpoints between contexts;
- to make tools available both for the end users and for the automated Personal Research Assistant (PRA), in both cases via an API.

Contexts for queries can concern a specific group of documents (subcorpora), time period, named entity and other aspects supported by the enriched data and static analyses and indices built in WP2 and WP3, as well as corpus-level textual analysis from the tools of this WP themselves. Since all of these types of contexts can, by means of the metadata query tools provided in other WPs, be reduced to a set of documents corresponding to the context, we defined a **context** as either:

1. *a set of query parameters* to the query interface defined by WP6; or
2. *a list of document IDs* that are the result of the query.

Our work in this task can be divided broadly into two parts. The first part is the analysis of document sets using TMs to extract a group of topics that is descriptive of the document set and provides a basis for distinguishing between sets. We discuss this work in Section 3. The second part, discussed in Section 4, concerns the analysis of *diachronic* corpora – those spanning a long time period. Here we used the dynamic topic models, which take into account the temporal dimension of the data either for estimating topic prominence or during learning the topics themselves. In addition to investigating topic change, we also developed methods for detecting and quantifying changes in word meaning over time.

Section 5 specifies the comparative analysis methods and visualizations of topic evolution implemented in the Investigator and Demonstrator.

Section 6 outlines our work with digital humanities collaborators using TMs and other quantitative textual analysis methods in their research. We describe work we have done on with the DH scholars in the NewsEye consortium on analyzing discourse dynamics with documents from the NewsEye collection.

Finally, Section 7, summarises the work we have done for this task.

2. Introduction to topic models

In this section, we will provide a brief introduction on the topic models (TMs) we used in this task. Since this task is about the application of TMs to compare sets of documents, we will skip over the technical details of how these models infer topics and focus on what these models provide. More details of the models can be found in public Deliverable D4.5.

Standard TMs such as LDA learn topics from a set of documents, where a *topic* is actually a probability distribution over the vocabulary. The group of words that have the highest probabilities in the topic probability distribution are considered to be the most important words for that topic and tell us something about what the topic is about.

Static TMs such as LDA learn static topics, meaning that each topic has a single distribution over the vocabulary. In the case of documents with timestamps covering some time span, such as news articles, we also want to capture dynamic co-occurrence patterns that evolve through time. Dynamic topic models (DTM, [1]) capture themes or topics discussed in a set of time-stamped documents, and how the words related to these topics change in prominence with time. In DTM, the dataset is divided into *time slices* and the model infers topic distributions that evolve in each consecutive time slice.

Aside from topic distributions over the vocabulary, TMs also estimate a probability distribution for each document over the topics. This tells us how much of each document is about a topic. Together, these two sets of distributions, which we will refer to as the *topic-term distributions* and *document-topic distributions*, will be used to statistically analyse and compare individual documents or sets of documents with each other.

Other variants on this basic model structure, such as capturing multiple languages in the same model, are covered by *D4.5*. However, in this work, we have focused on these two types of models only.

3. Comparing document sets with topic models

In this section, we discuss methods to characterise document sets using topics from the trained topic models developed in the public Deliverable D4.5. We provided tools to statistically analyse the contents of a group of documents and then use that to make comparative analyses between groups.

We used topic models as the basis for this task because topic models allow us to relate our statistical analyses back to the concept of *topics* which in this context means a group of related words that is interpretable to the user. Moreover, the unsupervised nature of topic models and the methods discussed here indicate that we can apply them to large collections from different newspapers with minimal adjustments. This is useful for the analysis of historical newspapers because it allows users to draw insights from these collections in a data-driven manner.

We report the results of these methods on a manually annotated document collection, and then we present work done with external collaborators that applied these methods to a large dataset of user-generated comments from an online news forum to gain insights in comment moderation. Our results indicated that while the methods show promising insights using a small document collection and a small

topic model (20 topics), it is more useful on large collections with larger topic models (100 topics). The methods for extracting prominent topics from a dataset and extracting shared and distinctive topics between two datasets are implemented in the Investigator.

We begin by introducing a manually annotated dataset, composed of documents from the NewsEye collection, which we used to develop our methods and then discuss the results of our methods on this dataset.

3.1. Dataset: The Return Migration corpus

To develop the methods for comparing document sets, we used a subcorpus assembled as part of NewsEye by Sarah Oberbichler, a historian from the University of Innsbruck (UIBK-ICH), pertaining to her research question on return migration. The *Return Migration* corpus is composed of 9,642 articles from the newspaper *Arbeiter Zeitung* from 1918. These articles were selected with a fuzzy keyword search (keyword search with *regular expressions*) using the following keywords:

- rückwander.*
- rückkehrer.*
- heimkehr.*
- repatriier.*
- heimgekehrte.*
- (zu)?rück(ge)?kehr(t|(en))?
- heimath?
- flüchtlinge
- auswanderer.*
- ausgewanderte.*
- emigrant.*
- (ö|(oe?))sterreich
- mutterland
- soldat.*
- kriegsgefangene.*
- gefangschaft

Additionally, Oberbichler also annotated a subset of documents from this subcorpus according to their relevance to her research question. The *Relevant* set is composed of 437 articles in the subcorpus deemed to be relevant, while the *Not-Relevant* set is composed of 88 articles that, despite containing the keywords listed above, are not relevant to the research. Most of our experiments were conducted on this subcorpus and these document sets.

Now we demonstrate various comparison methods applied to the two document sets, *Relevant* and *Not-Relevant*. It is worth noting that this may not be typical of the eventual use of these methods in NewsEye, since, whilst *Relevant* is made up of documents relating to a particular coherent theme, *Not-Relevant* may have no coherent theme, or if it does, it may be difficult to distinguish from *Relevant*. A more realistic use case is one where a user has selected two sets that may be assumed to be somewhat coherent and distinct, and wishes to know *how* they differ. In Section 3.8 we present a use case in automatic comment moderation where comments (in this case, each comment is a document) are classified as either *Blocked* or *Non-blocked* according to the moderation rules of an online news forum. We discuss how, in this scenario, our methods provide insights into the characteristics of a *Blocked* or *Non-blocked*

	***Return Migration* topics**
1	frau redner polizei arbeiterschaft richter
2	soldaten truppen november gestern kilogramm
3	ungarn rumänien ungarischen wegen graf
4	deutschen regierung oesterreich deutschland deutsche
5	freitag samstag paris tonnen königsegg
6	kaiser kinder französischen wien französische
7	sofort aufgenommen gesucht lohn gute
8	wien stück preise bürgermeister liter
9	bulgarien italienischen gagisten fabriken bulgarischen
10	telephon tür gasse stock sowie
11	verhandlungen wiener vertreter wien regierung
12	wien petersburg garde petersburger trotzky
13	ganz krieg krieges jahre oesterreich
14	soldaten heimat kriegsgefangenen rußland rückkehr
15	krone wien wäsche adler friedrich
16	montag nachmittags wien samstag gasthaus
17	london mai juli gefangenen berlin
18	frau krone jahre mann wien
19	tel wien viertelj xvi heller
20	arbeiter regierung angestellten arbeit gesetz

Table 1: Topics from the *Return Migration* corpus.

comment.

3.2. Trained topic model

We trained an LDA topic model for 20 topics with the *Return Migration* corpus using the Gensim library[1]. We tokenized, lemmatized and lowercased the corpus, removed some common stopwords and reduced the vocabulary to the top 5,000 words according to their TF-IDF (term frequency-inverse document frequency) score. Table 1 lists the topics learned by the model.

3.3. Topic prominence ranking

We ranked the prominence of topics in a document set using the *mean document-topic distribution* of the set. We computed the distributions independently for all the documents in the set and then average them. It should be noted that by averaging these distributions, we lost some information about the individual documents. This might not be a big problem when a document set is relatively homogeneous (they display similar topic proportions), but it can become an issue when there are outlier documents in a set (we discuss this more in Section 3.7).

In any case, the mean document-topic distribution serves as a way to aggregate the topic distributions of a document set and get the top topics in that set.

[1] https://radimrehurek.com/gensim/models/ldamodel.html

3.3.1. Topic ranking of the *Relevant* document set

The mean document-topic distribution for the *Relevant* document set is:

$$\begin{bmatrix} 0.01 & 0.12 & 0.03 & 0.13 & 0.01 & 0.03 & 0.03 & 0.02 & 0.02 & 0.02 & \dots & \\ & \dots & 0.05 & 0.02 & 0.07 & 0.12 & 0.03 & 0.01 & 0.06 & 0.11 & 0.01 & 0.06 \end{bmatrix}$$

We can use this to rank the topics from most to least prominent. Topic indices are the same as shown in Table 1.

	Top topics for the *Relevant* document set
4	deutschen regierung oesterreich deutschland deutsche (0.13)
2	soldaten truppen november gestern kilogramm (0.12)
14	soldaten heimat kriegsgefangenen rußland rückkehr (0.12)
18	frau krone jahre mann wien (0.11)
13	ganz krieg krieges jahre oesterreich (0.07)
20	arbeiter regierung angestellten arbeit gesetz (0.06)
17	london mai juli gefangenen berlin (0.06)
11	verhandlungen wiener vertreter wien regierung (0.05)
3	ungarn rumänien ungarischen wegen graf (0.03)
6	kaiser kinder französischen wien französische (0.03)
7	sofort aufgenommen gesucht lohn gute (0.03)
15	krone wien wäsche adler friedrich (0.03)
10	telephon tür gasse stock sowie (0.02)
9	bulgarien italienischen gagisten fabriken bulgarischen (0.02)
8	wien stück preise bürgermeister liter (0.02)
12	wien petersburg garde petersburger trotzky (0.02)
19	tel wien viertelj xvi heller (0.01)
16	montag nachmittags wien samstag gasthaus (0.01)
5	freitag samstag paris tonnen königsegg (0.01)
1	frau redner polizei arbeiterschaft richter (0.01)

3.3.2. Topic ranking of the *Not Relevant* document set

We performed the same analysis of the *Not Relevant* document set. Remember that this set of documents contains keywords related to the historian's research, but its articles are deemed not relevant to the research question. The mean document-topic distribution for this set is:

$$\begin{bmatrix} 0.05 & 0.07 & 0.03 & 0.13 & 0.03 & 0.04 & 0.02 & 0.02 & 0.02 & 0.01 & \dots & \\ & \dots & 0.03 & 0.03 & 0.09 & 0.04 & 0.02 & 0.02 & 0.05 & 0.23 & 0.02 & 0.03 \end{bmatrix}$$

We ranked the topics according to this mean vector (topic indexes are the same as shown in Table 1).

	Top topics for the *Not Relevant* document set
18	frau krone jahre mann wien (0.23)
4	deutschen regierung oesterreich deutschland deutsche (0.13)
13	ganz krieg krieges jahre oesterreich (0.09)
2	soldaten truppen november gestern kilogramm (0.07)
1	frau redner polizei arbeiterschaft richter (0.05)
17	london mai juli gefangenen berlin (0.05)
6	kaiser kinder französischen wien französische (0.04)
14	soldaten heimat kriegsgefangenen rußland rückkehr (0.04)
12	wien petersburg garde petersburger trotzky (0.03)
11	verhandlungen wiener vertreter wien regierung (0.03)
20	arbeiter regierung angestellten arbeit gesetz (0.03)
5	freitag samstag paris tonnen königsegg (0.03)
3	ungarn rumänien ungarischen wegen graf (0.03)
9	bulgarien italienischen gagisten fabriken bulgarischen (0.02)
8	wien stück preise bürgermeister liter (0.02)
7	sofort aufgenommen gesucht lohn gute (0.02)
15	krone wien wäsche adler friedrich (0.02)
16	montag nachmittags wien samstag gasthaus (0.02)
19	tel wien viertelj xvi heller (0.02)
10	telephon tür gasse stock sowie (0.01)

3.4. Extracting distinctive topics

We are interested in contrasting document sets according to the topics that are distinctive of them. We extracted the group of topics that distinguish one document set from another by comparing the topic rankings of the different sets as we did in the previous section. For each topic in Set A (here the *Relevant* set), we searched for the rank of that same topic in Set B (here the *Not Relevant* set) and take their difference. We repeated this for all topics. We refer to the resulting vector of this operation as the *topic rank difference*.

The topics that have a large value in the topic rank difference are the topics that are popular in Set A (high rank) but not as popular in Set B (low rank). Since subtraction is not an associative operation, if we want to extract the same information for Set B, we repeat the operation but with the operands in reverse order.

3.4.1. Topic rank difference between *Relevant* and *Not Relevant*

The topic rank for each of the 20 topics for the *Relevant* set is:

$$\begin{bmatrix} 4 & 2 & 14 & 18 & 13 & 20 & 17 & 11 & 3 & 6 & 7 & 15 & 10 & 9 & 8 & 12 & 19 & 16 & 5 & 1 \end{bmatrix}$$

The topic rank for the *Not Relevant* set is:

$$\begin{bmatrix} 18 & 4 & 13 & 2 & 1 & 17 & 6 & 14 & 12 & 11 & 20 & 5 & 3 & 9 & 8 & 7 & 15 & 16 & 19 & 10 \end{bmatrix}$$

The topic rank difference between the *Relevant* set and the *Not Relevant* set:

$$\begin{bmatrix} 1 & 2 & 5 & -3 & -2 & 5 & -1 & 2 & 4 & -3 & 5 & 5 & 7 & 0 & 0 & -7 & 2 & 0 & -7 & -15 \end{bmatrix}$$

We then re-ranked the topics according to this topic rank difference. Since we are only interested in the topics that have a high prominence in a document set, we show only the top 10 topics of the *Relevant* set under the new ranking.

	Top distinctive topics for the *Relevant* document set
20	arbeiter regierung angestellten arbeit gesetz (0.06)
14	soldaten heimat kriegsgefangenen rußland rückkehr (0.12)
2	ungarn rumänien ungarischen wegen graf (0.03)
11	verhandlungen wiener vertreter wien regierung (0.05)
2	soldaten truppen november gestern kilogramm (0.12)
4	deutschen regierung oesterreich deutschland deutsche (0.13)
17	london mai juli gefangenen berlin (0.06)
13	ganz krieg krieges jahre oesterreich (0.07)
18	frau krone jahre mann wien (0.11)
6	kaiser kinder französischen wien französische (0.03)

3.4.2. Topic rank difference between *Not Relevant* and the *Relevant*

If we want to find the distinctive topics of the *Not Relevant* set compared to the *Relevant* set, we compute the topic rank difference but with the operands reversed. This gives us the topic rank difference:

$$\begin{bmatrix} 3 & -1 & 2 & -2 & 15 & 1 & 3 & -5 & 7 & -2 & -5 & 7 & -4 & 0 & 0 & -5 & -5 & 0 & -2 & -7 \end{bmatrix}$$

As before, we re-rank the top 10 topics of this document set according to the topic rank difference.

	Top distinctive topics for the *Not Relevant* document set
1	frau redner polizei arbeiterschaft richter (0.05)
12	wien petersburg garde petersburger trotzky (0.03)
18	frau krone jahre mann wien (0.23)
6	kaiser kinder französischen wien französische (0.04)
13	ganz krieg krieges jahre oesterreich (0.09)
17	london mai juli gefangenen berlin (0.05)
4	deutschen regierung oesterreich deutschland deutsche (0.13)
11	verhandlungen wiener vertreter wien regierung (0.03)
2	soldaten truppen november gestern kilogramm (0.07)
14	soldaten heimat kriegsgefangenen rußland rückkehr (0.04)

3.5. Extracting shared topics

Aside from extracting the distinctive topics of a pair of document sets, we might also want to extract the topics that they have in common. This would be the topics that are prevalent in both document sets.

In this case, we can point-wise multiply the mean topic-document vectors of the two document sets and re-rank the topics such that those with the highest products are highly ranked. The idea here is that the product of the mean topic vectors would tell us that whether *both* (not just one) sets use the same topic highly.

Point-wise addition would not give the same information since there could be cases where one topic has a very high proportion in Set A but a very low proportion in Set B and by adding these proportions together, the topic might be highly ranked when they should not be.

3.5.1. Shared topics of *Relevant* and *Not Relevant*

The point-wise product of the mean document-topic vectors of these two sets:

$$\begin{bmatrix} 0.0 & 0.01 & 0.0 & 0.02 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & \dots & \\ & \dots & 0.0 & 0.0 & 0.01 & 0.01 & 0.0 & 0.0 & 0.0 & 0.03 & 0.0 & 0.0 \end{bmatrix}$$

Now we re-rank the topics according to this product vector.

	Top shared topics with point-wise multiplication
18	frau krone jahre mann wien (0.03)
4	deutschen regierung oesterreich deutschland deutsche (0.02)
2	soldaten truppen november gestern kilogramm (0.01)
13	ganz krieg krieges jahre oesterreich (0.01)
14	soldaten heimat kriegsgefangenen rußland rückkehr (0.01)

3.6. Set operations for extracting distinctive and shared topics

A simpler and more robust approach to extracting topics shared between document sets and topics that distinguish between sets is to treat the top topics of each document sets as a set of items and taking the set intersection to get the set of shared topics and the set difference to get the distinctive topics.

The advantage of this approach is that, beyond a certain threshold, it does not rely on topic probabilities, which can be useful for topic models where document-topic distributions are not as sparse as they are in LDA. This means that all topics are treated as equally important if it passes some threshold (for instance, if it is in the top 10 topics). This approach is also more robust for large numbers of topics when the probabilities are too low that differences between them are no longer significant. A drawback of this approach is that we can no longer rank which topics are most important in the shared or distinctive topics. We demonstrate this method using the top 10 topics of the *Relevant* and *Not-Relevant* documents.

3.6.1. Shared topics using set intersection

The set intersection of the top 10 topics of both document sets are (in no particular order):

	Topics in the set intersection
18	frau krone jahre mann wien
4	deutschen regierung oesterreich deutschland deutsche
13	ganz krieg krieges jahre oesterreich
2	soldaten truppen november gestern kilogramm
17	london mai juli gefangenen berlin
6	kaiser kinder französischen wien französisch
14	soldaten heimat kriegsgefangenen rußland rückkehr
11	verhandlungen wiener vertreter wien regierung

3.6.2. Distinctive topics using set difference

The set difference between the top 10 topics of the *Relevant* vs *Not-Relevant* documents (and vice-versa), in no particular order:

	***Relevant* distinctive topics using set difference**
20	arbeiter regierung angestellten arbeit gesetz
3	ungarn rumänien ungarischen wegen graf

	***Not-Relevant* distinctive topics using set difference**
1	frau redner polizei arbeiterschaft richter
12	wien petersburg garde petersburger trotzky

From these results we can see that of the top 10 topics of each document set, they have eight topics in common which gives us an idea of the degree of overlap in the themes expressed in the two document sets. Topic 20 emerges as a distinctive topic for the *Relevant* documents while Topics 1 and 12 are distinctive of the *Not-Relevant* documents, similar to what we found in Section 3.4. In Section 3.8, we apply this method to a larger topic model (100 topics) and between several document sets.

3.7. Quantifying similarity between documents and document sets

It is useful to have a single measure that expresses the difference between document sets based on their topic distribution. We can compute the Jensen-Shannon divergence (JSD), which measures the distance between probability distributions, between the mean document-topic distributions of two document sets, or we can compute the pairwise JSD between each pair of documents belonging to different sets and get the mean of these pairwise divergences (mean cross-set pairwise divergence).

If we find that the distance between sets is too small then there might not be a lot of difference in terms of topic usage between the document sets and, in that case, our methods for extracting distinctive topics might not give reliable or intuitive results. If there is a larger difference between sets, we can expect that it makes more sense to try to qualify the differences using the methods above to gain insight into *what* the difference is.

JSD between the mean document-topic distributions	0.21
Mean cross-set pairwise JSD	0.51

The low JSD between the mean document-topic distributions means there is a high degree of overlap in the topic usage of the two document sets, in line with our finding in the previous section on the set intersection of the top topics.

Another observation here is that the JSD between the mean topic distributions of the two document sets is a lot lower than cross-set mean pairwise JSD even though these methods both measure the distance between document sets. This might be because the internal divergences for the documents are high, meaning that within the document sets, individual documents are not very similar to other documents in the same set.

Currently, all our experiments on clustering documents according to topics and extracting topics have been on LDA topic models, but the methods should also be applicable to other topic models such as dynamic topic models (DTM) [1] and embedding-based models.

DTM will be especially useful for document sets where documents are from different time periods, since the document-topic distribution of a document using a DTM will be contextualized for the specific time period. Analysis of historical newspapers using DTM is discussed in Section 4. In Section 3.8, we used an LDA-like embedding-based topic model, specifically the Embedded Topic Model (ETM) [2], for extracting comments from user-generated comments.

3.8. Use case: Topic analysis of comments in a news forum

As a use case of using topic models to compare between document sets, we present work we did with external collaborators on a system that incorporates text and topic information in a classification task. We developed a topic-aware neural network classifier to classify user comments from an online news forum as either **Blocked** (comment violates one or more of the news platform's comment moderation rules) or **Non-blocked** (does not violate any rules). Our results showed that adding topic information not only improves performance, it also resulted in a more confident model. This work has been published and presented at the International Conference on Recent Advances in Natural Language Processing (RANLP 2021) on 1-3 September 2021 and published in its proceedings [3].

3.8.1. Dataset: 24sata user comments

In our experiments, we used the comments dataset from *24sata*, Croatia's most widely read newspaper[2]. The moderator labels included not only a label for blocked comments, but a record of the reason for the decision according to a 9-class moderation policy. We also linked comments to their respective articles and use the news section the article is filed under as the section that a comment belonged to. Although the full dataset included comments from 2016 to 2019, we only used the 2018 data for training and the 2019 data for testing. Table 2 shows the statistics of the training and validation sets and Table 3 shows the details of the test set.

Comment Moderation Data			
	Blocked	Non-blocked	Blocking Rate
Train	4,984	75,016	6.23%
Valid	642	9,358	6.42%
Test	37,271	438,142	7.84%
Topic Modelling Data			
	Blocked	Non-blocked	Blocking Rate
Train	34,863	36,725	48.70%
Valid	4,880	5,120	48.80%

Table 2: Details of datasets used experiments.

Section (– Subsection)	**Blocked**	**Non-blocked**	**Blocking Rate**
Kolumne (*Columns*)	655	6382	9.31%
Lifestyle	2,426	30,985	7.26%
Show	6,827	58,896	10.39%
Sport	5,882	80,820	6.78%
Tech	382	7,173	5.06%
Vijesti (*News*)	20094	239835	7.73%
– Crna kronika (*Crime*)	5,917	62,471	8.65%
– Hrvatska (*Croatia*)	3,527	45,170	7.70%
– Politika (*Politics*)	6,088	80,264	7.05%
– Svijet (*World*)	2,625	31,459	7.24%

Table 3: Details per section, and (for section Vijesti) subsection, of the comment moderation test set.

3.8.2. Proposed topic-aware models

Text encoder. We came up with several model architectures that combine a language model with topic features extracted from a topic model. For the comment text representation, we use a bidirectional LSTM (BiLSTM) [4]. For a given comment, the text is passed through an embedding layer then a BiLSTM where the output of the final hidden state is taken as the encoded representation of the comment.

Topic model. For the topic features, we used topics from a trained **Embedded Topic Model (ETM)** [2]. In the ETM, the topic-term distribution for topic k, β_k, is induced by a matrix of word embeddings ρ and the topic embedding α_k which is a point in the embedding space:

$$\beta_k = softmax(\rho^T \alpha_k) \tag{1}$$

The topic embeddings, α, are learned during topic inference while the word embeddings ρ can be pretrained or also learned during topic inference. In this work, we use pretrained embeddings.

The document-topic distribution of a document d, θ_d, is drawn from the logistic normal distribution (LN) whose mean and variance come from an inference network:

$$\theta_d \sim LN(\mu_d, \sigma_d) \tag{2}$$

[2] http://24sata.hr/

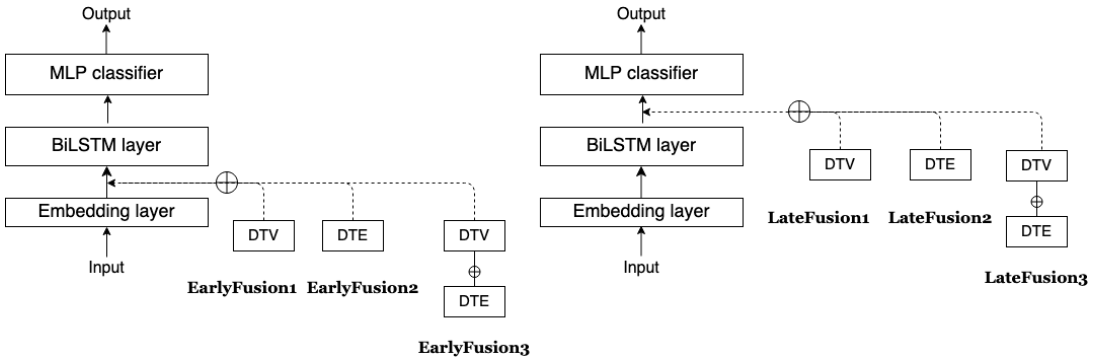


Figure 2: Proposed model architectures combining text and topic features.

Given a trained ETM, we inferred the θ_d of an unseen document d which we take as our **document-topic vector (DTV)**. Then we computed the **document-topic embedding (DTE)** as the weighted sum of the embeddings of the topics in doc d, where the weight corresponds to the probability of the topic in that document:

$$DTE = \sum_{k=0}^{K} \alpha_k \theta_{d,k} \tag{3}$$

where α_k is the topic embedding of topic k, and $\theta_{d,k}$ is the probability of topic k in doc d.

Fusion methods. We proposed two fusion mechanisms to combine the comment text and topic representation: *early* and *late* fusion. In early fusion, topic features are concatenated with the comment word embeddings and then passed to the BiLSTM. In **EarlyFusion1 (EF1)**, only DTV is concatenated with the word embeddings; **EarlyFusion2 (EF2)** uses DTE instead of DTV; and **EarlyFusion3 (EF3)** uses both DTE and DTV. In late fusion, topic features are concatenated with the output representations of the BiLSTM, and passed to the MLP for classification. Again, **LateFusion1 (LF1)** uses DTV; **LateFusion2 (LF2)** uses DTE; and **LateFusion3 (LF3)** uses both. Figure 2 shows the architectures of our proposed models.

3.8.3. Experimental setup

Baseline models As baselines, we used the following models trained only on text *or* topics:

- **Text only**: BiLSTM model with the comment text alone as input. The embedding layer is initialized with pretrained embeddings.
- **Document-topic vectors (DTV)**: MLP classifier with document-topic vectors as input.
- **Document-topic embedding (DTE)**: MLP classifier with document-topic embeddings.
- **DTV+E**: MLP classifier with concatenated document-topic vectors and embeddings.

Hyperparameters We used 300D word2vec embeddings, pre-trained on the Croatian Web Corpus [5], for training the ETM and to initialize the embedding layer of the BiLSTM. The ETM is trained for 500 epochs for 100 topics with default hyperparameters from the original implementation. The BiLSTM is composed of one hidden layer of size 128 with dropout set to 0.5. We limit the comment length to the first 200 words. The MLP classifier is composed of one fully-connected layer, one hidden layer of size

Section – Subsection	Text only	Topics only			Text+Topic Combinations					
		DTV	DTE	DTV+E	EF1	EF2	EF3	LF1	LF2	LF3
All	62.97	62.20	59.3	58.33	66.33	66.58	65.61	**67.37**	66.22	66.95
Kolumne	59.86	59.65	56.25	55.33	62.40	62.90	63.13	63.25	62.38	**63.6**
Lifestyle	69.21	70.07	65.93	64.47	72.73	70.9	69.36	72.00	72.39	**72.92**
Show	61.97	61.30	58.62	57.60	65.24	65.63	64.26	**66.50**	65.00	65.86
Sport	63.22	61.42	58.61	57.90	67.11	67.86	66.74	**68.26**	67.14	67.82
Tech	64.87	66.37	63.17	62.55	67.72	68.74	67.65	68.76	67.68	**69.15**
Vijesti (*News*)	62.38	61.49	58.79	57.77	65.58	65.99	65.24	**66.77**	65.53	66.24
– Crna kronika	64.67	63.98	61.03	59.84	68.10	68.88	68.11	**69.60**	67.89	68.88
– Hrvatska	63.61	63.50	60.10	58.93	67.24	66.86	65.95	67.90	67.12	**67.95**
– Politika	57.93	56.49	54.95	54.20	60.51	61.52	60.84	**61.61**	60.63	61.30
– Svijet	63.58	62.55	59.62	58.35	66.83	66.95	66.33	68.44	67.21	67.57

Table 4: Classifier performance measured as macro-F1.

64, a ReLU activation, and a sigmoid for classification with the classification threshold set to 0.5. We train all models for 20 epochs with early stopping based on the loss in the validation set.

3.8.4. Results

In Table 4, we present the performance of the baseline and proposed models, measured as macro F1-score. All models combining text and topics perform better than the models that used only text or topic information. Surprisingly, the DTV model performed comparatively better than the DTE and DTV+E models, and performed almost as well as the text-only model; however, we show in Section 3.8.6 below that DTV is much less confident in its predictions than the text-only model. Overall, the best performing model was LF1, which improved the text-only model's performance by $+4.4\%$ (67.37% vs 62.97%); and improved by a similar amount over [6]'s results using mBERT (macro-F1 score 62.07 for year 2019).

Interestingly, we see wide variation in performance across news sections. We observed that Lifestyle and Tech are the easiest sections (best F1 over 0.72) while Politika (*Politics*) was the most difficult (best F1 below 0.62). The main cause appeared to be that Lifestyle and Tech have the highest proportion of spam comments: on average, 49.44% of blocked comments in the test set are spam, but for Lifestyle and Tech this number rose to 77.25% and 69.63%, respectively. As for the Politics section, we hypothesised that, excluding spam, the topics discussed in blocked and non-blocked comments have high overlap (Section 3.8.5).

3.8.5. Topic analysis

We also analysed how the topic distributions differ between blocked and non-blocked comments and across the sections. Our aim was to understand what subjects are discussed in these two comment classes and across the different sections, to gain insight into what characterises a blocked comment and a non-blocked one, and whether this varies between different sections.

We took the top topics of a document set by taking the mean of the topic distributions of all the docu-

Croatian football	dinamo, hajduk, zagreb, zagrebu, placu, europi, zagreba *(dynamo, hajduk, zagreb, zagreb, market, europe, zagreb)*
State and government	država, države, državi, vlasti, državu, vlade, vlada *(state, states, state, authorities, state, governments, government)*
Moderately offensive	gluposti, sramota, sram, glup, jadni, jadan, jadno, budale *(nonsense, shame, disgrace, stupid, miserable, miserable, miserable, fools)*
Death and illness	žena, žene, ljudi, osoba, osobe, ženu, smrt, čovjeka *(woman, women, people, person, persons, woman, death, human)*
Civil war	srbi, hrvata, tito, srba, srbije, srbiji, srbima, srbija *(serbs, croats, tito, serbs, serbia, serbia, serbs, serbia)*

Table 5: Selected topics with English translations. The first two topics are prevalent in non-blocked comments, the next two are prevalent in blocked comments, and the last is prevalent in both classes.

ments in that set and ranking the topics according to their probability in this mean distribution. In this analysis, the document sets were the blocked and non-blocked comments. We took the top 15 topics for analysis because this is the average number of topics used by the comments (by this we mean the number of topics in a comment with a probability greater than zero).

For the entire test data, the top topics of non-blocked comments covered a diverse range of subjects from politics to football to scientific research (Figure 3). The top topics in blocked comments were dominated by spam and insults. Table 5 shows some of these topics (labels are manually assigned by native speaker). In Figure 3 we also see many topics shared between blocked and non-blocked comments.

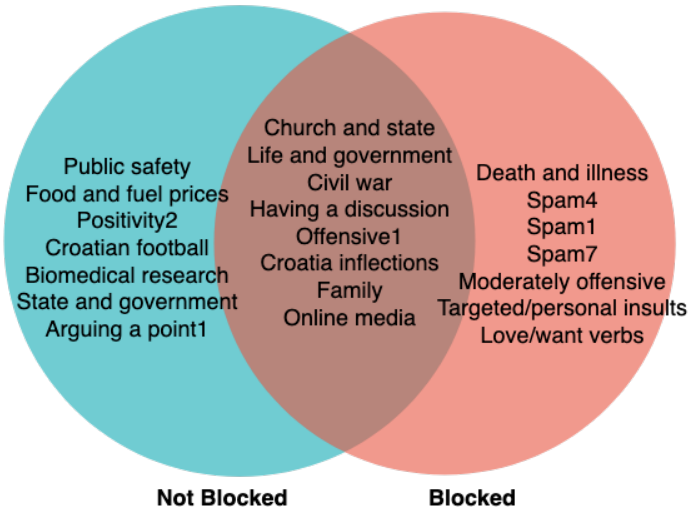


Figure 3: Top topics of the blocked and non-blocked comments for the entire test set.

We illustrated how different topics intersect between blocked and non-blocked comments across and between sections by looking at the top topics of the easiest and most difficult sections, Lifestyle and Politics, respectively. Figure 4 shows the top topics of these sections and the intersections between them. In Politics, blocked comments tended toward spam and targeted insults. Non-blocked topics were about public safety, finances and scientific research. Moreover, there were many overlapping topics between blocked and non-blocked. This suggested that blocked and non-blocked comments in Politics

discuss the same subjects. This supported our hypothesis that one reason why comments in Politics are difficult to classify was that thematically, blocked (excluding spam) and non-blocked comments tended to be similar. In Lifestyle, blocked topics were dominated by spam and while there were topics on offensive words and insults, they were not as prevalent as the spam ones. The non-blocked topics were about family and relationships and commenters arguing with each other. In terms of topic overlaps between Lifestyle and Politics, blocked comments in both sections were about spam and targeted insults, while non-blocked comments used a more positive tone.

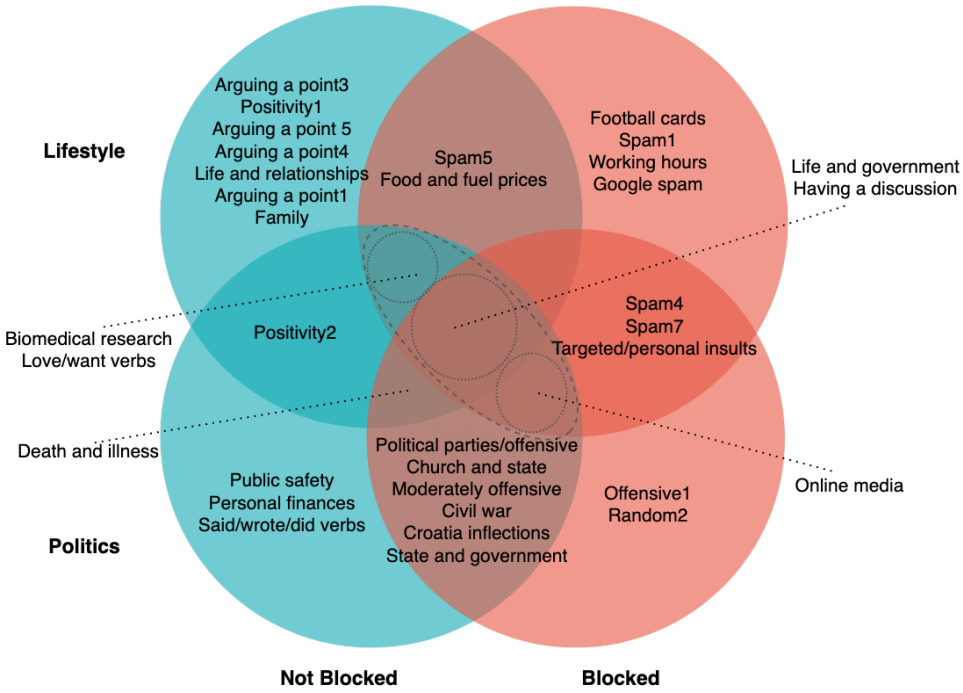


Figure 4: Top topics of the blocked and non-blocked comments of the Lifestyle and Politics sections.

3.8.6. Analysis of Classifier Outputs

In general, we observed that blocked comments tended to use similar topics across different sections, while non-blocked comments have more diverse topics. Blocked comments across sections had more in common with each other than non-blocked ones. Topics in non-blocked comments tended to be more relevant to their news section: for instance, family and relationships were not discussed a lot in the Politics section, while Lifestyle commenters did not tend to talk about the government and political parties.

To analyse confidence, we gradually increased the classification threshold from 0.5 to 1.0 in increments of 0.05. For every new threshold, we plotted the macro-F1 for the different models (Figure 5). We compared the confidence of four models: DTV, Text only, EF2 (the strongest early fusion model), and LF1 (the overall best-performing model). The most confident model was LF1 and the least confident was DTV. The two fusion classifiers displayed similar levels of confidence. The Text-only classifier was not as confident as the fusion classifiers but still more confident than DTV. This suggested that adding topic features to text not only improved performance, it also increased classifier confidence.

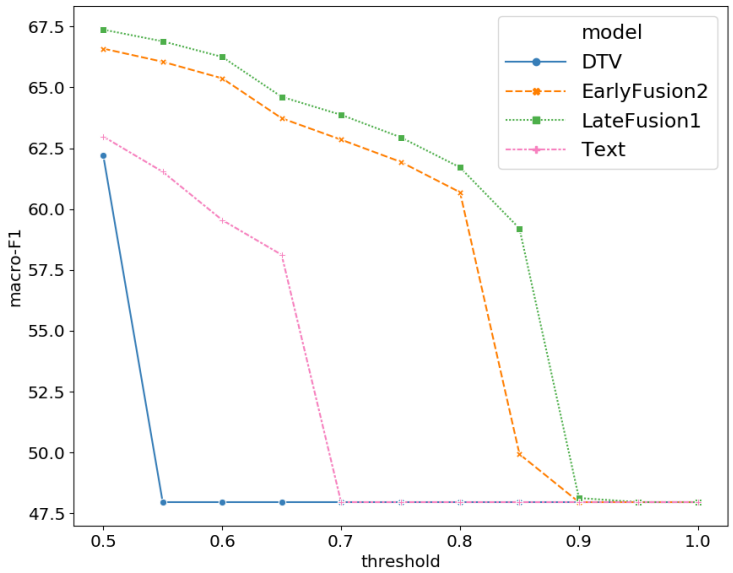


Figure 5: Confidence of the top performing models.

3.8.7. Summary of findings

In this section, we presented a use case of topic-based analysis of documents from different contexts (blocked/non-blocked or from different news sections). We proposed a model that incorporates topic and text information to classify comments in a news forum as blocked or non-blocked. Our analysis showed that blocked comments tend to be more homogeneous across news sections while non-blocked comments cover a more diverse range of topics and are more relevant to the section they appear in. Moreover, our results show that combining text and topic features improves the overall model performance and makes the model more confident in its predictions.

3.9. Visual topic modelling

The topic models presented so far use text data to learn topics. However, other modalities such as images are also useful. Multimodal models that map data from different modalities (text and image, for instance) in the same space are currently gaining popularity in machine learning [7, 8, 9]. Here we present the Visual Topic Model (VTM), a topic model that takes paired images and texts during training and learns topics from their embeddings. During testing/inference time, the model can either take in text *or* image and obtain a topic distribution. This work has been presented at the MediaEval 2021 workshop on the 14th of December 2021. The working notes paper is available at https://2021.multimediaeval.com/paper37.pdf.

3.9.1. Method

The Visual Topic Model (VTM) is an extension of the Contextualized Topic Model (CTM) [10]. CTM is a family of neural topic models that are trained to take as input text embeddings and to produce

as an output the bag-of-words reconstruction. The model trains an inference network to estimate the parameters of the topic distribution of the input. During inference time, this topic distribution is used as the model output to describe texts unseen during training.

Thus, to train a model, each input instance has two parts: text embeddings and bag-of-words representation (BoW). Our main contribution is that we replace text embeddings with visual embeddings and demonstrate that they can be used to train a topic model. The ZeroShot CTM model uses the BoW representation only to compute loss, i.e. this information is not needed during inference time. Since we have a training set that consists of aligned text and image pairs, we can use the texts to produce the BoW representation and use it to train a model.

To obtain image embeddings, we use CLIP, a pre-trained model that produces text and image embeddings in the same space [7]. CLIP representations for text and image are already aligned. However, this is not a requirement for VTM: in our preliminary experiments, we used ViT [11] for image and German BERT for texts[3]. The results obtained using non-aligned embeddings were only slightly worse than those with CLIP embeddings. Topic models converge to similar results because they use the same BoW to compute loss (the alignment of embeddings simplifies this process but is not necessary).

This basic procedure, i.e. training image and text models independently, produces similar but not aligned topic models. Topics could be slightly different, and even similar topics are organized in different (random) order. To increase the similarity between text and image models, we use *knowledge distillation*. In this approach, a student model uses a different input than a teacher (e.g. image instead of text) but should produce the same result.

CTM uses a sum of two losses: reconstruction loss and divergence loss. The reconstruction loss ensures that the reconstructed BoW representation is not far from the true one. The divergence loss, measured as KL-divergence between *priors* and *posteriors*, ensures a *diversity property*, that is desired for any topic model: a topic has large probabilities only for a small subset of words and a document has high probabilities only for a small subset of topics.

In the knowledge distillation approach, we leave the reconstruction loss intact but replace divergence loss with KL-divergence in regard to the *teacher output*. The assumption here is that since a teacher model is already trained to be diverse and a student model is trained to mimic the teacher, the student does not need priors. Experiments supported this assumption.

We use knowledge distillation in two versions: *joint model* and *text-teacher*. In the joint approach, we first train a joint model that takes as input a concatenation of text and image embeddings, then we train two student models for image and text separately. In the second approach, we first train a text model as the teacher and then an image model as a student. We try 60 and 120 topics with both joint and text-teacher approaches.

Model	Correct in Top100	MRR@100	Recall@5	Recall@10
baseline (CLIP)	1,225	0.169	0.22	0.30
joint 120 topics	767	0.043	0.06	0.09
joint 60 topics	698	0.030	0.04	0.07
text teacher 120 topics	816	0.042	0.05	0.09
text teacher 60 topics	757	0.037	0.05	0.08

Table 6: Visual topic modelling results on retrieving the correct image for a news article.

3.9.2. Results

The results are presented in Table 6. As a baseline, we use the cosine similarities between CLIP embeddings, without any domain adaptation for the text[4]. As can be seen from the table, the best results are obtained with CLIP embeddings, that are used without any fine-tuning to the training set. They are able to find the correct image in 1225 cases out of 1915 and has a Mean Reciprocal Rank (MRR) of 0.17. The best VTM model finds the correct image in 816 cases out of 1915 and yields an MRR of 0.03.

These results to some extent correspond to our previous observation that topic modelling is not the best method for document linking [12]. The probable explanation for that might be that topic modelling produces a sparse representation of the data. While CLIP embeddings are continuous vectors and could represent an almost infinite amount of information, in topic modelling, dimensions are not independent due to the diversity requirement, described above. It can be seen from Table 6 that models that have more topics yield better performance.

Another interesting observation is that models that use the text model as a teacher for a visual model work better than joint models. This is an unexpected result, since one would expect that a model that has access to full information could serve as a better teacher. It is possible that text bears less noise: a text model uses the same text for contextual and BoW representation, while an image could be completely random.

Though according to our results, CLIP embeddings outperform VTM, the ability to illustrate text topic might be a desirable property for some applications, as well as topic interpretability.

4. Analysis of discourses over time

In this section, we discuss methods for the comparative analysis of textual content over time. The NewsEye collection contains diachronic corpora spanning several decades. We want to show what trends or topics were discussed when, how these topics were discussed, and how the popularity of these topics changes from year to year. This kind of analysis is especially interesting for history, since, by nature, is concerned with the temporal dimension in the data and differences between epochs.

Work on diachronic corpus analysis is done on two levels: the lexical level, i.e. detection of word semantics and usage change, and the discourse level, i.e. investigating discourse dynamics that are not

[3] https://huggingface.co/bert-base-german-cased

[4] We use an implementation provided as a part of Sentence Bert library: https://www.sbert.net/examples/applications/image-search/README.html

necessary reflected in language change but nevertheless could be discovered in large text collections.

On the lexical level, we investigated applicability of a recent contextualized language model (BERT) for detection of word meaning change over time. We run experiments on several standard datasets; our method demonstrated performance comparable with the state of the art on this task. In addition, we proposed a novel method to scale up word meaning change detection that may potentially lead to new applications in digital humanities.

On the discourse level, we investigated applicability of topic models to capture changes of thematic trends in large diachronic collections. We applied two topic models—DTM and LDA—to the Finnish 19th century newspapers, i.e. a Finnish part of the NewsEye dataset. We proposed a novel sampling and data aggregation procedures to perform analysis on large massive of the data. This work was done in close collaboration with historians, which allows us to find advantages and disadvantages of the two models and make several observations on their applicability for historical use cases.

In addition, we proposed a novel deep learning method to find discourse change in diachronic data. Since this method is supervised and historical data do not have an annotation needed to train the model, we performed these experiments on synthetic datasets generated from modern Finnish news. However, we demonstrated that the model could find meaningful results on real data, thus the method is potentially applied to historical data as well.

4.1. Word meaning change detection

Semantic change detection is the task of detecting and analysing word evolution in textual data. Each word has a variety of senses and connotations, constantly evolving through usage in social interactions and changes in cultural and social practices. Identifying and understanding these changes allows detection of cultural and linguistic trends and possibly predict future changes.

A large majority of modern methods for semantic shift detection leverage word embeddings. The detailed overview of the field could be found in recent surveys [13, 14, 15]. Few recent studies that employed contextualised embeddings, e.g. [16, 17].

The most usual formulation for this problem is the following:

- the dataset consists of *target words* and a *corpus*, which contains texts from at least two time periods;
- the task is to compute a measure of semantic shift for each target word relying on the corpus data;
- the words are then ranked by the strength of the change and performance is evaluated as Spearman rank correlation between obtained ranking and a manually annotated ground truth.

4.1.1. Method

We use BERT, pre-trained or fine-tuned on the task-specific corpus. To generate target-word embeddings a model is fed with sentences containing a target words. A sentence embedding is generated for each of the input sentences by summing the last four encoder output layers of BERT.

Method	Spearman	
	Pretrained BERT	Fine-tuned for 5 epochs
averaging	0.349	0.341
k-means, k = 3	0.444	0.392
k-means, k = 5	0.443	0.508
k-means, k = 7	0.434	0.491
k-means, k = 10	0.443	0.466
affinity propagation	**0.486**	**0.510**

Table 7: Correlations between detected semantic change and manually annotated list of semantic drifts.

We employ two methods to measure semantic shift using contextual embeddings: averaging and clustering. **Averaging** is a simple aggregation approach where all target-word usage representations from a given time period are averaged. Then the cosine distance between two averaged time-specific representations of the word, to measure semantic shift.

Clustering of word usage representations results in sets of word usages, where each set is expected to correspond to a single word sense or a specific context. From the output of the clustering algorithms, we create two time-specific cluster distributions by normalizing the cluster counts within each period. Then the Jensen-Shannon divergence (JSD) between two time period-specific distributions is used to measure the semantic change. We used two clustering techniques, namely *affinity propagation* and *k-means*

4.1.2. Experiments

In the first experiment evaluated our method on a human-annotated dataset [18] consisting of 100 words from various frequency ranges, labelled by five annotators according to the level of semantic change between the 1960s and the 1990s. The most significantly changed words from the dataset are, for example, *user* and *domain*; words for which the meaning remain intact, are, for example *justice* and *chemistry*. To fine-tune BERT, we use the Corpus of Historical American English (COHA) [5]. We focus our experiments on the most recent data in this corpus, from the 1960s to the 1990s (1960s has around 2.8 million and 1990s 3.3 million words), to match the manually annotated data.

Results of this work is presented in Table 7. The proposed method, affinity propagation on the fine-tuned BERT model, yields the highest Spearman rank correlation. Results obtained using pretrained and fine-tuned models are consistent: in both runs, averaging yields lower performance than clustering and affinity propagation is the best clustering method. This work has been presented at the Temporal Web Analytics Workshop at the Web Conference 2020 and published in the workshop's post-proceedings [19].

The following set of experiments were conducted under within the framework of the SemEval-2020 Task 1—Unsupervised Lexical Semantic Change Detection [20]. The task deals with detection of semantic change in temporal corpora containing texts in four languages: English, German, Latin and Swedish. The challenge defines two subtasks: Subtask 1 is a binary classification, i.e., to determine whether a word has changed or not; SubTask 2 aims at ranking a set of target words according to their rate of

[5] https://www.english-corpora.org/coha/

semantic change. In this section, we present the results obtained on the second subtask, were our team qualified as 5th also proved the best for the Latin corpus.

The basic method is based on BERT-embeddings clustering, as in the previous section. We also try several heuristics to filter out clusters that potentially contain noise and can distort comparison between time periods: we removed clusters containing only one or two instances; we filtered out sentences where a target word is used as a proper noun; we removed clusters if number of proper names were 5 times bigger than number of sentences.

In addition to context-depended embeddings, we generate 300-dimensional Word2Vec for each time slice and aligned two embedding space as in [21]. The cosine distance between representations of the same word from two time slices is used to estimate the semantic change.

The best result, as shown in Table 8, was obtained by an ensemble of a method using word2vec static embeddings and clustering of fine-tuned BERT contextual embeddings further improved with NE filtering. The clustering-based methods are outperformed by embeddings averaging and word2vec-based method, especially for Swedish corpus where the basic method produced results close to random. The variety among languages is significant, and the results averaged on all four corpora can be misleading. This work has been presented at the Fourteenth Workshop on Semantic Evaluation (SemEval 2020) and published in the workshop's proceedings [22].

4.1.3. Scalability and Interpretability

The main limitation of the cluster-based methods is the scalability in terms of memory consumption and time: clustering is applied to each word in the corpus separately, and all occurrences of a word need to be aggregated into clusters. For large corpora with large vocabularies, where some words can appear millions of times, the use of these methods is severely limited. This is the main reason why most of the research in this area has dealt with pre-defined lists of few hundred words.

For historical research, a data-driven approach is more desirable. In an ideal case, a researcher would like to see a list of the most changed words in the corpus without specifying any *a priori* knowledge. Methods based on static embeddings are more feasible for large-scale processing, but they are less interpretable since they look at the neighborhood of a word in each time period to interpret the change and ignore the fact that a word can have more than one meaning.

Thus, we propose a scalable method for clustering of contextual embeddings that generates interpretable representations and outperforms other cluster-based methods. The main idea is that we do not need to use all mentions of a word in the corpus to form its contextualized representation. In this word, we limit embeddings store for each word in each time slice with only 200 most distinct instances. For all other word mentions, we find the most similar vector among those 200, and then average the two vectors. Averaging is done while reading the corpus, which allowed for storing 200 vectors for more than 7 thousand words. Then clustering is applied on top of those 200 vectors. We also proposed using Wasserstein distance to measure the difference between cluster distributions across periods, which allows us to improve previous results.

The combination of scalable clustering with the interpretation pipeline opens new opportunities for di-

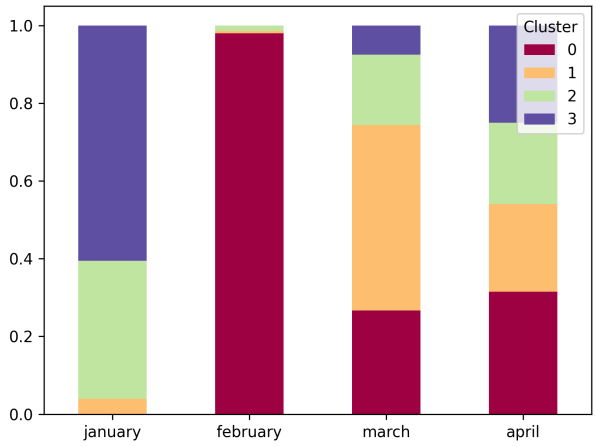
	Input	Method	Post-Processing	AVG	English	German	Latin	Swedish
pretrained BERT	aff-prop, JSD	-		0.278	0.216	0.488	0.481	-0.072
fine-tuned BERT	aff-prop, JSD	-		0.298	0.313	0.436	0.467	-0.026
fine-tuned BERT	aff-prop, JSD	NE, small clusters		0.291	**0.413**	0.310	0.472	-0.029
fine-tune BERT	averaging, cosine dist	-		0.397	0.315	0.565	**0.496**	0.212
word2vec aligned	cosine dist	-		0.394	0.341	**0.691**	0.131	**0.413**
Ensemble aff-prop + w2v	distance multiplication	NE, small clusters		**0.442**	0.361	0.603	0.460	0.343

Table 8: SemEval Task 1-2 results: Spearman correlation with ground truth.

achronic corpus exploration. We demonstrate how it could be used to analyze the Aylien Coronavirus News Dataset[6]. The corpus contains about 500k news articles related to COVID-19 from January to April 2020, unevenly distributed over the months (160M words in March, 41M in February, 35M in April and 10M in January). We split the corpus into monthly chunks and apply our scalable usage change detection method for *all* words that appear more than 50 times within each period and determine the most changing words.

Among top-10 most changing words the word *diamond* is related to the cruise ship "Diamond Princess", which suffered from an outbreak of COVID-19 and was quarantined for several weeks. The word *king*, which is the second most changing word, is related to the King County, Washington, where the first confirmed COVID-19 related death in the USA appeared, and to the Netflix show "Tiger King", which was released in March. Thus, the primary context for this word changed several times, which is reflected in our results.

The interpretability of our method is illustrated in Figure 6, where we present clusters obtained for word *diamond*, the most changing word according to our results. The left part shows cluster proportion in each time slice; the right part shows the most prominent keywords for each cluster. This work has been presented at the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2021) and published in its proceedings [23].



#	Keywords
0	diamond princess, cruise ship, princess cruise, japanese, tested positive, confirm, ship diamond
1	neil diamond, comic, sweet caroline, trump, song, diamond said, comic book,
2	diamond hill, hill capital, diamond jubilee, diamond mountain, league postponed, portfolio, athletics
3	diamond industry, black diamond, jewellery, hong kong, diamond ring, surat diamond, india

Figure 6: Cluster distributions per month and top keywords for each cluster for word *diamond*.

[6] https://aylien.com/blog/free-coronavirus-news-dataset

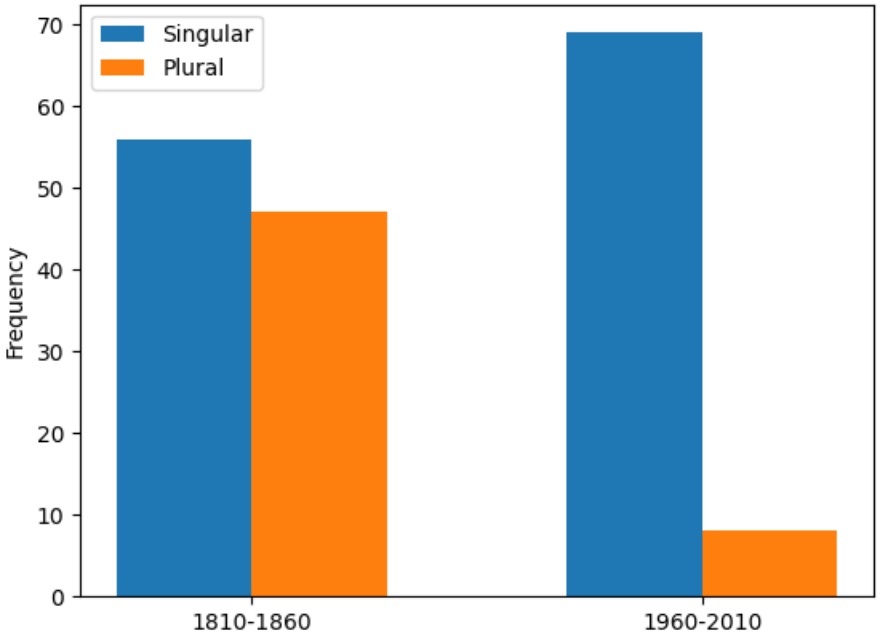


Figure 7: Changes in the number category distribution for the English noun *lass* over time, calculated on the English corpora of the SemEval 2020 shared task 1 [20]

.

4.1.4. Morpho-syntactic approach to semantic change

Though contextualized language models provide a powerful instrument to encode semantics, they do not provide explicit information on syntactic and morphological properties of words. However, it has been long known in linguistics that semantics, morphology and syntax are strongly interrelated [24, 25]. There is an evidence from cognitive linguistics that "mental concordance" stores interrelated information on semantic and syntactic preferences of words [26], rather than syntax and lexicon as separated systems.

Thus, even within a project aimed at development of embedded semantic representations, it is crucial to pay attention to other language dimensions, presented in alternative forms. This work has been presented at the 25th Conference on Computational Natural Language Learning (CoNLL 2021) where we applied *grammatical profiling* [27, 28] for semantic change detection [29].

In this work, we rely on observation that semantic change is often accompanied by changes in morpho-syntactic preferences of the words. For example, the English noun *lass* originally meant YOUNG WOMAN but in the 20th century its new SWEETHEART meaning became more dominant. This was accompanied by a sharp decrease in plural usages (*lasses*), as shown in Figure 7.

Grammatical profiling yields surprisingly good evaluation scores across different languages and datasets, without any language-specific tuning. For Latin, a language with rich morphology, our methods even establish a new SOTA in Subtask 2 of SemEval'20 Task 1. Nevertheless, our results indicate that in general grammatical profiling cannot compete with state-of-the-art methods based on large pre-trained language models, since they have the potential to encode both semantics and grammar. Yet reaching the highest possible scores on the task was not our goal. Instead, the aim of our study was to

demonstrate that more attention should be paid to the relation between grammar and semantic change.

4.2. Discourse Change Detection with Topic Modelling

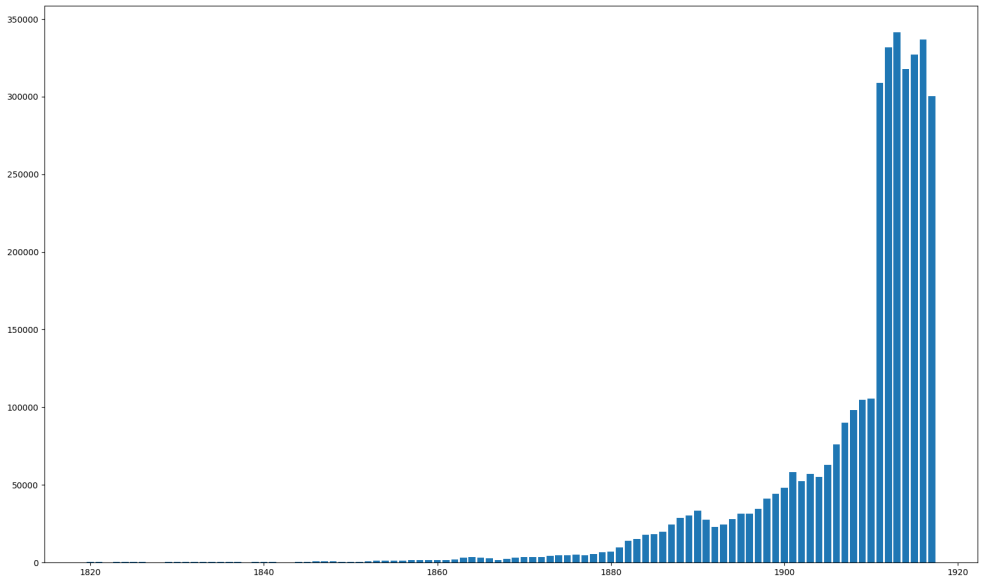


Figure 8: Number of articles for each year in the Finnish portion NLF collection starting in 1820.

4.2.1. Topic evolution using dynamic topic models

To study how the words related to a topic changes over time, we train a dynamic topic model [1] (DTM) on 64 years of the NLF collection (1854-1917) with 50 topics. The Finnish portion of this corpus covers more than a hundred years (1790-1917). OCR quality and number of articles vary considerably from year to year (see Figure 8).

We randomly sub-sample the data such that we have 100 randomly selected articles for each year because DTM is difficult to scale as the amount of data and number of time slices grows. Moreover, we want to have a dataset that represents all time slices equally.

To give an idea of the kinds of topics inferred for this collection, Figure 9 shows the top words (high probability words) of each topic averaged over all time slices. Since this is a dynamic topic model, the topic changes slightly from year to year, therefore, we take the mean of the topic-term distributions for the same topic for all time slices and use this mean distribution to show the *mean top words* of a topic.

Take, as an example, a topic about the Finnish legislative assembly. This topic is dominated by words related to the Russian Empire in the 1850s and 1860s, and gradually words related to the senate and parliament become more prominent towards the end of the 19th century. Figure 10 shows the changing prominence of words related to this topic. We see that words such as *keisari* (emperor) and *keisarillinen* (imperial) are prominent in the 1860s, while words related to Finnish institutions such as

Figure 9: Topics found by DTM trained on the NLF corpus. The highest probability words are shown for each topic, from the topic-term distribution averaged over all time slices.

senaatti (senate) and *sääty* (estate) become prominent in the 1890s and beyond. Some words such as *Suomi* (Finland) are prominent throughout the topic, as is to be expected.

Additionally, we also show how much each topic was used in the corpus during each year. In this case, we use the whole NLF (National Library of Finland) collection for the 64 years we are interested in *without sampling*. To do this, we take the topic-term distributions, learned as above; for each year and for each article in that year, we infer the document-topic distribution by iteratively sampling the topic assignment of each word in the document with a Gibbs sampler [30]. After doing this inference for all articles, we can compute how much each topic was used in a specific year, and this gives us an idea of a topic's popularity and how that popularity changes from year to year. We normalize topic usage for each year such that they are comparable to each other. Figure 11 shows the normalized topic usage for 64 years of the NLF collection.

In collaboration with the DH researchers from Helsinki (UH-DH), we applied dynamic topic models to investigate discourses from Finnish newspapers that have declined over the years due to a variety of

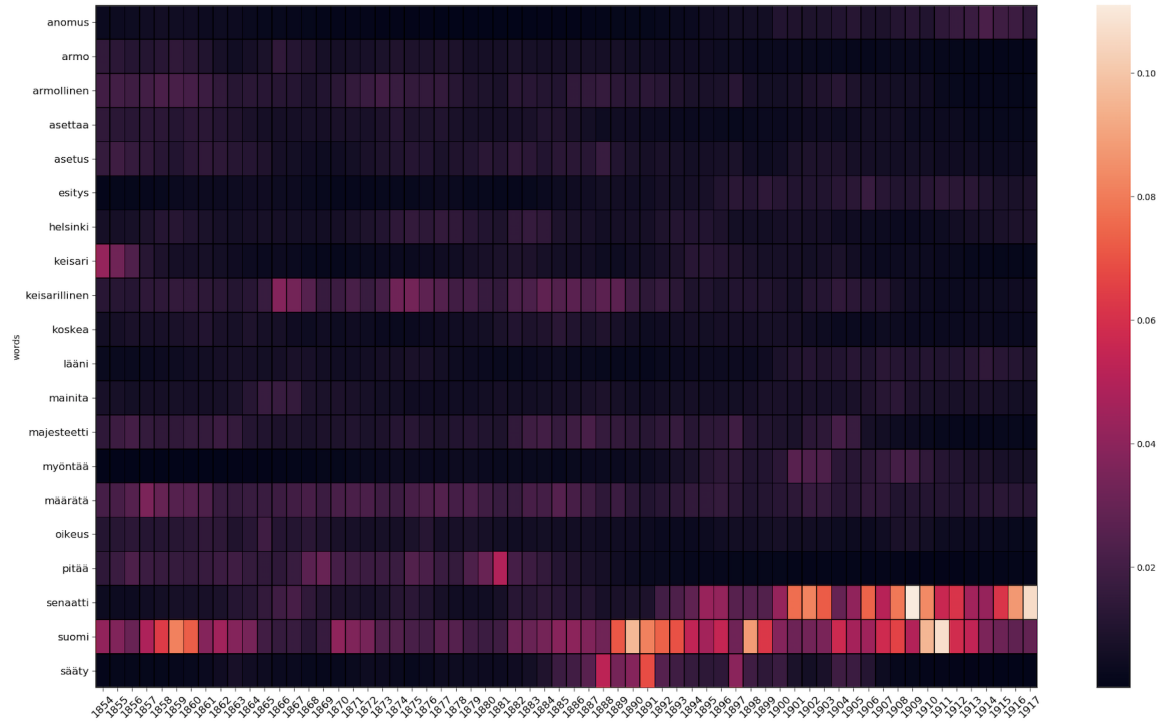


Figure 10: Heatmap showing the changing prominence of some top words for a topic on the Finnish legislative assembly

societal and historical factors. This work has been presented at the conference on Digital Humanities in the Nordic Countries 2020 and published in its post-proceedings [31].

4.2.2. Other Methods to Track Discourse Dynamics

In digital humanities research in general and in particularly in computational history, research questions are generally very complex and involve a lot of uncertainty, thus ground truth needed for numerical evaluation is usually unavailable. Moreover, quite often a historical study deals with a specific use case, which means that the data is a single non-annotated dataset without proper split into training and test subsets. To overcome this difficulty, we propose an evaluation on multiple *synthetic* datasets. The idea is to exploit manually assigned categories, that are labelling articles in many news collections. Distinct periods and spikes in the data could be mimicked by sampling from a single label according to a certain pattern, while all other categories are sampled randomly. Then the task is to implement a model 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. Synthetic data is widely used in a lexical semantic change detection [32, 33], but we are unaware of any similar work performed at the discourse level and exploiting news categories for similar purpose.

We run our experiments on modern Finnish news datasets. We build multiple synthetic datasets for YLE news archive [7]. These datasets are used to evaluate trend detection methods.

[7] Freely available from Finnish language bank: http://urn.fi/urn:nbn:fi:lb-2017070501

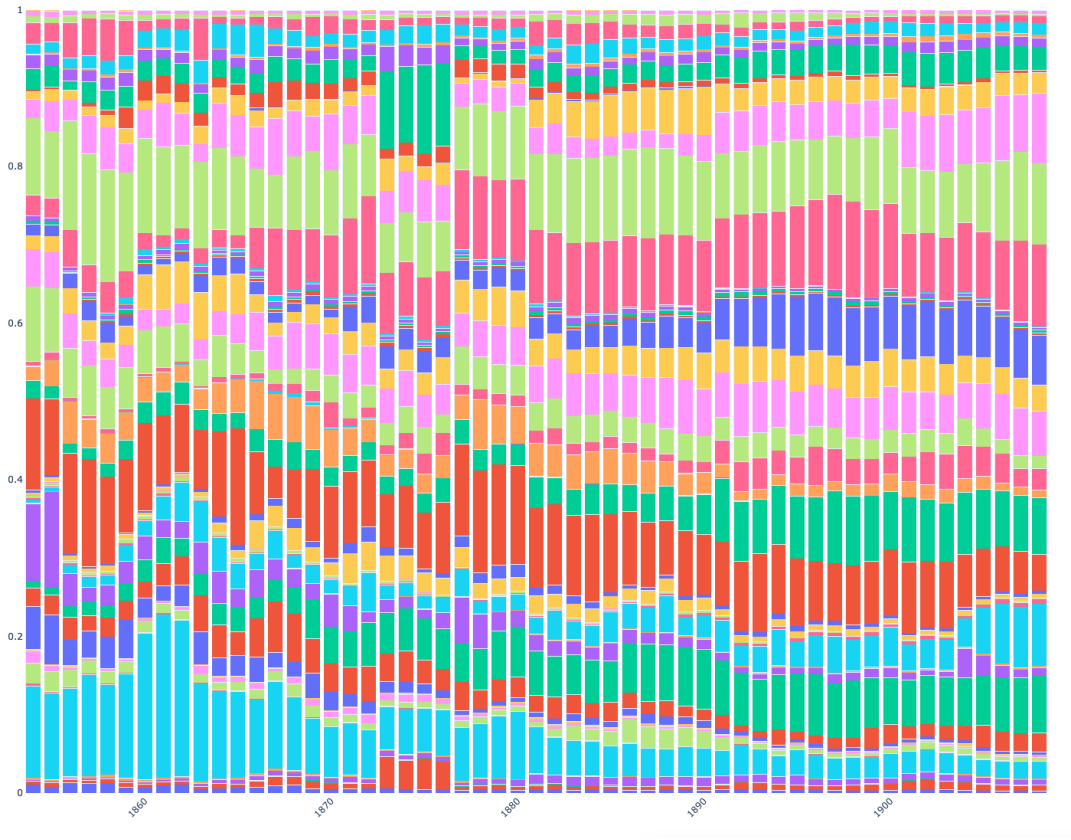


Figure 11: Normalized topic share during each year of the NLF dataset.

The basic approach that we use consist of two steps: first breaking the news collection into smaller datasets and then classifying these datasets as either stable or non-stable. For the first step we use either k-means or LDA, for the second step sequence-to-sequence neural networks and few simpler baseline methods. Our experiments show that synthetic datasets allow us to rank methods as either more or less suitable for the task. Application to the modern STT (Finnish News Agency) dataset allows us to find some interesting phenomena, though recall is still a problem for our method. Our best performing model, which is illustrated in Figure 12 is a combination of bidirectional LSTM and CNN models.

For a qualitative assessment, we use another Finnish corpus: The Finnish News Agency (STT) Archive[8]. The corpus consists of the STT newswire articles for the period between 1992-2018. We limit our experiments to the data from years 2007-2008, which does not overlap in time with YLE dataset.

Our experiments demonstrate that a model trained on synthetic datasets is able to extract meaningful results from the real data. One example is shown in Figure 13. The cluster used to form a timeline in the plot is associated with party politics. The date of the Finish parliamentary elections is shown with green vertical line. This date is positioned in two automatically determined pivot points, as it seems natural that elections are actively discussed in news some time before and after the event.

This work has been presented at HistoInformatics 2021, the 6th International Workshop on Computational History, and published in its proceedings [34].

[8] freely available for research use via Kielipankki: http://urn.fi/urn:nbn:fi:lb-2019041501

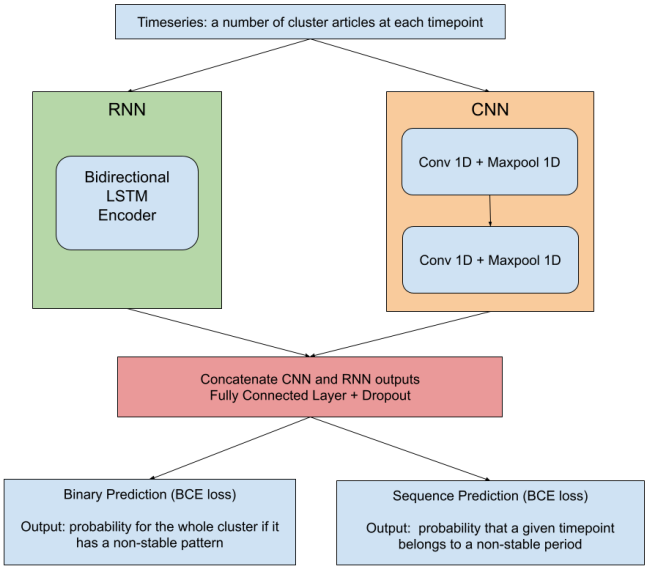


Figure 12: A best-performing model for the task of automatic finding non-stable periods in text collection. The input is a frequency timeline, built separately for each cluster obtained after the first (clustering) step. The outputs are: a binary decision on whether a timeline contains a period of non-stability and a sequence, which indicates the position of the non-stable period.

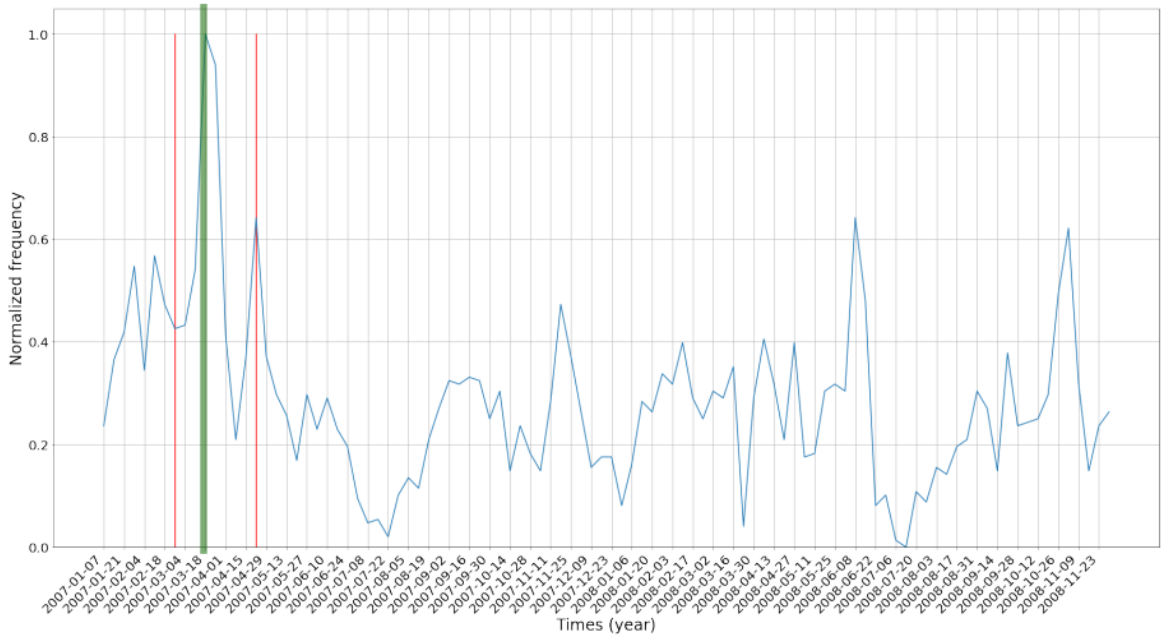


Figure 13: A non-stable cluster obtained on the STT data. Automatically detected pivot points show with red vertical lines. The documents within this cluster are mostly about politics and parties. The date of the Finnish Parliamentary elections is shown with green vertical line.

An alternative model for detecting discourse changes has also been developed as part of a Master's Thesis work by Mikko Lipsanen, a student in the Data Science Master's programme at the University of Helsinki. The model, shown in Figure 14, uses supervised contrastive loss [35] to train a deep neural

network to differentiate the representations of synthetic datasets with different discourse patterns from each other. After the contrastive pretraining, the model is used for downstream classification tasks both directly, by adding a classifier on top of the trained model, and indirectly, by performing unsupervised classification and change point detection to the data that has passed through the trained model.

The experiments with the model indicate that contrastive pretraining has potential to be used as part of a discourse change detection pipeline. The pretraining performed on discourse patterns generalizes in a way that allows the model to be used also for classifying whether individual time points belong to an unstable discourse pattern. However, further research is still needed to improve the model architecture and training so that it could encode relevant information at the document level. In addition, the possibility to implement contrastive pretraining for discourse change detection also in an unsupervised setting would extend the applicability of the model in different contexts.

The work on the thesis is still ongoing, and therefore the text is not yet available online.

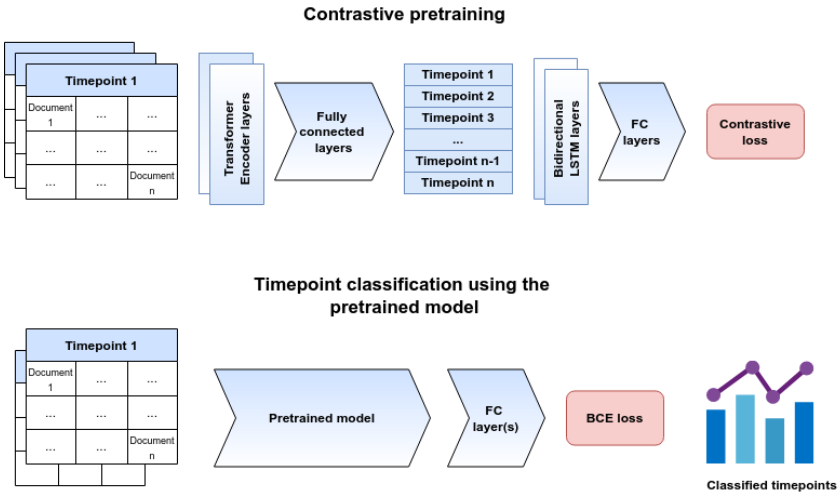


Figure 14: A deep neural network model developed for discourse change detection. The model trained with a contrastive loss function is used for a downstream classification task with only small changes in the architecture.

5. Implementation in the Demonstrator

Though most of the work presented in this deliverable was exploratory and the models that we implemented were fine-tuned for concrete use cases, some results were directly utilized in the NewsEye Demonstrator via the Personal Research Assistant.

Specifically, the methods for extracting top topics from datasets (Section 3.3) and extracting shared and distinct topics between datasets (Sections 3.5 and 3.4) are available for users via the Personal Research Assistant of the NewsEye platform. In addition, the Personal Research Assistant provides users with simple tools for building time series: if a dataset is big enough and consists of documents from different years, the tool splits the dataset in two, automatically finding the split point. The autonomous Investigator (Deliverable 5.6) uses at least one of the comparative analysis tool in every experiment, unless data are too small for meaningful analysis.

6. Use by Digital Humanities collaborators

We worked with the University of Helsinki DH group (UH-DH) on a paper on exploring discourse dynamics in nineteenth-century Finnish newspapers. Our work examines discourses and discussions that were popular in the past but have since disappeared due to a variety of factors. This work was presented at the Digital Humanities in the Nordic Countries Conference (DHN 2020) and published in its post-proceedings [31].

Another line of collaboration involved detection of usage change in Finnish words with suffix *ism*. These words refer to complex notions and help us to navigate complex social issues by using a simple one-word label for them. They are often associated with political ideologies, but on the other hand they are present in many other domains of language, especially culture, science, and religion. Historically, this has not always been the case. We studied "isms" in the NLF collection documents published from 1820 to 1917 in Finland. We used diachronic word embeddings and clustering to trace how new "isms" entered the lexicon and how they relate to one another over time. We were able to show how they became more common and entered more and more domains. Still, the uses of "isms" as traditions for political action and thinking stand out in our analysis. This work resulted in a workshop paper [36] and a journal publication in the Journal of Data Mining and Digital Humanities [37].

7. Conclusion

In this deliverable, we report on the work we have done for T4.2 in the following areas.

Comparing document sets. We developed methods to summarize document sets and compare individual documents and documents sets with each other using topic models. We demonstrated how we can use the topic distributions in document sets to extract topics that distinguish one set from another. We also present work where we used the methods we developed to analyze user-generated comments from an online news forum.

Discourse analysis over time Since the NewsEye collection spans decades, we want to investigate how the temporal aspect of the documents affects topics. We discussed the topic models we used to investigate how the words related to a topic change over time, and show methods to estimate the prominence of a topic over time. We also discussed methods we developed for quantifying changes in word meaning over time.

Collaboration. We worked with digital humanities scholars in the NewsEye consortium and applied our methods to their research questions. We presented a paper at a DH conference on using different kinds of topic models to explore the changing prominence of discussions in newspapers over time. In addition, we also collaborated on work that explored changes in ideological terminologies over time.

References

[1] David M Blei and John D Lafferty. “Dynamic topic models”. In: *Proceedings of the 23rd international conference on Machine learning*. ACM. 2006, pp. 113–120.

[2] Adji B Dieng, Francisco JR Ruiz, and David M Blei. “Topic modeling in embedding spaces”. In: *Transactions of the Association for Computational Linguistics* 8 (2020), pp. 439–453.

[3] Elaine Zosa, Ravi Shekhar, Mladen Karan, and Matthew Purver. “Not All Comments Are Equal: Insights into Comment Moderation from a Topic-Aware Model”. In: *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*. 2021, pp. 1652–1662.

[4] Mike Schuster and Kuldip K Paliwal. “Bidirectional recurrent neural networks”. In: *IEEE transactions on Signal Processing* 45.11 (1997), pp. 2673–2681.

[5] Jan Šnajder, Sebastian Padó, and Željko Agić. “Building and Evaluating a Distributional Memory for Croatian”. In: *51st Annual Meeting of the Association for Computational Linguistics*. 2013, pp. 784–789.

[6] Ravi Shekhar, Marko Pranjić, Senja Pollak, Andraž Pelicon, and Matthew Purver. “Automating News Comment Moderation with Limited Resources: Benchmarking in Croatian and Estonian”. In: *Journal for Language Technology and Computational Linguistics (JLCL)* 34.1 (Sept. 2020).

[7] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. *Learning Transferable Visual Models From Natural Language Supervision*. Tech. rep.

[8] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. “Scaling up visual and vision-language representation learning with noisy text supervision”. In: *arXiv preprint arXiv:2102.05918* (2021).

[9] Donghyun Kim, Kuniaki Saito, Kate Saenko, Stan Sclaroff, and Bryan Plummer. “Mule: Multimodal universal language embedding”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 07. 2020, pp. 11254–11261.

[10] Federico Bianchi, Silvia Terragni, Dirk Hovy, Debora Nozza, and Elisabetta Fersini. “Cross-lingual Contextualized Topic Models with Zero-shot Learning”. In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, Apr. 2021, pp. 1676–1683. URL: https://www.aclweb.org/anthology/2021.eacl-main.143.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *International Conference on Learning Representations*. 2020.

[12] Elaine Zosa, Mark Granroth-Wilding, and Lidia Pivovarova. “A Comparison of Unsupervised Methods for Ad hoc Cross-Lingual Document Retrieval”. In: *LREC 2020 Language Resources and Evaluation Conference 11–16 May 2020*, p. 32.

[13] Andrey Kutuzov, Lilja Øvrelid, Terrence Szymanski, and Erik Velldal. “Diachronic word embeddings and semantic shifts: a survey”. In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, 2018, pp. 1384–1397. URL: http://aclweb.org/anthology/C18-1117.

[14] Nina Tahmasebi, Lars Borin, and Adam Jatowt. “Survey of Computational Approaches to Diachronic Conceptual Change”. In: *CoRR* 1811.06278 (2018).

[15] Xuri Tang. “A state-of-the-art of semantic change computation”. In: *Natural Language Engineering* 24.5 (2018), pp. 649–676.

[16] Renfen Hu, Shen Li, and Shichen Liang. “Diachronic sense modeling with deep contextualized word embeddings: An ecological view”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019, pp. 3899–3908.

[17] Mario Giulianelli, Marco Del Tredici, and Raquel Fernández. “Analysing Lexical Semantic Change with Contextualised Word Representations”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 3960–3973. URL: https://www.aclweb.org/anthology/2020.acl-main.365.

[18] Kristina Gulordava and Marco Baroni. “A distributional similarity approach to the detection of semantic change in the Google Books Ngram corpus.” In: *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*. Edinburgh, UK: Association for Computational Linguistics, 2011, pp. 67–71. URL: http://aclweb.org/anthology/W11-2508.

[19] Matej Martinc, Syrielle Montariol, Elaine Zosa, and Lidia Pivovarova. “Capturing Evolution in Word Usage: Just Add More Clusters?” In: *Companion Proceedings of the Web Conference 2020*. 2020, pp. 343–349.

[20] Dominik Schlechtweg, Barbara McGillivray, Simon Hengchen, Haim Dubossarsky, and Nina Tahmasebi. “SemEval-2020 Task 1: Unsupervised Lexical Semantic Change Detection”. In: *Proceedings of the Fourteenth Workshop on Semantic Evaluation*. 2020, pp. 1–23.

[21] William L Hamilton, Jure Leskovec, and Dan Jurafsky. “Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2016, pp. 1489–1501.

[22] Matej Martinc, Syrielle Montariol, Elaine Zosa, and Lidia Pivovarova. “Discovery Team at SemEval-2020 Task 1: Context-sensitive Embeddings Not Always Better than Static for Semantic Change Detection”. In: *Proceedings of the Fourteenth Workshop on Semantic Evaluation*. 2020, pp. 67–73.

[23] Syrielle Montariol, Matej Martinc, and Lidia Pivovarova. “Scalable and Interpretable Semantic Change Detection”. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2021, pp. 4642–4652.

[24] Ronald W Langacker. *Foundations of cognitive grammar: Theoretical prerequisites*. Vol. 1. Stanford university press, 1987.

[25] Hans Henrich Hock and Brian D Joseph. *Language history, language change, and language relationship: An introduction to historical and comparative linguistics*. Walter de Gruyter GmbH & Co KG, 2019.

[26] Michael Hoey. *Lexical Priming: A New Theory of Words and Language*. Routledge, 2005.

[27] Stefan Th Gries and Dagmar Divjak. “Behavioral profiles: a corpus-based approach to cognitive semantic analysis”. In: *New directions in cognitive linguistics* 57 (2009), p. 75.

[28] Laura A Janda and Olga Lyashevskaya. “Grammatical profiles and the interaction of the lexicon with aspect, tense, and mood in Russian”. In: *Cognitive linguistics* 22.4 (2011), pp. 719–763.

[29] Andrey Kutuzov, Lidia Pivovarova, and Mario Giulianelli. “Grammatical Profiling for Semantic Change Detection”. In: *Proceedings of the 25th Conference on Computational Natural Language Learning*. 2021, pp. 423–434.

[30] Ivan Vulić, Wim De Smet, Jie Tang, and Marie-Francine Moens. "Probabilistic topic modeling in multilingual settings: An overview of its methodology and applications". In: *Information Processing & Management* 51.1 (2015), pp. 111–147.

[31] Jani Marjanen, Elaine Zosa, Simon Hengchen, Lidia Pivovarova, and Mikko Tolonen. "Topic Modelling Discourse Dynamics in Historical Newspapers". In: *Post-Proceedings of the 5th Conference Digital Humanities in the Nordic Countries (DHN 2020), Riga, Latvia, October 21-23, 2020*. Ed. by Sanita Reinsone, Inguna Skadina, Janis Daugavietis, and Anda Baklane. Vol. 2865. CEUR Workshop Proceedings. CEUR-WS.org, 2020, pp. 63–77. URL: http://ceur-ws.org/Vol-2865/paper6.pdf.

[32] Alex Rosenfeld and Katrin 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.

[33] Philippa Shoemark, Farhana Ferdousi Liza, Dong Nguyen, Scott Hale, and Barbara 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.

[34] Quoc Quan Duong, Lidia Pivovarova, and Elaine Zosa. "Benchmarks for Unsupervised Discourse Change Detection". In: *HistoInformatics 2021–6th International Workshop on Computational History*. 2021.

[35] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. "Supervised Contrastive Learning". In: *Advances in Neural Information Processing Systems* 33 (2020).

[36] Jani Marjanen, Lidia Pivovarova, Elaine Zosa, and Jussi Kurunmäki. "Clustering ideological terms in historical newspaper data with diachronic word embeddings". In: *5th International Workshop on Computational History, HistoInformatics 2019*. CEUR-WS. 2019.

[37] Jani Marjanen, Jussi Kurunmäki, Lidia Pivovarova, and Elaine Zosa. "The expansion of isms, 1820-1917: Data-driven analysis of political language in digitized newspaper collections". In: (2020).

A. Manuscript: Topic Modelling Discourse Dynamics in Historical Newspapers

Topic Modelling Discourse Dynamics in Historical Newspapers

Jani Marjanen[1*][0000−0002−3085−4862], Elaine Zosa[2*][0000−0003−2482−0663], Simon Hengchen[3][0000−0002−8453−7221], Lidia Pivovarova[2][0000−0002−0026−9902], and Mikko Tolonen[1][0000−0003−2892−8911]

[1] Helsinki Computational History Group, University of Helsinki
[2] Department of Computer Science, University of Helsinki
[3] Språkbanken, University of Gothenburg[‡]
firstname.lastname@{helsinki.fi,gu.se}

Abstract. This paper addresses methodological issues in diachronic data analysis for historical research. We apply two families of topic models (LDA and DTM) on a relatively large set of historical newspapers, with the aim of capturing and understanding discourse dynamics. Our case study focuses on newspapers and periodicals published in Finland between 1854 and 1917, but our method can easily be transposed to any diachronic data. Our main contributions are a) a combined sampling, training and inference procedure for applying topic models to huge and imbalanced diachronic text collections; b) a discussion on the differences between two topic models for this type of data; c) quantifying topic prominence for a period and thus a generalization of document-wise topic assignment to a discourse level; and d) a discussion of the role of humanistic interpretation with regard to analysing discourse dynamics through topic models.

Keywords: discourse dynamics, Finland, historical newspapers, nineteenth century, topic modeling, topic modelling

1 Introduction

This paper reports our experience on studying discursive change in Finnish newspapers from the second half of the nineteenth century. We are interested in grasping broad societal topics, discourses that cannot be reduced to mere words, isolated events or particular people. Our long-lasting goal is to investigate a global change in the presence of such topics and especially finding discourses that have disappeared or declined and thus could easily slip away in modern research. We believe that these research questions are better approached in a data-driven way without deciding what we are looking for beforehand, though the choice of the most suitable techniques for such research is still an open problem.

In this paper we focus on developing methodology. Choosing available algorithms for analysis guides possible outcomes as they are designed to be operationalised in

[‡] SH was affiliated with the University of Helsinki for most of this work.
[*] Equal contribution.

certain ways. Approaching our goal with mere word counts is counterproductive due to the sparseness of the language and the variety of discourse realisations in a given text. Further, word counts are unreliable with historical data due to never ending language change, spelling variations and text recognition errors.

Thus, as many other papers in the area of digital humanities, we utilize topic modelling as a proxy to discourses. In particular, we apply the "standard" Latent Dirichlet Allocation model [3, LDA] and its extension the Dynamic Topic Model [2, DTM], which is developed specifically to tackle temporal dynamics in data. However, any model has its limitations and tends to exaggerate certain phenomena while missing other ones. We focus on the difference between models and try to reveal their limitations in historical data analysis from the point of view that is relevant for historical scholarship.

Our main contributions are the following:

- We propose a **combined sampling, training and inference procedure** for applying topic models to large and imbalanced diachronic text collections.
- We discuss differences between two topic models, paying special attention to how they **can be used to trace discourse dynamics**.
- We propose a method to quantify **topic prominence for a period** and thus to generalize document-wise topic assignment to a discourse level.
- We **acknowledge and discuss the drawbacks of topic stretching,** which is typical for DTM. It is commonly known that DTM sometimes represents topics beyond the time period, but thus far there is no discussion in how researchers should tackle this for humanities questions.

In order to illustrate the appropriateness of the proposed methodology we discuss two use cases, one relating to discourses on church and religion and one that relates to education. The role of religion and education has been studied extensively in historical scholarship but there are no studies that deal with these topics through text mining of large-scale historical data. These two topics were chosen due to the the fact that the former was in general a discourse in decline relating to the process of secularization in Finnish society, whereas the latter increased in the second half of the nineteenth century and relates to the modernization of Finnish society and the inclusion of a larger share of the population in the sphere of basic education. In addition to these two interlinked discursive trends, we also use other examples to illustrate the strengths and weaknesses of LDA and DTM for this type of historical research.

2 Data

Our dataset is from the digitised newspaper collection of the National Library of Finland (NLF). This dataset contains articles from *all* newspapers and most periodicals that have been published in Finland from 1771 to 1917. Several studies have used parts of this dataset to investigate such issues as the development of the public sphere in Finland, the evolution of ideological terms in nineteenth-century Finland and the changing vocabulary of Finnish newspapers [36, 17, 16, 11, 21, 22, 25, 29, 12].

The full collection includes articles in Finnish, Swedish, Russian, and German. In this work we focus only on the Finnish portion starting from 1854 because this is the point where we determined we have sufficient yearly data to train topic models. The resulting subset has over 3.6 million articles and is composed of over 2.2 billion tokens. Figure 1a shows that the number of tokens published per year in Finnish-language papers increased steadily. The average article has 526 tokens but article length varies widely from year to year, as seen in Figures 1b and 1c which show the average article length and the number of articles per year. As made clear by these figures, there is a noticeable difference in the number of articles and average article length after 1910. This shift does not reflect the actual articles in the newspapers, but is the result of a change of OCR engine used to digitise the collection [20]. While the raw data is publicly available, we used the lemmatised version of the newspaper archive produced by Eetu Mäkelä, whom we thank.

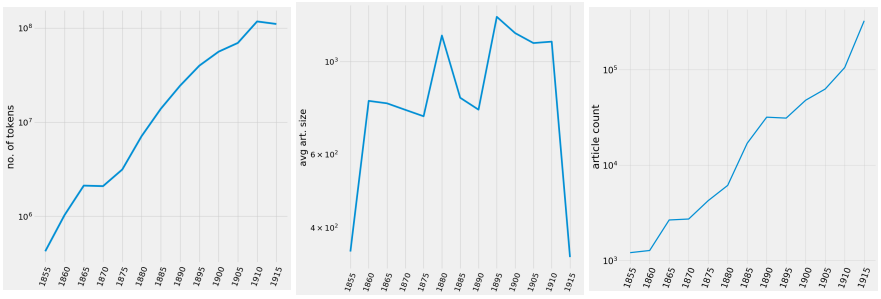
Still, even if the article segmentation differs in the latter period, Fig. 1a shows that there is steady increase in the vocabulary used in the Finnish-language newspapers published in the second half of the nineteenth century. They also covered more themes and regions. This entailed a process of diversification and modernization of the Finnish press, which has been widely discussed in historiography. As a collection, the newspapers vary a lot in style and focus. Some larger newspapers mainly contain political content, whereas others are rather specialised, and yet others thrived by giving a voice to the local public [35, 22, 16, 32]. This means that any analysis done on the entirety of the newspapers, like topic models, tend to balance out some of the differences between newspapers. This variety in the content, is also something that make newspapers such an interesting source material for historical research that is interesting in an overview of society. Although some issues were obviously not discussed because of taboo, courtesy or censorship, most of the themes present in public discourse are recorded in the newspapers and thus accessible to us in the present. Hence, we believe newspapers are an especially good source of assessing how the role of particular discourses changed over time.

2.1 Preprocessing the data

Given the size of the data and its inherent nature, notoriously the OCR quality and the unbalanced data from different time slices, we performed a series of pre-processing steps on the data.[1]

Despite prior work (albeit on English), showing that stemming has no real advantage for likelihood and topic coherence and can actually degrade topic stability [30], we follow [40, 10, 13] and use a lemmatised version of the corpus. Indeed, the work in [10] hints at the fact that Finnish, being much more inflected than English, would benefit from lemmatisation, whereas in [40, 13] the authors stem so as to reduce the huge number of token types due to OCR issues which impacts the performance of topic

[1]The more apt phrase "purposeful data modification", coined by [34], advocates that our material is not mere data that can go through a standardised "pre-processing" pipeline. Rather, the data is modified and altered only for the specific purposes of this study, and following this study's technical and scientific requirements only.



(a) Corpus size (b) Average article size (c) No. of articles

Fig. 1: Characteristics of the NLF dataset

modelling [38]. After lemmatisation, we remove tokens that occur less than 40 times in the collection, stopwords, punctuation marks and tokens with less than 3 characters. These are additional measures to further reduce the vocabulary size and mitigate the impact of OCR noise.

3 Topic Models

3.1 LDA

Topic modelling is an unsupervised method to extract topics from a collection of documents. Typically, a topic is a probability-weighted list of words that together express a theme or idea of what the topic is about. One of the most popular topic modelling methods currently in use is Latent Dirichlet Allocation (LDA), which is "a generative probabilistic model for collections of discrete data such as text corpora" [3]. It has been extensively used in the digital humanities to extract certain themes from a collection of texts [4]. In this model, a document is a mixture of topics and a topic is a probability distribution over a vocabulary. A limitation of LDA for historical research, in its vanilla form, is that it does not account for the temporal aspect of the data: every document in the collection is "considered synchronic", as time is simply not a variable in the model. Many document collections such as news archives, however, are diachronic—the documents are from different points in time, and scholars wish to study the evolution of topics.

There are different ways to overcome this limitation. One possibility is to split the data into time slices and train LDA separately on each slice. However, in this case LDA models for each slice would be independent of each other and there is no straightforward approach of matching topics from independent models trained on disjoint data. Another possibility, which we explore in this paper, is to train a single model for a subset of the whole data set over the entire time period and then use *topic prominence* as proxy for the dynamics of discourses over time.

To do this, we compute the prominence of a topic in a given year by summing up the topic contribution for each document in that year and then normalise this number by the sum of all topic contributions from all topics for that year, as in Equation 1.

$$P(z_k|y) = \frac{\sum_{j=1}^{|D_y|} P(z_k|d_j)}{\sum_{i=1}^{T} \sum_{j=1}^{|D_y|} P(z_i|d_j)} \quad (1)$$

where y is a year in the dataset, k is a topic index, D_y is the number of documents in year y, d_j is the j^{th} document in year y and T is the number of topics in the model.

The large size of the collection and its unbalanced nature is a problem for training topic models. It is computationally expensive to train a model with millions of articles and the resulting model would be heavily biased towards the latter years of newspaper collection because it has far more data. To overcome these issues, we sampled the collection such that we have a roughly similar data size for each year of the collection and as a result, we also get a vastly reduced dataset. However, to have a model of discourse dynamics that reflects the collection more closely, we compute topic prominence using the entire collection and not just the sampled portion. We do this by inferring the topic proportions of all the documents in the collection and using these inferred distributions to compute topic prominence.

3.2 DTM

As mentioned above, there are topic models that explicitly take into account the temporal dynamics of the data. One such model is the dynamic topic model (DTM). DTM is an extension of LDA that is designed to capture dynamic co-occurence patterns in diachronic data. In this model, the document collection is divided into discrete time slices and the model learns topics in each time slice with a contribution from the previous time slice. This results in topics that evolve slightly–words changing in saliency in relation to a topic–from one time step to the next.

However, DTM also has its own limitations. It is based on an assumption that each topic should be to some extent present in each time slice, which is not always the case with real-world data such as news archives where events and themes can sometimes disappear and then re-appear at some point in the future.

Perhaps more importantly for historical research, a weakness of DTM lies in its design: to accomplish alignment across time the topic model is fit across the whole vocabulary and thus smoothing between time slices is applied. As a result, events end up being "spread out" before and after they are known to happen. This problem only becomes evident after a thorough analysis: similar models in different fields such as lexical semantic change present the same issue – the dynamic topic model SCAN [7] generates a "plane" top word for the year 1700 (two centuries ahead of the Wright Flyer, and well before the word's first attested sense of "aeroplane"), while similar model GASC [26, 23] encounters the same weakness when modelling Ancient Greek. There is unfortunately no easy way to bypass this obstacle, which is particularly problematic when studying historical themes.

For both the LDA and DTM models, we use the Gensim implementation [28] with default model hyperparameters.

4 Related Work

Topic models are widely used in the digital humanities and social sciences to draw insights from large-scale collections [4] ranging from newspaper archives to academic journals. In this section, which we do not claim to be exhaustive, we discuss some of the previous works that aimed to capture historical trends in large data collections or used such collections to study discourses using topic models. All in all, these examples highlight that there is a need to discuss how topic models can be used to capture discursive change.

In [24] the authors use Latent Semantic Analysis, another topic modelling method, to study historical trends in eighteenth-century colonial America with articles from the *Pennsylvania Gazette*. Their work also used topic prominence to show, for instance, an increased interest in political issues as the country was heading towards revolution. The authors of [40] fit several topic models on Texan newspapers from 1829 to 2008. To discover interesting historical trends, the authors slice their data into four time bins, each corresponding to historically relevant periods. Such a slicing is also carried out in [9], where the author fits LDA models on Dutch-language Belgian socialist newspapers for three time slices that are historically relevant to the evolution of workers rights, with the aim of generating candidates for lexical semantic change.

Topic modelling has also been used in discourse analysis of newspaper data. In [37] the authors applied LDA to a selection of Italian ethnic newspapers published in the United States from 1898 to 1920 to examine the changing discourse around the Italian immigrant community, as told by the immigrants themselves, over time. They proposed a methodology combining topic modelling with close reading called discourse-driven topic modelling (DDTM). Another study examined anti-modern discourse in Europe from a collection of French-language newspapers [5]. In this case, however, the authors primarily use LDA as a tool to construct a sub-corpus of relevant articles that was then used for further analysis. Modernization was also an issue in the study of Indukaev [14], who uses LDA and word embeddings to study changing ideas of technology and modernization in Russian newspapers during the Medvedev and Putin presidencies.

LDA was not designed for capturing trends in diachronic data and so several methods have been developed to address this, such as DTM, Topics over Time [39, TOT], and the more recent Dynamic Embedded Topic Model [6, DETM], an extension of DTM that incorporates information from word embeddings during training. As far as we are aware, DTM and TOT have not been used for historical discourse analysis or applied to large-scale data collections. In the original papers presenting these methods, DTM was applied to 30,000 articles from the journal *Science* covering 120 years and TOT was applied to 208 State of the Union Presidential addresses covering more than 200 years. This was to demonstrate the evolution of scientific trends for the former and the localisation of significant historical events for the latter. Recently DETM was applied on a dataset of modern news articles about the COVID-19 pandemic where the authors observed differences between countries in how the pandemic and the reactions to it were framed [19].

In the mentioned cases researchers tackle the interpretative part of using topic models for humanistic research in different ways. Like Pääkkönen and Ylikoski [27] state, they toggle between some sort of topic realism, that is, using topic models to grasp

something that exists in the data, and topic instrumentalism, that is, using topic models to find something that can be further studied. Only Bunout [5] is a clear case of topic instrumentalism. All the other studies depart from some sort of realist position, and attempt to grasp policy shifts, ideas, discourses or framings of topics through topic models, but end up with correctives of some kind by highlighting the interpretative element [24, 37], by deploying formal evaluation by historians [9] or by using other quantitative methods to fine tune the results [14]. The interpretative aspect seems especially important when it comes to deciding on what researchers use the topics to study as they can reasonably relate to historical discourses, the semantics of related words, or simply ideas. How the topics are seen to represent these or, more likely, how the researchers use the topics to make an interpretation about these based on the topics, requires a strong element of interpretation [27]. Studies show that interpreters prefer to be able to go back to actual texts in order to make sense of topics [18], which is more than reasonable, but it also seems that there is a further need for researchers to understand how different topic-modelling methods represent diachronic data. Without this knowledge it is difficult to assess to which degree and for which time periods researchers need to manually assess individual documents.

5 Use Cases

What a discourse is, has been heavily theorised within the different strands of discourse analysis [1], but the advent of digital methods that can handle large textual data sets require quite some adjustment of discourse analysis as we know it. Like this article, others have turned to topic models to grasp changes in discourse [37, 5], but this article seeks specifically to discuss the interpretation that is required when we use topic models to study discourse dynamics. The probabilistic topic models set clear boundaries between topics and in doing so might merge or separate things that historians might regard as coherent topics. However, where the probabilistic model enforces boundaries, human interpretation in general is very bad at setting those boundaries and usually just identifies the core of a discourse or topic, but cannot say where it ends.

To get at the tension between topics and discourses, we approached the material without a predefined idea about which topics we wanted to study in order to keep the study as data-driven as possible. Our interest was to use topic modelling to capture topics that could in a meaningful way be related to societal discourses, that is themes that cannot be narrowed down to individual words, but still are reasonably coherent and form at least loose topics. To this end, we trained topic models with $k \in \{30; 50\}$, inferred topic distributions for the whole collection and inspected models by carefully going through the top words in each topic and using PyLDAVis[2] [31] to study overlap between topics and salience of terms per topic in LDA and heatmap visualizations for DTM. All topics were annotated and evaluated from the point of view of historical interpretation. We then opted to use the 50-topic model to study discourse changes over time. As is common, a portion of the topics seemed incoherent or were clearly the result of the layout in newspapers (e.g. boilerplate articles about prices etc.) and

[2] `https://github.com/bmabey/pyLDAvis`

did not produce interesting information about societal discourses. Further, some of the topics clearly overlap, so that a cluster of 2-5 topics can reasonably be seen as related to a particular societal discourse. The advantage of choosing 50 topics over 30 lies precisely in the possibility of merging topics later on in interpretation, while splitting them is more difficult.

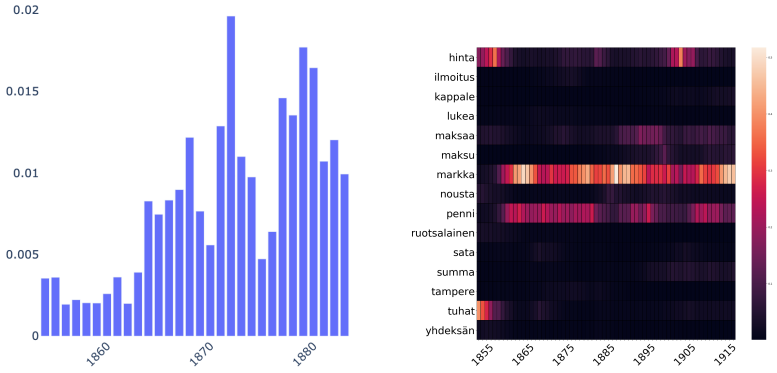
To discuss the benefits of LDA and DTM, we chose to focus on two specific themes, the discourse relating to religion and religious offices, and education. They are both rather neatly identifiable in the data, but display different trends. The former is in decline over the period of interest, whereas the latter increases in topic prominence. They can also be related to large scale processes in Finland, religious discourse to the secularization of society and education to the modernization of civic engagement.

5.1 DTM and Stretching of Topics

The two topic modelling methods perform in somewhat different ways. As mentioned, DTM is designed to incorporate temporal change in the topics, which means it includes a stronger sense of continuity in its representations of data. Whether or not this is desirable, depends on the research question, but our contention is that for studies interested in discursive change, this is either a problem or at least it is something that needs to be factored in making the historical interpretation. If we want to understand when certain discourses became dominant, declined, or even disappeared, this type of stretching cannot be allowed.

An exceptionally illustrative example of stretching among our fifty topics, is an introduction of the Finnish mark as a currency (Fig. 2a). With top words such as “mark”, “penny”, “price”, “thousand”, “pay” etc. the topic comes across as one with high internal coherence. We also see that the topic grows in prominence over time, from being relatively modest in the 1850s to gradually increased prominence after 1860. This makes sense, as the mark was adopted as currency in the year 1860 and after that self-evidently figured in public discourse. However, when we look at a heatmap visualization of the topic (Fig. 2b), we see how the topic stretches from the period 1854–1859 to the period 1860–1917, that is, from the period before the introduction of the mark to the period it was in use. After 1860 the words “mark” and “penny” are by far the most dominant terms in the topic, but for the period before 1860, the dominant terms are “price” and “thousand.” It is clear that “mark”, “penny”, “price”, and “thousand” are words that can belong to the same topic, but the heatmap representation clearly shows that the focus in the topic shifts. It is almost as if two related topics are merged as to represent one topic over the whole time period. In a situation where a historical interpretation highlights a change in past discourse, DTM produces continuity.

While there is obviously no right answer as to when one topic is stretched a bit or when different topics are simply merged together to provide a temporally continuous topic, it seems that DTM is especially problematic if one wants to study discourses that emerge or disappear in the middle of a time period studied. This means that any historical analysis using DTM requires a component of historical interpretation of not only topic coherence, but also topic coherence *over time*. Here, relying on word embeddings like in [14] can help, but this is primarily a task for evaluating the topics.



(a) Introduction of the Finnish mark in 1860 (y-axis indicates the topic probability)

(b) Heat map of terms linked to the introduction of the Finnish mark in 1860.

Fig. 2: Topic related to the introduction of the Finnish mark in 1860 (DTM). The most prominent terms in the heatmap are are "Mark" = *markka*, "penny" = *penni*, "price" = *hinta*, "thousand" = *tuhat*, "pay" = *maksu* and *maksaa*.

The speed of topic evolution can be controlled by a parameter in the DTM model. However, the 'ideal' amount of stretching is difficult to assess. For analysing discourse, this might in some cases be productive as it can point at links between nearby discourses, but is largely problematic as it hides discontinuities in the data. It becomes even problematic when dealing with material factors, like the introduction of the Finnish mark, as the stretching effect is likely to produce anachronistic representations, that is, placing something in the wrong period of time. Dealing with anachronism can perhaps be seen as one of the cornerstones of the historian's profession, which makes DTM as an anachronism prone method a poor match for historical study. Avoiding anachronisms completely is impossible, most historians would agree, but knowing when to avoid them and how to communicate about anachronistic elements in historical interpretation is key to history as a discipline [33].

5.2 Religion and Secularization

Our model performed well in grasping topics that relate to religion. The initial expectation regarding the discourse dynamics was that religious topics would be in decline. We hoped that using a topic model would be a way of showing this quantitatively. Results obtained from both LDA and DTM, presented in Figures 3a and 3b respectively, harmonize with our initial hypothesis, but do so differently. The DTM and LDA outputs cannot be aligned in any other way than manual interpretation by domain experts. In doing this we simply regarded topics that included several words that denote religious practices or offices as religious. Thus, the definition of "religious" is is rather narrow, but it also seems to match the topics that emerged from our data.

In order to inspect the discourse dynamics of religious topics, we have combined several topics that related to religious themes in the LDA model, whereas in the latter, DTM model, we only chose one topic to be represented.[3]

To our knowledge, topic models have not been used to study discursive change regarding secularization. However, in line with some earlier qualitative assessments [15], we hypothesize that this decline in religious discourse entails two interrelated developments: 1) Religion did not disappear from public discourse, but instead changed and disappeared from certain *types* of discourses. In the early nineteenth century, religion had a much more holistic presence in public discourse, meaning that religious metaphors and religious expressions and topics were used at a much vaster scale. 2) Over the course of the nineteenth century, religious topics became more focused. This means a segmentation of public discourse so that religious topics were increasingly confined to particular journals or genres.

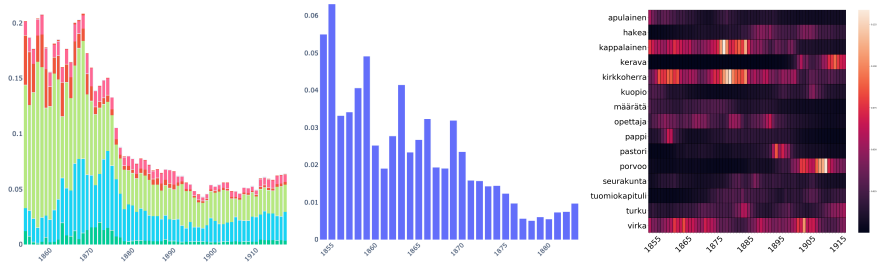
Keeping in mind the issue of stretching with DTM, we can look into the shifting saliency of words within the topic of religious offices and notice a shifting focus over time (Fig. 3c). In the early 1900s terms relating to "holding an office" and names of particular congregations become more dominant in the topic. This, again, suggests that DTM as a method does some stretching. There is a downside and an upside to this. On the one hand, the stretching distorts the topic prominence a bit by making it look like there is more continuity than in the LDA visualization. However, this may not be that crucial as the declining trends in Fig. 3a and Fig. 3b are rather similar. On the other hand, the stretching may be good for detecting conceptual links between different groups of words. In this particular case the stronger link between religious offices and some towns like Kerava and Porvoo, is probably indicative of a move of religious discourse from an overarching question to something that is more likely dealt with in conjunction to matters at local parishes. That is, religious offices were more often than before dealt with in connection to local congregations. This is in line with our above-mentioned assumption about religious discourse becoming more distinct.

5.3 Education and Modernity

While we expected religious themes to decline and become less central, we assumed there would be some themes that partly overlap with religion, but also would show an increasing trend. One example of this is the topic of education, which has historically been heavily interwoven with the church, but at the same time when basic education became available for a higher amount of people, it also became central in questioning the role of the church and religion. Education in nineteenth-century Finland was both central for ensuring conformity of the Lutheran faith, but paradoxically also was a vehicle of secularization. [8]

As in the case of religious discourse, alignment between DTM and LDA can only be made through human interpretation. It seems, that in this case DTM captures one topic

[3] We also experimented with more data-driven methods to cluster topics, including for example methods based on Jensen-Shannon Divergence. They unfortunately did not need to clusters that our domain experts would make sense of. Nonetheless, despite this, we still believe this is an interesting avenue to pursue which could help answer the common 'number of topics' question often brought up within the field.



(a) Topics related to religion on decline (LDA)

(b) Development of religious topic (chaplain, priest and office) over time

(c) Heatmap of terms linked to office of religion topic.

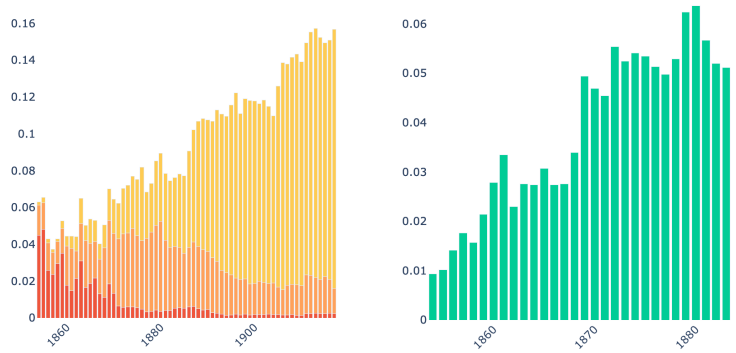
Fig. 3: Religious topics in LDA (a) and DTM (b,c); y-axis in (a, b) indicates the topics' probabilities. Most prominent terms in the heatmap are "chaplain" = *kappalainen*, "vicar" = *kirkkoherra*, "teacher" = *opettaja*, "priest" = *pappi*, "Porvoo" (a town), "parish" = *seurakunta*, "Turku" (a town), and "office" = *virka*.

that is fairly coherent, revolves around education and schooling, and is on the rise in the research period (Fig. 4b). For LDA, this is not the case, as an PyLDAVis inspection of most salient words across all fifty topics show that words like "school" and "folk school" appear mostly in three topics of which two are in decline and one heavily on the rise (Fig. 4a).

Interestingly, LDA and DTM seem to be pointing at a similar historical development. The two declining LDA topics are based on their most salient terms and are more focused on schools as buildings and institutions as well as teaching as a profession, whereas the topic on the rise includes salient vocabulary relating to, not only schools, but also meetings, civic engagements, and decision making. The DTM topic at hand shows a similar development which can be inspected in a heatmap of most salient terms over time. The terms "school", "child", and "teacher" dominate early in the period. By the end of the period the topic becomes broader, and terms like "municipality" and "meeting" have become more salient than the vocabulary relating to schools. Here the stretching of DTM creates the links that are also visible in the three LDA topics, and it shows a transformation in which educational issues are present in the whole topic, but focus shifts from concrete schools to civic engagement.

6 Conclusions

Our focus in this text has been on discourses that cannot be reduced to mere words, isolated events or particular people, but concern broader societal topics that either declined or gained in prominence. The interpretation of these topics and their contextualisation to nineteenth-century Finnish newspapers revealed clear topical cores that can be interpreted as an encouraging point of departure for further explorations based on topic models when aiming to understand Finnish public discourse through historical newspapers.



(a) Development of education topic over time (LDA)

(b) Development of education topic over time (DTM)

Fig. 4: Education topic in LDA and DTM; y-axis indicates the topics' probabilities

In this paper, we have learned that although it is difficult to pinpoint exactly where a discourse or topic ends, LDA and DTM can fairly reliably grasp many semi-coherent themes in past discourse and help us study the dynamics of discourses. However, our comparison of LDA and DTM as methods for getting at past discourse also shows that both methods require a very strong interpretative element in analysing historical discourses. DTM is much more prone to stretch or even merge topics, which requires an interpretative assessment of whether the stretching highlights interesting historical continuities or if it hides historical discontinuities that would require attention. We found that producing heatmaps of term saliency over time for each topic is a very useful way of doing this type of assessment. For LDA, stretching is not so much a problem, but often it seems interpretation is needed in seeing which topics logically relate to one another. While historical discourse analysis is traditionally tied strongly to a tradition of hermeneutic interpretation, the use of topic models to grasp discourse dynamics does not remove that need even if they allow for a quantification of discourse dynamics over time.

While we regard stretching in DTM as a predominantly negative feature, in some cases it can be useful. In the topics relating to education discussed above, the stretching in DTM actually points out links in discourses and is quite productive for the interpretative process of trying to figure out discourse dynamics. However, also in this case, the relevance of historical interpretation should be highlighted because it is very hard to tell whether the stretching of topics is an accurate reflection of the data or a shortcoming of the model. This can be addressed only by relating visualisations of topics to existing historical research and reading source texts. Humanities scholars are in general very good at making such interpretations, but it also needs to be noted that when we move further into the domain interpretative scholarship, we also lose some of the benefits of working with quantifying models. While it would be foolish to claim that a topic model represents data in a way that it provides simple facts about historical development, our use cases show that if we seek to find more reliable quantification LDA may

provide better results than DTM. Further, using LDA moves the interpretative stage further down in the research process, as it is likely to be about evaluating the connections between different topics over time. In DTM, the interpretation is likely moved forward to an evaluation of how well the algorithm did this merging topics. On this sense, our take on topic models harmonises with [27] who stress the role of humanistic interpretation, but for the sake of transparency suggest pushing the interpretation stage later in the research process.

Acknowledgements

This work has been supported by the European Union Horizon 2020 research and innovation programme under grants 770299 (NewsEye) and 825153 (EMBEDDIA). SH is funded by the project *Towards Computational Lexical Semantic Change Detection* supported by the Swedish Research Council (2019–2022; dnr 2018-01184).

References

1. Angermuller, J., Maingueneau, D., Wodak, R. (eds.): The discourse studies reader: Main currents in theory and analysis. John Benjamins Publishing, Amsterdam, the Netherlands ; Philadelphia PA (2014)
2. Blei, D.M., Lafferty, J.D.: Dynamic topic models. In: Proceedings of the 23rd international conference on Machine Learning. pp. 113–120 (2006)
3. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. Journal of Machine Learning Research **3**(Jan), 993–1022 (2003)
4. Brauer, R., Fridlund, M.: Historicizing topic models, a distant reading of topic modeling texts within historical studies. In: International Conference on Cultural Research in the context of Digital Humanities, St. Petersburg: Russian State Herzen University (2013)
5. Bunout, E.: Grasping the anti-modern discourse on Europe in the digitised press or can text mining help identify an ambiguous discourse? (2020)
6. Dieng, A.B., Ruiz, F.J., Blei, D.M.: The dynamic embedded topic model. arXiv preprint arXiv:1907.05545 (2019)
7. Frermann, L., Lapata, M.: A Bayesian model of diachronic meaning change. Transactions of the Association for Computational Linguistics **4**, 31–45 (2016)
8. Hanska, J., Vainio-Korhonen, K. (eds.): Huoneentaulun maailma: kasvatus ja koulutus Suomessa keskiajalta 1860-luvulle. Suomalaisen Kirjallisuuden Seuran toimituksia, 1266:1, Suomalaisen kirjallisuuden seura, Helsinki (2010), publication Title: Huoneentaulun maailma : kasvatus ja koulutus Suomessa keskiajalta 1860-luvulle
9. Hengchen, S.: When Does it Mean? Detecting Semantic Change in Historical Texts. Ph.D. thesis, Université libre de Bruxelles (2017)
10. Hengchen, S., Kanner, A.O., Marjanen, J.P., Mäkelä, E.: Comparing topic model stability between Finnish, Swedish, English and French. In: Digital Humanities in the Nordic Countries (2018)
11. Hengchen, S., Ros, R., Marjanen, J.: 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)
12. Hengchen, S., Ros, R., Marjanen, J., Tolonen, M.: A data-driven approach to studying changing vocabularies in historical newspaper collections. Digital Scholarship in the Humanities (2021)

13. Hill, M.J., Hengchen, S.: Quantifying the impact of dirty OCR on historical text analysis: Eighteenth century collections online as a case study. Digital Scholarship in the Humanities **34**(4), 825–843 (2019)
14. Indukaev, A.: Studying Ideational Change in Russian Politics with Topic Models and Word Embeddings. In: Gritsenko, D., Wijermars, M., Kopotev, M. (eds.) Palgrave Handbook of Digital Russia Studies. Palgrave Macmillan, Basingstoke (2021)
15. Juva, M.: Valtiokirkosta kansankirkoksi: Suomen kirkon vastaus kahdeksankymmentäluvun haasteeseen. WSOY, Porvoo (1960)
16. Kokko, H.: Suomenkielisen julkisuuden nousu 1850-luvulla ja sen yhteiskunnallinen merkitys. Historiallinen Aikakauskirja **117**(1), 5–21 (2019)
17. La Mela, M., Tamper, M., Kettunen, K.: Finding Nineteenth-century Berry Spots: Recognizing and Linking Place Names in a Historical Newspaper Berry-picking Corpus. In: Navarretta, C., Agirrezabal, M., Maegaard, B. (eds.) DHN 2019 - Digital Humanities in the Nordic Countries. pp. 295–307. CEUR Workshop Proceedings, CEUR (2019), https://cst.dk/DHN2019/DHN2019.html
18. Lee, T.Y., Smith, A., Seppi, K., Elmqvist, N., Boyd-Graber, J., Findlater, L.: The human touch: How non-expert users perceive, interpret, and fix topic models. International Journal of Human-Computer Studies **105**, 28–42 (Sep 2017). https://doi.org/10.1016/j.ijhcs.2017.03.007, https://linkinghub.elsevier.com/retrieve/pii/S1071581917300472
19. Li, Y., Nair, P., Wen, Z., Chafi, I., Okhmatovskaia, A., Powell, G., Shen, Y., Buckeridge, D.: Global surveillance of covid-19 by mining news media using a multi-source dynamic embedded topic model. In: Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics. pp. 1–14 (2020)
20. Mäkelä, E., Tolonen, M., Marjanen, J., Kanner, A., Vaara, V., Lahti, L.: Interdisciplinary collaboration in studying newspaper materiality. In: Krauwer , S., Fišer, D. (eds.) Twin Talks Workshop at DHN 2019. pp. 55–66. CEUR Workshop Proceedings, CEUR-WS.org, Germany (2019)
21. Marjanen, J., Pivovarova, L., Zosa, E., Kurunmäki, J.: Clustering ideological terms in historical newspaper data with diachronic word embeddings. In: 5th International Workshop on Computational History, HistoInformatics 2019. CEUR-WS (2019)
22. Marjanen, J., Vaara, V., Kanner, A., Roivainen, H., Mäkelä, E., Lahti, L., Tolonen, M.: A national public sphere? Analyzing the language, location, and form of newspapers in Finland, 1771–1917. Journal of European Periodical Studies **4**(1), 54–77 (2019)
23. McGillivray, B., Hengchen, S., Lähteenoja, V., Palma, M., Vatri, A.: A computational approach to lexical polysemy in Ancient Greek. Digital Scholarship in the Humanities **34**(4), 893–907 (2019)
24. Newman, D.J., Block, S.: Probabilistic topic decomposition of an eighteenth-century american newspaper. Journal of the American Society for Information Science and Technology **57**(6), 753–767 (2006)
25. Oiva, M., Nivala, A., Salmi, H., Latva, O., Jalava, M., Keck, J., Domínguez, L.M., Parker, J.: Spreading News in 1904: The Media Coverage of Nikolay Bobrikov's Shooting. Media History **26**(4), 391–407 (Oct 2020). https://doi.org/10.1080/13688804.2019.1652090, https://www.tandfonline.com/doi/full/10.1080/13688804.2019.1652090
26. Perrone, V., Palma, M., Hengchen, S., Vatri, A., Smith, J.Q., McGillivray, B.: GASC: Genre-aware semantic change for Ancient Greek. In: Proceedings of the 1st International Workshop on Computational Approaches to Historical Language Change. pp. 56–66. Association for Computational Linguistics, Florence, Italy (Aug 2019). https://doi.org/10.18653/v1/W19-4707, https://www.aclweb.org/anthology/W19-4707

27. Pääkkönen, J., Ylikoski, P.: Humanistic interpretation and machine learning. Synthese (Sep 2020). https://doi.org/10.1007/s11229-020-02806-w, http://link.springer.com/10.1007/s11229-020-02806-w
28. Řehůřek, R., Sojka, P.: Software Framework for Topic Modelling with Large Corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. pp. 45–50. ELRA, Valletta, Malta (May 2010), http://is.muni.cz/publication/884893/en
29. Salmi, H., Paju, P., Rantala, H., Nivala, A., Vesanto, A., Ginter, F.: The reuse of texts in Finnish newspapers and journals, 1771–1920: A digital humanities perspective. Historical Methods: A Journal of Quantitative and Interdisciplinary History pp. 1–15 (Sep 2020). https://doi.org/10.1080/01615440.2020.1803166, https://www.tandfonline.com/doi/full/10.1080/01615440.2020.1803166
30. Schofield, A., Mimno, D.: Comparing apples to apple: The effects of stemmers on topic models. Transactions of the Association for Computational Linguistics **4**, 287–300 (2016)
31. Sievert, C., Shirley, K.: Ldavis: A method for visualizing and interpreting topics. In: Proceedings of the workshop on interactive language learning, visualization, and interfaces. pp. 63–70 (2014)
32. Sorvali, S.: "Pyydän nöyrimmästi sijaa seuraavalle" – Yleisönosaston synty, vakiintuminen ja merkitys autonomian ajan Suomen lehdistössä. Historiallinen Aikakauskirja **118**(3), 324–339 (2020)
33. Syrjämäki, S.: Sins of a historian: Perspectives on the problem of anachronism. Ph.D. thesis, Tampere University Press, Tampere (2011), oCLC: 816367378
34. Thompson, L., Mimno, D.: Authorless topic models: Biasing models away from known structure. In: Proceedings of the 27th International Conference on Computational Linguistics. pp. 3903–3914. Association for Computational Linguistics, Santa Fe, New Mexico, USA (Aug 2018), https://www.aclweb.org/anthology/C18-1329
35. Tommila, P., Landgrén, L.F., Leino-Kaukiainen, P.: Suomen lehdistön historia 1. Sanomalehdistön vaiheet vuoteen 1905. Kustannuskiila, Kuopio (1988)
36. Vesanto, A., Nivala, A., Rantala, H., Salakoski, T., Salmi, H., Ginter, F.: Applying BLAST to text reuse detection in finnish newspapers and journals, 1771-1910. In: Proceedings of the NoDaLiDa 2017 Workshop on Processing Historical Language. pp. 54–58 (2017)
37. Viola, L., Verheul, J.: Mining ethnicity: Discourse-driven topic modelling of immigrant discourses in the USA, 1898–1920. Digital Scholarship in the Humanities (2019)
38. Walker, D., Lund, W.B., Ringger, E.: Evaluating models of latent document semantics in the presence of ocr errors. In: Proceedings of the 2010 conference on empirical methods in natural language processing. pp. 240–250 (2010)
39. Wang, X., McCallum, A.: 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. pp. 424–433 (2006)
40. Yang, T.I., Torget, A., Mihalcea, R.: Topic modeling on historical newspapers. In: Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities. pp. 96–104 (2011)

B. Manuscript: Insights into Comment Moderation from a Topic-aware Model

Not All Comments are Equal: Insights into Comment Moderation from a Topic-Aware Model

Elaine Zosa
University of Helsinki
elaine.zosa@helsinki.fi

Ravi Shekhar
Queen Mary University of London
r.shekhar@qmul.ac.uk

Mladen Karan
◇Queen Mary University of London
m.karan@qmul.ac.uk

Matthew Purver◇,†
†Jožef Stefan Institute
m.purver@qmul.ac.uk

Abstract

Moderation of reader comments is a significant problem for online news platforms. Here, we experiment with models for automatic moderation, using a dataset of comments from a popular Croatian newspaper. Our analysis shows that while comments that violate the moderation rules mostly share common linguistic and thematic features, their content varies across the different sections of the newspaper. We therefore make our models topic-aware, incorporating semantic features from a topic model into the classification decision. Our results show that topic information improves the performance of the model, increases its confidence in correct outputs, and helps us understand the model's outputs.

1 Introduction

Most newspapers publish their articles online, and allow readers to comment on those articles. This can increase user engagement and page views, and provides readers with an important route to public freedom of expression and opinion, with the ability to interact and discuss with others. Comment sections usually provide some degree of anonymity;[1] while improving accessibility, this can also encourage inappropriate behaviour, and publishers therefore usually employ some moderation policy to regulate content and to ensure legal compliance (in some cases, publishers can be held responsible for user-contributed content on their sites).

One possible approach is a 'moderate then publish' policy, in which comments must be approved by a moderator before they appear; this requires significant manpower and introduces delays and limitations into the user conversation (for example, the New York Times only allows comments for one day after article publication[2]). On the other hand, a 'publish then moderate' strategy, in which comments are published immediately, and later removed if necessary, is less effective at blocking toxic or illegal content. Combined with the increase in comment volumes in recent years there is increasing interest in automatic moderation methods (see e.g. Pavlopoulos et al., 2017a), either as standalone tools or for integration into human moderators' practices (Schabus and Skowron, 2018).

Detecting comments that need moderators' attention is usually approached as a text classification task (see e.g. Pavlopoulos et al., 2017a); but comments can be blocked for a range of reasons (Shekhar et al., 2020). One is the presence of offensive language, a well-studied NLP task (see Section 2 below); however, others include advertising or spam, illegal content, spreading misinformation, trolling and incitement — all distinct categories which might be expected to show distinct features, and perhaps to vary according to the content being commented on. Another aspect that distinguishes the comment moderation task from the usual text classification tasks in NLP is the need for interpretable or explainable models: if classifiers are to be used by human moderators within publishers' working practices, they must be able to understand the outputs (Švec et al., 2018).

Here, we therefore investigate models which can provide both an aspect of interpretability and the ability to take account of the topics being discussed, by incorporating topic information into the comment classifier. Specifically, we incorporate semantic representations learned by the Embedded Topic Model (ETM) (Dieng et al., 2020) into a classifier pipeline based on Long Short-Term Memory (LSTM) networks (Hochreiter and Schmidhuber, 1997). Our model improves performance

[1] Some newspapers allow completely anonymous posting; some require commenters to create an account with a username, but this does not usually reveal their true identity.

[2] NYT Comment FAQ: https://nyti.ms/2PF02kj

Proceedings of Recent Advances in Natural Language Processing, pages 1652–1662
Sep 1–3, 2021.
https://doi.org/10.26615/978-954-452-072-4_185

by 4.4% over a text-only approach on the same dataset (Shekhar et al., 2020), and is more confident in the correct decisions it makes. Inspection of the topic distributions reveals how different newspaper sections have different language and topic distributions, including differences in the kind of comments that need moderation.[3]

2 Related Work

Automated news comment moderation Most research on this task so far formulates it as a text classification problem: for a given comment, the model must predict whether the comment violates the newspaper's policy. However, approaches to classification vary. Nobata et al. (2016) use a range of linguistic features, e.g. lexicon and n-grams. Pavlopoulos et al. (2017a) and Švec et al. (2018) use neural networks, specifically RNNs with an attention mechanism. Recently, Tan et al. (2020) and Tran et al. (2020) apply a modified BERT model (Devlin et al., 2019) while Schabus et al. (2017) use a bag-of-words approach.

Some approaches go beyond the comment text itself: Gao and Huang (2017) add information like user ID and article headline into their RNN to make the model context-aware; Pavlopoulos et al. (2017b) incorporate user embeddings; Schabus and Skowron (2018) incorporate the news category metadata of the article. However, no work so far investigates automatic modelling of topics (rather than relying on categorical metadata), or applies this to the comments rather than just their parent articles.

Some steps towards model intepretability and output explanation have also been taken: both Švec et al. (2018) and Pavlopoulos et al. (2017a) use an attention saliency map to highlight possibly problematic words. However, we are not aware of any work using higher-level topic information as a route to understanding model outputs.

Available datasets Several datasets have been created for the news comment moderation task. Nobata et al. (2016) provide 1.43M comments posted on Yahoo! Finance and News over 1.5 years, in which 7% of the comments are labelled as abusive via a community moderation process. Gao and Huang (2017) contains 1.5k comments from Fox News, annotated with specific hateful/non-hateful labels as a post-hoc task, and having 28% hateful comments. However, both are relatively small, and their labelling methods mean that neither dataset is entirely representative of the moderation process performed by newspapers.

Pavlopoulos et al. (2017a) provides 1.6M comments from Gazzetta, a Greek sports news portal, over c.1.5 years. Here, 34% of comments are labelled as blocked, and the labels are derived from the newspaper's human moderators and journalists. Schabus et al. (2017) and Schabus and Skowron (2018) provide a dataset from a German-language Austrian newspaper with 1M comments posted over 1 year, out of which 11,773 comments are annotated using seven different rules.

More recently, Shekhar et al. (2020) present a dataset from 24sata, Croatia's most widely read newspaper.[4] This dataset is significantly larger (10 years, c.20M comments); and moderator labels include not only a label for blocked comments, but also a record of the reason for the decision according to a 9-class moderation policy. However, their experiments show that classifier performance is limited, and transfers poorly across years. Here, we therefore use this dataset (see Section 3), with a view to improving performance and applying a topic-aware model to improve and better understand the robustness in the face of changing topics.

Related tasks More attention has been given to related tasks, most prominently the detection of offensive language, hate speech, and toxicity (Pelicon et al., 2021). A comprehensive survey of dataset collection is provided by Poletto et al. (2020) and Vidgen and Derczynski (2020).[5]

Topic Modelling Topic models capture the latent themes (also known as *topics*) from a collection of documents through the co-occurence statistics of the words used in a document. Latent Dirichlet Allocation (LDA) (Blei et al., 2003), a popular method for capturing these topics, is a generative document model where a document is a mixture of topics expressed as a probability distribution over the topics and a topic is a distribution over the words in a vocabulary. The Embedded Topic Model (ETM, Dieng et al., 2020) is an LDA-like topic modelling method that exploits the semantic information captured in word embeddings during topic inference. The advantage of ETM over LDA

[3] Source code available at https://github.com/ezosa/topic-aware-moderation

[4] http://24sata.hr/

[5] http://hatespeechdata.com/ provides a comprehensive list of relevant datasets.

Comment Moderation Data			
	Blocked	Non-blocked	Blocking Rate
Train	4984	75016	6.23%
Valid	642	9358	6.42%
Test	37271	438142	7.84%
Topic Modelling Data			
	Blocked	Non-blocked	Blocking Rate
Train	34863	36725	48.70%
Valid	4880	5120	48.80%

Table 1: Details of datasets used in experiments.

Section (– Subsection)	Blocked	Non-blocked	Blocking Rate
Kolumne (*Columns*)	655	6382	9.31%
Lifestyle	2426	30985	7.26%
Show	6827	58896	10.39%
Sport	5882	80820	6.78%
Tech	382	7173	5.06%
Vijesti (*News*)	20094	239835	7.73%
– Crna kronika (*Crime*)	5917	62471	8.65%
– Hrvatska (*Croatia*)	3527	45170	7.70%
– Politika (*Politics*)	6088	80264	7.05%
– Svijet (*World*)	2625	31459	7.24%

Table 2: Details per section, and (for section Vijesti) sub-section, of the comment moderation test set.

is that it combines the advantages of word embeddings with the document-level dependencies captured by topic modelling and has been shown to produce more coherent topics than regular LDA.

3 Dataset

We use the 24sata comment dataset (Shekhar et al., 2020; Pollak et al., 2021), introduced in Section 2. This contains c.21M comments on 476K articles from the years 2007-2019[6], written in Croatian. The dataset has details of comments blocked by the 24sata moderators, based on a set of moderation rules–these vary from hate speech to abuse to spam (see Shekhar et al., 2020, for rule description). The dataset also identifies the article under which a comment was posted, together with the section/subsection of the newspaper the article appeared in. These sections/sub-sections relate to the content of the article: for example, the Sport section contains sports-related news while the Kolumne (*Columns*) section contains opinion pieces. The largest section, Vijesti (*News*), is further subdivided as shown in Table 2.

3.1 Data Selection

In this work, we use data from 2018 for training and validation of the topic model and classifiers and data from 2019 for testing. This reflects the realistic scenario where we use data collected from the past to make predictions. For training and validation, we randomly select 50,000 articles out of 65,989 articles from 2018, sampling from the nine most-representative sections/sub-sections (Table 2). Each article comes with c.50 comments on average.

To train the topic model, we sample around 80,000 comments across these articles, with a roughly equal split between blocked and non-blocked comments. This is to encourage a diverse mix of topics from both comment classes. As a preprocessing step we remove comments with less than 10 words from the training data (see Table 1 (lower part)). To train the classifiers, we randomly sample around 80,000 comments such that the sampled set has the same blocking rate as the entire 2018 dataset.

For the test set, we then use all 475,413 comments associated with the 17,953 articles from 2019. Table 1 (upper part) provides the dataset details, with comment moderation blocking rate. For the test set, Table 2 provides details on the section and sub-section of the related articles. These top nine sections account for more than 95% of the comments of the entire test set.

3.2 Content Analysis

To gain some insight into the content of blocked comments, we analyze the linguistic differences between blocked and non-blocked comments and across different sections. First, we compare comment length. As we can see from Table 3, blocked and non-blocked comments have, on average, similar lengths. However, if we further divide blocked comments into two sub-groups — spam and non-spam — we find that on average, spam comments are longer than other comments. We observe a similar pattern across different sections.

Next, we measure lexical diversity using mean-segmental type-token ratio (MSTTR). The MSTTR is computed as the mean of type-token ratio for every 1000 tokens in a dataset to control for dataset size (van Miltenburg et al., 2018). From Table 3, we see that non-blocked comments have higher MSTTR (i.e. higher lexical diversity) than blocked comments (0.62 vs 0.46). However, when we again divide blocked comments into spam and non-spam,

[6]Dataset is available at http://hdl.handle.net/11356/1399

we observe that non-spam blocked comments have a similar MSTTR to non-blocked comments (0.61 vs 0.62), while spam comments have much lower MSTTR (0.35 vs 0.61). This suggests that blocked comments (excluding spam) have as rich a vocabulary as non-blocked. Again, we see a similar pattern across different news sections.

	Mean length	MSTTR
All	23.06	0.61
Non-blocked	23.01	0.62
Blocked	23.65	0.46
Blocked (non-spam)	19.16	0.61
Blocked (Spam only)	28.23	0.35

Table 3: Mean-segmental TTR and average length of comments

Now we look at the top bigrams of each class. We collect all bigrams that occur at least 50 times and rank them according to their pointwise mutual information (PMI) score. In general, we do not see many overlaps between the top bigrams of blocked and non-blocked comments across the different sections. Bigrams in blocked comments indicate spam messages such 'iskustva potrebnog' (*experience required*), 'redoviti student' (*full-time student*) and 'prilika pružila' (*opportunity given*). Removing spam comments, we encounter bigrams used for swearing such as 'pas mater' (*damn it*) and 'jedi govna' (*eat sh*t*). In the non-blocked comments, the top bigrams are more relevant to the section they appear in. For instance, in the Vijesti section, top bigrams include 'new york', 'porezni obveznici' (*taxpayers*) and 'naftna polja' (*oil fields*) while in Sports, top bigrams include 'all star', 'grand slam' and 'man utd'.

This suggests that the content of blocked comments tends to share commonalities across sections more than non-blocked comments; but again, these commonalities may be mostly within the spam category, with other blocked categories being more topic-dependent. Our next step therefore is to examine the use of topic modelling to capture these dependencies, with a view to using topic information to improve a moderation classifier.

4 Topic Modelling

We now apply a topic model to gain insight into what characterises a blocked comment and a non-blocked one, and whether this varies between different sections where different subjects are discussed.

4.1 Topic Model

We use the Embedded Topic Model (ETM, Dieng et al., 2020) as our topic model since it has been shown to outperform regular LDA and and other neural topic modelling methods such as NVDM (Miao et al., 2016). We also want to take advantage of ETM's ability to incorporate the information encoded in pretrained word embeddings trained on vast amounts of data to produce more coherent topics. In the ETM, the topic-term distribution for topic k, β_k, is induced by a matrix of word embeddings ρ and its respective topic embedding α_k which is a point in the word embedding space:

$$\beta_k = softmax(\rho^T \alpha_k) \quad (1)$$

The topic embeddings are learned during topic inference while the word embeddings can be pretrained or also learned during topic inference. In this work, we use pretrained embeddings.

The document-topic distribution of a document d, θ_d, is drawn from the logistic normal distribution whose mean and variance come from an inference network:

$$\theta_d \sim LN(\mu_d, \sigma_d) \quad (2)$$

Given a trained ETM, we can infer the **document-topic distribution (DTD)** of an unseen document. In addition, we can also compute a **document-topic embedding (DTE)** as the weighted sum of the embeddings of the topics in a document, where the weight corresponds to the probability of the topic in that document:

$$DTE = \sum_{k=0}^{K} \alpha_k \theta_{d,k} \quad (3)$$

where α_k is the topic embedding of topic k, and $\theta_{d,k}$ is the probability of topic k in doc d.

4.2 Topic Analysis

Now we analyse the usage of topics in our test set. We trained the ETM for 100 topics on the training set and inferred the topic distributions of the comments in the test set. For analysis, we extract the top topics in a set of comments. To do this, we take the mean of the topic distributions over the comments in the set and rank the topics according to their weight in this mean distribution. We then take the top 15 topics for analysis because this is the average number of topics in a comment with a non-zero probability in our test set. Note that in this analysis we only use the document-topic

distributions and not the document-topic embeddings. To more easily discuss the topics here we provide concise labels for each topic as interpreted by a native speaker. Automatic labelling of topics is a non-trivial task and an area of active research (Bhatia et al., 2016; Alokaili et al., 2020; Popa and Rebedea, 2021).

First, we examine the prevalent topics in the blocked and non-blocked comments, separately. The top topics of non-blocked comments cover a diverse range of subjects from politics to football while the top topics in blocked comments are dominated by spam and offensive language (Figure 1). However, we also see many topics shared between blocked and non-blocked comments. [7].

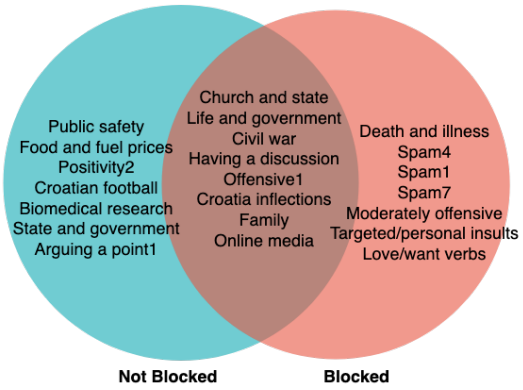


Figure 1: Top topics of the blocked and non-blocked comments for the entire test set.

Next we illustrate how different topics intersect and diverge between blocked and non-blocked comments across sections by looking at the top topics of two thematically-different sections, Lifestyle and Politika (*Politics*).

Figure 2 shows the top topics of these sections and the intersections between them. In Politics, blocked comments tend toward spam and targeted insults. Non-blocked topics include public safety and finances. However, we also see that more than half of the top topics overlap between blocked and non-blocked. This suggests that, thematically, there isn't a very clear distinction between blocked and non-blocked comments in the Politics section.

In Lifestyle, blocked topics are dominated by spam and while there are topics on offensive insults, they are not as prevalent as the spam-related ones. The non-blocked topics are about family and relationships and commenters arguing with each other. Compared to Politics, we see a clearer distinction between topics in blocked and non-blocked in this section. In terms of topic overlaps between Lifestyle and Politics, blocked comments in both sections are dedicated to spam and insults while non-blocked comments focus on positive sentiments.

The combination of certain topics also provide an indication of the classification of the comment. For instance, we notice the use of topics about football cards in comments that do not do not discuss the sport (for instance, football cards as a topic is prominent in the blocked Lifestyle comments). It turns out that some commenters use the red and yellow cards from football as metaphors for being banned or having their comments blocked by moderators (12% of comments that use these metaphors are blocked by moderators). On the other hand, comments that use the football cards topics *and* any of the sports-related topics are likely to be a genuine discussion of football (only 5% of such comments are blocked by moderators). We show some examples of these comments in Table 5.

So clearly there is a distinction between the usage of topics in the non-blocked and blocked comments. We therefore think it is a good idea to propose a model which incorporates topic information into a comment moderation classifier.

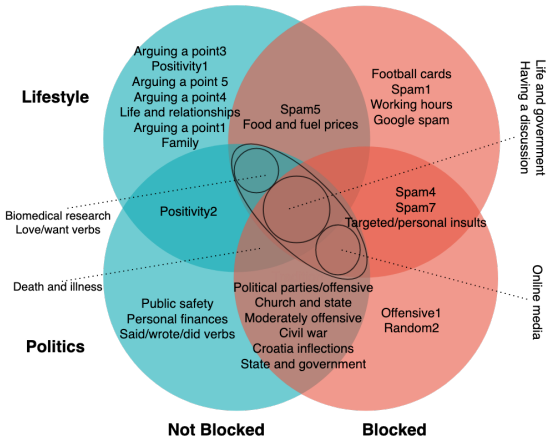


Figure 2: Top topics of the blocked and non-blocked comments in the Lifestyle and Politics sections.

5 Topic-aware Classifier

Our aim is to improve comment moderation predictions by combining textual features with document-level semantic information in the form of topics. To this end, we test several model architectures that combine a language model with topic features.

For the comment text representation, we use a

[7] All 100 topics and labels are available at https://github.com/ezosa/topic-aware-moderation

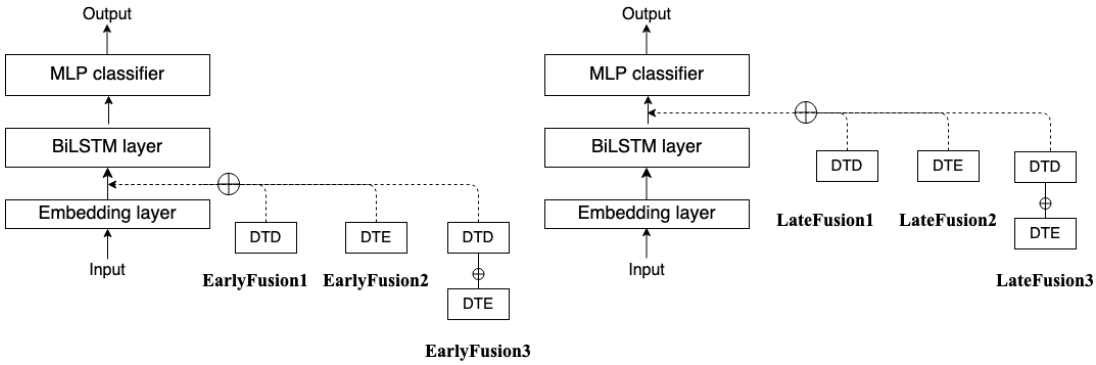


Figure 3: Architectures combining text and topic features. DTD is the topic distribution of a document while DTE is the topic embedding.

bidirectional LSTM (BiLSTM, Schuster and Paliwal, 1997). The comment text is given as input to an embedding layer then a BiLSTM layer where the output of the final hidden state is taken as the encoded representation of the comment. For the topic representations, we use the topic distributions (DTD) and topic embeddings (DTE) discussed in Section 4.1.

We propose two fusion mechanisms to combine the text and topic representations: *early* and *late* fusion. In early fusion, topic features are concatenated with the output of the embedding layer and then passed to the BiLSTM layer. In **EarlyFusion1 (EF1)**, only DTD is concatenated with the word embeddings; **EarlyFusion2 (EF2)** uses DTE instead of DTD; and **EarlyFusion3 (EF3)** uses both DTE and DTD. In late fusion, topic features are concatenated with the output representation of the BiLSTM layer, and passed to the MLP for classification. Again, **LateFusion1 (LF1)** uses DTD; **LateFusion2 (LF2)** uses DTE; and **LateFusion3 (LF3)** uses both. Figure 3 shows the architectures.

Our model is inspired by the Topic Compositional Neural Language Model (TCNLM, Wang et al., 2018) and the Neural Composite Language Model (NCLM, Chaudhary et al., 2020) that incorporate latent document-topic distributions with language models. Both of these models simultaneously learn a topic model and a language model through a joint training approach. The NCLM introduced the use of word embeddings to generate an explanatory topic representation for a document in addition to the document-topic proportions. In our work, instead of using the word embeddings of the top words of the latent topics of a document (where the number of top words is a hyperparameter), we leverage the topic embeddings learned by ETM and combine them with the document-topic proportions to produce the document-topic embeddings (DTE). Also unlike the TCNLM and NCLM, we use pre-trained topics in our model so as to easily de-couple and analyse the influence of topics in the classifier performance. Another related work is TopicRNN (Dieng et al., 2016), a model that uses topic proportions to re-score the words generated by the language model. The topics generated by this model, however, have been shown to have lower coherences compared to NCLM (Chaudhary et al., 2020).

6 Experimental Setup

Dataset As discussed in Section 3.1, we use the 2018 data as the training and validation sets of our topic-aware classifier and the 2019 data as the test set. Details of the train and validation sets are shown in Table 1 and the test set in Table 2.

Baseline models To assess how topic information improves comment classification, we use as baselines the following models trained only on text *or* topics:

- **Text only**: a classifier with BiLSTM & MLP layers, similar to Figure 3 but with comment text alone as input.
- **Document-topic distribution (DTD)**: MLP only, document-topic distributions as input.
- **Document-topic embedding (DTE)**: MLP only, document-topic embeddings as input.
- **DTD+E**: MLP only, concatenated document-topic distributions and embeddings.

Hyperparameters We use 300D word2vec embeddings, pretrained on the Croatian Web Corpus (HrWAC, Ljubešić and Erjavec, 2011; Šnajder, 2014), for training the ETM and to initialize the embedding layer of the BiLSTM. The ETM is trained

for 500 epochs for 100 topics using the default hyperparameters from the original implementation [8]. The BiLSTM is composed of one hidden layer of size 128 with dropout set to 0.5. The MLP classifier is composed of one fully-connected layer, one hidden layer of size 64, a ReLU activation, and a sigmoid for classification with the classification threshold set to 0.5. We use Adam optimizer with $lr = 0.005$. We train all classifiers for 20 epochs with early stopping based on the validation loss.

7 Results

In Table 4, we present the performance of the baselines and proposed models, measured as macro F1-scores. All models that combine text and topic representations perform better than the models that use only text *or* topics. Of the baseline models, the DTD model performs comparatively better than the DTE and DTD+E models, and surprisingly performs almost as well as the Text-only model; however, we show in Section 8 below that DTD is much less confident in its predictions than the Text-only model. Overall, the best performing model is LF1, which improves the Text-only model's performance by $+4.4\%$ (67.37% vs 62.97%); and improves by a similar amount over Shekhar et al.'s results using mBERT (macro-F1 score 62.07 for year 2019).

Interestingly, we see a wide variation in performance across news sections. We observe that comments in Lifestyle and Tech are the easiest to classify (best F1 over 72.00) while Politika (*Politics*) is the most difficult (best F1 around 61.61). The main cause appears to be that Lifestyle and Tech have the highest proportion of spam comments: on average, 49.44% of blocked comments in the test set are spam, but for Lifestyle and Tech this number rises to 77.25% and 69.63%, respectively. As for the Politics section, the most likely reason the comments are difficult to classify is that, excluding spam, there is a high degree of overlap in the subjects discussed in the blocked and non-blocked comments (see the topic analysis in Section 4.2).

7.1 Analysis of Classifier Outputs

In general, we observe that blocked comments tend to use similar topics across different sections while non-blocked comments have more diverse topics. Of the nine sections that we analyzed, there are five topics that are prominent in blocked comments in all sections ('Targeted/personal insults', 'Spam4', 'Spam7', 'Online media', and, 'Having a discussion') and only three topics prominent in non-blocked comments ('Having a discussion', 'Online media', and, 'Life and government'). This suggests that blocked comments are more semantically-coherent across sections than non-blocked ones. In contrast, topics in non-blocked comments tend to be more relevant to their respective sections: for instance, family and relationships are not discussed a lot in the Politics section, while Lifestyle commenters do not tend to talk about political issues.

The higher topical coherence then of blocked comments explains why a text classification approach can achieve reasonable performance; but the variation in blocked comment content between some sections explains why adding topic information improves our classification results.

Next, we analyze the confidence of classifiers and examine some of the outputs of the models. To analyze confidence, we gradually increase the classification threshold from 0.5 to 1.0 in increments of 0.05. For every new threshold, we plot the macro-F1 for the different models (Figure 4). We compare the confidence of four models: DTD, Text-only, EF2 (the strongest early fusion model), and LF1 (the overall best-performing model). We find that the most confident model is LF1 and the least confident is DTD. The two fusion classifiers display similar levels of confidence. The Text-only classifier is not as confident as the fusion classifiers but still more confident than DTD. This suggests that adding topic features to text not only improves performance, it also increases classifier confidence.

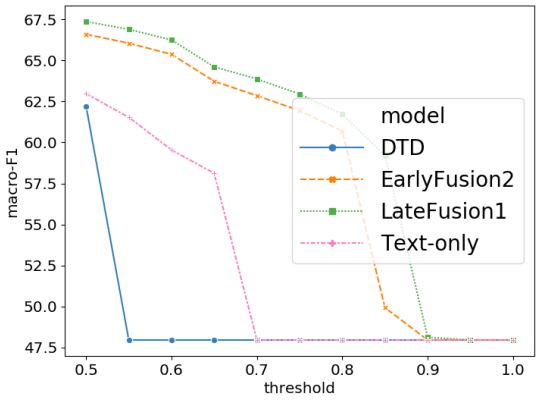


Figure 4: Confidence of the top performing models.

In Table 5 we give some examples of comments and the classifier decisions of the Text-only classifier and LF1 (our best-performing fusion model) and their top topics (topics with $prob > 0.10$). The

[8] https://github.com/adjidieng/ETM

Section – Subsection	Text only	Topics only			Text+Topic Combinations					
		DTD	DTE	DTD+E	EF1	EF2	EF3	LF1	LF2	LF3
All	62.97	62.20	59.3	58.33	66.33	66.58	65.61	**67.37**	66.22	66.95
Kolumne	59.86	59.65	56.25	55.33	62.40	62.90	63.13	63.25	62.38	**63.6**
Lifestyle	69.21	70.07	65.93	64.47	72.73	70.9	69.36	72.00	72.39	**72.92**
Show	61.97	61.30	58.62	57.60	65.24	65.63	64.26	**66.50**	65.00	65.86
Sport	63.22	61.42	58.61	57.90	67.11	67.86	66.74	**68.26**	67.14	67.82
Tech	64.87	66.37	63.17	62.55	67.72	68.74	67.65	68.76	67.68	**69.15**
Vijesti (*News*)	62.38	61.49	58.79	57.77	65.58	65.99	65.24	**66.77**	65.53	66.24
– Crna kronika	64.67	63.98	61.03	59.84	68.10	68.88	68.11	**69.60**	67.89	68.88
– Hrvatska	63.61	63.50	60.10	58.93	67.24	66.86	65.95	67.90	67.12	**67.95**
– Politika	57.93	56.49	54.95	54.20	60.51	61.52	60.84	**61.61**	60.63	61.30
– Svijet	63.58	62.55	59.62	58.35	66.83	66.95	66.33	68.44	67.21	67.57

Table 4: Classifier performance measured as macro-F1.

Comment	Label	Text-only	LF1	Top topics
1. konačno. gamad lopovska crno bijela prevarantska (*finally. the black and white cheating thieving bastards*)	1	1 (0.501)	1 (0.687)	Arguing a point, Political parties (offensive)
2. ...dobro jutro,moze crveni karton za novinara koji je osmislio naslov ;-) (*... good morning, how about a red card for the journalist who came up with this title ;-)*)	1	0 (0.315)	0 (0.456)	Football cards
3. Ne bum komentiral, dosta mi je kazni od žutih i crvenih kartona. Strah me je cenzure i bradate cure. (*No comment, I'm tired of getting yellow and red cards. I'm afraid of censorship and bearded ladies.*)	0	0 (0.054)	0 (0.335)	Football cards, Random
4. Koji kurac Rumunjski sudac ne da koji karton više Čehima. Pa svake tri minute sa leđa sruše Olma !!!! *(Why the fuck does the Romanian referee not give a few cards more to the Czechs, They tackle Olm from behind every three minutes.)*	0	0 (0.303)	1 (0.587)	Targeted/personal insults
5. Baš ste jadnici kao i ovi sa 24sata koji u ovome uživaju ! (*All of you are lame as well as those from 24sata who enjoy this.*)	1	0 (0.171)	0 (0.229)	Online media, Moderately offensive
6. Google sada plaća između 15.000 i 30.000 dolara mjesečno za rad na mreži od kuće. Pridružio sam se ovom poslu prije 3 mjeseca i zaradio 24857 dolara u prvom mjesecu ovog posla. >>> URL (*Google now pays between 15.000 and 30.000 dollars per month for working remotely from home. I started this job 3 months ago and made 24857 dollars in the first month of this job. >>> URL*)	0	1 (0.67)	1 (0.90)	Spam4

Table 5: Sample comments and classifier decisions.

first example contains swearing which both models pick up on and classify as blocked although LF1 is more confident in its decision then Text-only. In the second example, both models predict the wrong label but LF1 treats this as a borderline case because it is targeted at the moderators. However since this is only a mild provocation of the moderators, this might be a case where the gold label is incorrect. The topics also pick up on the fact that this comment talks about football cards but only has a tenuous connection to the sport ("getting a red card" is an expression used for "being banned"). In contrast, the third comment also uses the banning sense of "card" but is not directed at anyone, and is thus labeled as 0 (non-blocked), which both models get right. Again the topics indicate that the comment is not really about the sport. The fourth example shows a case where "cards" are mentioned in their standard football sense but also contains a swear word, making the gold label of 0 (non-blocked) questionable. The better performance of LF1 on such examples, compared to Text-only, implies that

LF1 is better aware of the different semantics of "card" (sports-related vs. metaphorical), likely due to added topic information.

The fifth example contains a moderately offensive insult that is not directed at any single group except the 24sata readership in general. One reason why both classifiers do not get this right is that the word *jadnici* is not strong enough to be considered offensive. Finally the last example is clearly a spam comment that both classifiers correctly classify but for which the gold label is incorrect.

Overall, compared to the Text-only model, we find that LF1 more often than not improves the confidences (and sometimes the classification), especially in cases in which the gold label is clear. This is valuable in practice, as better confidences might lead to better prioritisation of comments for manual moderation, reducing the time required to remove the most problematic ones.

8 Conclusion

In this work, we propose a model to combine document-level semantics in the form of topics with text for comment moderation. Our analysis shows that blocked and non-blocked comments have different linguistic and thematic features, and that topics and language use vary considerably across news sections, including some variation in the comments that should be blocked. We also found that blocked comments tend to be more semantically coherent across sections than non-blocked ones. We therefore see that the use of topics in our model improves performance, and gives more confident outputs, over a model that only uses the comment text. The model also provides topic distributions, interpretable as keywords, as a form of an explanation of its prediction. As future work, we plan to incorporate comment, article, and user metadata into the model.

Acknowledgements

This work has been supported by the European Union Horizon 2020 research and innovation programme under grants 770299 (NewsEye) and 825153 (EMBEDDIA), and by the UK EPSRC under grant EP/S033564/1.

Ethics and Impact Statement

Data The dataset and annotations are provided by the publisher of 24sata.hr, Styria Media Group, for research purposes and deposited in the CLARIN repository. The authors of the comments are anonymised. The researchers used the data as-is and did not modify or add annotations.

Intended Use The models we present here are intended to assist comment moderators in their work. We do not recommend that the model be deployed in the moderation process without a human-in-the-loop.

Potential Misuse The models and the analysis of their performance we provide in this paper could be used by malicious actors to gain an insight into the comment moderation process and find loopholes in the process. However, we think such a risk is unlikely and the impact it might have outweighs the potential benefits of models intended to assist human moderators such as the ones we present here.

References

Areej Alokaili, Nikolaos Aletras, and Mark Stevenson. 2020. Automatic generation of topic labels. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1965–1968.

Shraey Bhatia, Jey Han Lau, and Timothy Baldwin. 2016. Automatic labelling of topics with neural embeddings. *arXiv preprint arXiv:1612.05340*.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

Yatin Chaudhary, Hinrich Schütze, and Pankaj Gupta. 2020. Explainable and discourse topic-aware neural language understanding. In *International Conference on Machine Learning*, pages 1479–1488. PMLR.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Adji B. Dieng, Francisco J. R. Ruiz, and David M. Blei. 2020. Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics*, 8:439–453.

Adji B Dieng, Chong Wang, Jianfeng Gao, and John Paisley. 2016. Topicrnn: A recurrent neural network with long-range semantic dependency. *arXiv preprint arXiv:1611.01702*.

Lei Gao and Ruihong Huang. 2017. Detecting online hate speech using context aware models. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 260–266, Varna, Bulgaria. INCOMA Ltd.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Nikola Ljubešić and Tomaž Erjavec. 2011. hrWaC and slWaC: Compiling web corpora for Croatian and Slovene. In *International Conference on Text, Speech and Dialogue*, pages 395–402. Springer.

Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural variational inference for text processing. In *International conference on machine learning*, pages 1727–1736. PMLR.

Emiel van Miltenburg, Ruud Koolen, and Emiel Krahmer. 2018. Varying image description tasks: spoken versus written descriptions. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 88–100.

Chikashi Nobata, J. Tetreault, A. Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive language detection in online user content. *Proceedings of the 25th International Conference on World Wide Web*.

John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. 2017a. Deeper attention to abusive user content moderation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1125–1135, Copenhagen, Denmark. Association for Computational Linguistics.

John Pavlopoulos, Prodromos Malakasiotis, Juli Bakagianni, and Ion Androutsopoulos. 2017b. Improved abusive comment moderation with user embeddings. In *Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism*, pages 51–55, Copenhagen, Denmark. Association for Computational Linguistics.

Andraž Pelicon, Ravi Shekhar, Blaž Škrlj, Matthew Purver, and Senja Pollak. 2021. Investigating cross-lingual training for offensive language detection. *PeerJ Computer Science*, 7:e559.

Fabio Poletto, Valerio Basile, Manuela Sanguinetti, Cristina Bosco, and Viviana Patti. 2020. Resources and benchmark corpora for hate speech detection: a systematic review. *Language Resources and Evaluation*, pages 1–47.

Senja Pollak, Marko Robnik-Šikonja, Matthew Purver, Michele Boggia, Ravi Shekhar, Marko Pranjić, Salla Salmela, Ivar Krustok, Tarmo Paju, Carl-Gustav Linden, Leo Leppänen, Elaine Zosa, Matej Ulčar, Linda Freienthal, Silver Traat, Luis Adrián Cabrera-Diego, Matej Martinc, Nada Lavrač, Blaž Škrlj, Martin Žnidaršič, Andraž Pelicon, Boshko Koloski, Vid Podpečan, Janez Kranjc, Shane Sheehan, Emanuela Boros, Jose G. Moreno, Antoine Doucet, and Hannu Toivonen. 2021. EMBEDDIA tools, datasets and challenges: Resources and hackathon contributions. In *Proceedings of the EACL Hackashop on News Media Content Analysis and Automated Report Generation*, pages 99–109, Online. Association for Computational Linguistics.

Cristian Popa and Traian Rebedea. 2021. Bart-tl: Weakly-supervised topic label generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1418–1425.

Dietmar Schabus and Marcin Skowron. 2018. Academic-industrial perspective on the development and deployment of a moderation system for a newspaper website. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Dietmar Schabus, Marcin Skowron, and Martin Trapp. 2017. One million posts: A data set of german online discussions. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1241–1244.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681.

Ravi Shekhar, Marko Pranjić, Senja Pollak, Andraž Pelicon, and Matthew Purver. 2020. Automating News Comment Moderation with Limited Resources: Benchmarking in Croatian and Estonian. *Journal for Language Technology and Computational Linguistics (JLCL)*, 34(1).

Jan Šnajder. 2014. DerivBase.hr: A high-coverage derivational morphology resource for Croatian. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3371–3377, Reykjavik, Iceland. European Language Resources Association (ELRA).

Andrej Švec, Matúš Pikuliak, Marián Šimko, and Mária Bieliková. 2018. Improving moderation of online discussions via interpretable neural models. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 60–65, Brussels, Belgium. Association for Computational Linguistics.

Fei Tan, Yifan Hu, Changwei Hu, Keqian Li, and Kevin Yen. 2020. TNT: Text normalization based pre-training of transformers for content moderation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4735–4741, Online. Association for Computational Linguistics.

Thanh Tran, Yifan Hu, Changwei Hu, Kevin Yen, Fei Tan, Kyumin Lee, and Se Rim Park. 2020. HABERTOR: An efficient and effective deep hatespeech detector. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7486–7502, Online. Association for Computational Linguistics.

Bertie Vidgen and Leon Derczynski. 2020. Directions in abusive language training data, a systematic review: Garbage in, garbage out. *Plos one*, 15(12):e0243300.

Wenlin Wang, Zhe Gan, Wenqi Wang, Dinghan Shen, Jiaji Huang, Wei Ping, Sanjeev Satheesh, and Lawrence Carin. 2018. Topic compositional neural language model. In *International Conference on Artificial Intelligence and Statistics*, pages 356–365. PMLR.

C. Manuscript: Capturing Evolution in Word Usage: Just Add More Clusters?

Capturing Evolution in Word Usage: Just Add More Clusters?

Matej Martinc*
matej.martinc@ijs.si
Jozef Stefan Institute
Slovenia

Syrielle Montariol*
syrielle.montariol@limsi.fr
LIMSI - CNRS, Univ. Paris-Sud, Univ. Paris-Saclay, Societé Générale
France

Elaine Zosa*
elaine.zosa@helsinki.fi
University of Helsinki
Finland

Lidia Pivovarova
lidia.pivovarova@helsinki.fi
University of Helsinki
Finland

ABSTRACT

The way the words are used evolves through time, mirroring cultural or technological evolution of society. Semantic change detection is the task of detecting and analysing word evolution in textual data, even in short periods of time. In this paper we focus on a new set of methods relying on contextualised embeddings, a type of semantic modelling that revolutionised the NLP field recently. We leverage the ability of the transformer-based BERT model to generate contextualised embeddings capable of detecting semantic change of words across time. Several approaches are compared in a common setting in order to establish strengths and weaknesses for each of them. We also propose several ideas for improvements, managing to drastically improve the performance of existing approaches.

CCS CONCEPTS

• **Computing methodologies** → **Lexical semantics**; *Cluster analysis*; • **Information systems** → *Language models.*

KEYWORDS

Semantic Change, Contextualised Embeddings, Clustering

ACM Reference Format:
Matej Martinc, Syrielle Montariol, Elaine Zosa, and Lidia Pivovarova. 2020. Capturing Evolution in Word Usage: Just Add More Clusters?. In *Companion Proceedings of the Web Conference 2020 (WWW '20 Companion), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3366424.3382186

1 INTRODUCTION

The large majority of data on the Web is unstructured. Amongst it, textual data is an invaluable asset for data analysts. With the large increase in volume of interaction and overall usage of the Web, more and more content is digitised and made available online, leading to a huge amount of textual data from many time periods becoming accessible. However, textual data are not necessarily homogeneous as they rely on a crucial element that evolves throughout time: language. Indeed, a language can be considered as a dynamic system where word usages evolve over time, mirroring cultural or technological evolution of society [1].

In linguistics, *diachrony* refers to the study of temporal variations in the use and meaning of a word. While analysing textual data from the Web, detecting and understanding these changes can be done for two primary goals. First, it can be used directly for linguistic research or social analysis, by interpreting the reason of the semantic change and linking it to real-world events, and by analysing trends, topics and opinions evolution [9]. Second, it can be used as a support for many tasks in Natural Language Processing (NLP), from text classification to information retrieval conducted on a temporal corpora where semantic change might occur.

To tackle semantic change, models usually rely on word embeddings, which summarise all senses and usages of a word within a certain time period into one vector. Measuring the distance between these vectors across time periods is used to detect and quantify the differences in meaning. But these methods do not take into consideration that most words have multiple senses, since all word usages are aggregated into a single static word embedding. Contextualised embedding models such as BERT [5] are capable of generating a separate vector representation for each specific word usage, making them more suitable for this task.

The goal of this paper is to establish the best way to detect semantic change in a temporal corpus by capitalising on BERT contextualised embeddings. First, several approaches for semantic shift detection from the literature are compared in a common setting in order to establish strengths and weaknesses of each specific method. Second, several improvements are presented, which manage to drastically improve the performance of existing approaches. Our code and models are publicly available[1].

2 RELATED WORK

A large majority of methods for semantic shift detection leverage dense word representations, i.e. embeddings. Word-frequency methods for detecting semantic shift that were popular in earlier studies [13, 16], are now rarely used. The detailed overview of the field could be found in recent surveys [22, 27, 28].

*The authors contributed equally to this research.

WWW '20 Companion, April 20–24, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7024-0/20/04.
https://doi.org/10.1145/3366424.3382186

[1] https://github.com/smontariol/AddMoreClusters

2.1 Static Word Embeddings for Semantic Change

The first research that employed word embeddings for semantic shift detection was conducted by [18]. The main idea was to train a separate embedding model for each time period. Since embedding algorithms are inherently stochastic and the resulting embedding sets are invariant under rotation, a procedure that makes these models comparable is needed. To solve this problem, they proposed the incremental model fine-tuning approach, where the weights of the model, trained on a certain time period, are used to initialize weights of a model trained on the next successive time period. Some improvements of the approach were later proposed by [24], who replaced the softmax function for the continuous skipgram model with a more efficient hierarchical softmax, and by [17], who proposed an incremental extension for negative sampling.

An alternative approach was proposed in [19], where embedding models trained on different time periods were aligned in a common vector space after the initial training using a linear transformation for the alignment. The approach was upgraded [31] by using a set of nearest neighbour words as anchors for the alignment.

The third alternative for semantic shift detection with static word embeddings is to treat the same words in different time periods as different tokens in order to get time specific word representations for each time period [6, 26]. Here, only one embedding model needs to be trained and no aligning is needed.

2.2 The Emergence of Contextualised Embeddings

While in static word embedding models each word from the predefined vocabulary is presented as a unique vector, in contextualised embeddings a separate vector is generated for each word mention, i.e. for each context the word appears in. The two most widely used contextual embeddings models are ELMo (Embeddings from LanguageModels [25]) and a more recent BERT (Bidirectional Encoder Representations from Transformers [5]). The approach of using contextual embeddings for semantic shift detection is fairly novel; we are aware of three recent studies that employed it.

In the first study, contextualised embeddings were applied in a controlled way [15]: for a set of polysemic words, a representation for each sense is learned using BERT. Then pretrained BERT is applied to a diachronic corpus, extracting token embeddings, that are matched to the closest sense embedding. Finally, the proportions for each sense are computed at each successive time slice, revealing the evolution of the distribution of senses for each target word. This method requires that the set of senses of each target word is known beforehand.

Another possibility is clustering all contextual embeddings for a target word into clusters representing the word senses or usages in a specific time periods [10]. K-means clustering and BERT contextual embeddings were used in this study. In addition, the incremental training approach proposed by [18] was used for diachronic fine-tuning of the model. Jensen-Shannon divergence (JSD), a measure of similarity between probability distributions, was used to quantify changes between word usages in different time periods. They also tested if domain adaptation of the model would improve the results of their approach by fine-tuning the model on an entire corpus rather than on specific time periods, however this yielded no performance improvements.

In the third, even more recent study, contextual embeddings for a specific word in a specific time period were averaged in order to generate a time specific word representation for each word in each period [23]. BERT embeddings are used in the study and cosine distance is used for measuring the difference between word representations in different time periods.

3 DATA

We rely on a small human-annotated dataset [12] to conduct the evaluation. The dataset consists of 100 words from various frequency ranges, labelled by five annotators according to the level of semantic change between the 1960s and the 1990s. They use a 4-points scale from "0: no change" to "3: significant change", and the inter-rater agreement was 0.51 ($p < 0.01$, average of pair-wise Pearson correlations). The most significantly changed words from the dataset are, for example, *user* and *domain*; words for which the meaning remain intact, are for example *justice* and *chemistry*. This dataset is a valuable resource and has been used to evaluate methods for measuring semantic change in previous research [7, 10]. Following previous work, we use the average of the human annotations as semantic change score. For evaluation, we compute Pearson and Spearman rank correlations between this score and a model output. The notion of the best model is based on Spearman correlations.

To train the models we use the Corpus of Historical American English (COHA) [2]. It contains more than 400 million words of text from the 1810s-2000s. As a historical corpus, it is smaller than the widely used Google books corpus [3] but it has the advantage that data from each decade are balanced by genre—fiction, magazines, newspapers, and non-fiction texts, gathered from various Web sources. We focus our experiments on the most recent data in this corpus, from the 1960s to the 1990s (1960s has around 2.8 million and 1990s 3.3 million words), to match the manually annotated data. The fine-tuning of the model is also done only on this subset.

4 METHODOLOGY

4.1 Context-dependent Embeddings

BERT is a neural model based on the transformer architecture [29]. It relies on a transfer learning approach proposed by [14], where in the first step the network is pretrained as a language model on large corpora in order to learn general contextual word representations. This is usually followed by a task specific fine-tuning step e.g., classification or, in our case, domain adaptation. BERT's novelty is an introduction of a new pretraining learning objective, a *masked language model*, where a percentage of words from the input sequence is masked in advance, and the objective is to predict these masked words from an unmasked context. This allows BERT to leverage both left and right context, meaning that a word w_t in a sequence is not determined just from its left sequence $w_{1:t-1} = [w_1, ..., w_{t-1}]$—as is the case in the traditional language modelling task—but also from its right word sequence $w_{t+1:n} = [w_{t+1}, ..., w_{t+n}]$.

[2] https://www.english-corpora.org/coha/
[3] http://googlebooks.byu.edu/

In our experiments we use the English BERT-base-uncased model with 12 attention layers and a hidden layer of size 768, which was pretrained on the Google Books Corpus [11] (800M words) and Wikipedia (2,500M words). For some of the experiments (see Table 1), we further fine-tune this model (as a *masked language model*) for up to 10 epochs on the COHA subcorpus described in Section 3 for domain adaptation.

Note that our fine-tuning approach deviates from the approaches presented in some of the related work [10] and we do not conduct any diachronic fine-tuning of the model using the incremental training approach similar to [18]. The hypothesis is that this step is not necessary due to contextual nature of embeddings generated by the model, which by definition are dependent on the context that is always time-specific.

Since we are using a pre-trained model we have to apply the BERT tokenization, which is based on byte-pair encodings [30]. In order to acquire contextual embeddings, the corpus documents are first split into sentences; each sentence is limited to 512 tokens and fed into the BERT model. A sequence embedding is generated for each of these sequences by summing last four encoder output layers of BERT[4]. Finally, this sequence embedding of size $sequence\ length \times embeddings\ size$ is cut into pieces, to get a separate contextual embedding for each token in the sequence.

4.2 Target Words Selection

In any practical application of semantic change detection, performing clustering for every word in the corpus would not be feasible in terms of computing time. Thus, we investigate several scalable metrics as a preliminary step to identify a set of words that may have undergone semantic change.

A first set of metrics relies on the computation of a *variation* measure, similarly to [20]. Variation is the cosine distance between each token embedding and a *centroid*, i.e. an average token embedding for a given word. The mean of these cosine distances is the *variation coefficient* of a word. The intuition is that for words that have many different senses and usages, the distance to the centroid would be higher than for words that are monosemous. However, this method does not make distinction between words that gain (loose) sense and polysemous words that stay stable across time.

To measure an evolution of word variation, we compute the variation coefficient inside each time slice t. Then, we take the average difference from one time step to another. This measure aims at detecting words that undergo changes in their level of polysemy. For example, in a corpus divided into T time slices:

$$Variation\ by\ time\ slice = \frac{\sum_{t=t_0}^{T} |Variation_t - Variation_{t-1}|}{T},$$

The second set of metrics relies on *averaging* all token embeddings at each time slice, and using the cosine distance as a measure of semantic drift between time slices. The total drift is the cosine distance between the average of token representations of the first time slice and of the last time slice. It represents the amount of change a word has undergone from the first to the last period, without taking into account the variations in between. The *averaging by time slice* computes the mean of the drifts from each time step to the next one, in order to measure the successive changes of word usage.

To evaluate and compare these measures we use all hundred words from the test set. In practice it is possible to choose a threshold (as a fraction of the size of the full vocabulary) to get a list of target words. Then, the heavier clustering techniques can be applied to this list.

4.3 Embeddings Clustering

The goal of the clustering step is to group the word occurrences by similar vector representation. Then JSD is used to compare cluster distribution across time periods, same as in [10]. The intuition is the following: if, for instance, a word acquired a novel sense in the latter time period, then a cluster corresponding to this sense only consists of word usages from this period but not the earlier ones, which would be reflected by a higher divergence. However, a cluster does not necessarily correspond to a precise sense of the word. Each cluster would rather represent a specific usage or context. Moreover, a word may completely change its context without changing the meaning. Consequently, determining the number of clusters is a tricky part.

For clustering we used k-means with various values for k and affinity propagation [8]. Affinity propagation has been previously used for various linguistic tasks, such as word sense induction [2, 21]. Affinity propagation is based on incremental graph-based algorithm, partially similar to PageRank. Its main strength is that number of clusters is not defined in advance but inferred during training. We also experiment with the approach inspired by [3], where clusters with less than two members are considered weak and merged with the closest strong cluster, i.e. clusters with more than two members.[5] We refer to this method as two-stage clustering.

5 EXPERIMENTS

We focus our analysis on comparing the various clustering approaches and the metrics to detect semantic change. Table 1 shows the Pearson and Spearman correlations between the models' outputs and the human-annotated drifts. We also report Silhouette scores for clustering.

We use a pretrained version of BERT [6] and BERT fine-tuned on the COHA subcorpus for up to 10 epochs. We make use of the Scikit-learn implementation of k-means and affinity propagation [7]. For k-means, we set the number of clusters k and use default parameters for the rest. Similarly, for affinity propagation, we use the default parameters set by the library.

A specificity of BERT is the representation of words with byte-pair encodings [30]. Thus, some words can be divided into several sub-parts; for example, in our list of hundred target words for evaluation, *sulphate* is divided into two byte-pairs *sul* and *##phate*, where *##* denotes the splitting of the word. This is also true for the words *medieval*, *extracellular* and *assay*. We decided to exclude these words from our analysis. Thus, strictly speaking our results

[4] We refer the reader to the original implementation of transformer in [29] for a detailed overview of each component in the architecture.

[5] Note that procedure in [3] is more complex: they first find one or more number of representatives for each datapoint and then clustering is applied over representatives, while in our work clustering is done over the instances themselves.

[6] https://pytorch.org/hub/huggingface_pytorch-transformers/

[7] https://scikit-learn.org/stable/modules/clustering.html

Table 1: Correlations between detected semantic change and manually annotated list of semantic drifts [12] between 1960s and 1990s.

Method	Pearson	Spearman	Silhouette
Related work			
Gulardova & Baroni, 2011 [12]	0.386	-	-
Frermann & Lapata, 2016 [7]	-	0.377	-
Giulianelli, 2019 [10]	0.231	0.293	-
Kutuzov, 2020 [20]	0.233	0.285	-
Pretrained BERT			
Target word selection			
Variation	0.070	0.015	-
Variation by decade	0.239	0.303	-
Averaging by decade	0.295	0.272	-
Averaging	**0.354**	**0.349**	-
Clustering			
k-means, k = 3	0.461	0.444	0.104
k-means, k = 5	0.476	0.443	0.096
k-means, k = 7	0.485	0.434	0.091
k-means, k = 10	0.478	0.443	0.086
2-stage clustering, Aff. propagation	0.530	0.485	-
Affinity propagation	**0.548**	**0.486**	0.039
Fine-tuned BERT for 5 epochs			
Target word selection			
Averaging	0.317	0.341	-
Clustering			
k-means, k=3	0.411	0.392	**0.105**
k-means, k=5	0.539	0.508	0.098
k-means, k=7	0.526	0.491	0.092
k-means, k=10	0.500	0.466	0.088
k-means, k=100	0.315	0.337	0.042
2-stage clustering, Aff. propagation	0.554	0.502	-
Affinity propagation	**0.560**	**0.510**	0.043

are not directly comparable to some of the other approaches in the literature that do not employ BERT.

At the top of Table 1 we overview all previous work on the same test set. To train the models, [13] used GoogleBooks Ngrams, [8] used an extended COHA corpus, and both [11] and [21] used a subcorpus of COHA, identical to the one used in our experiments. In fact, the setting in [11] is quite similar to our work, though our best model performance is much higher than in [11]; we will further discuss this discrepancy in Section 6.

As can be seen in Table 1, among all metrics used for target word selection averaging yields the highest correlation with the human annotations. This intuitively makes sense since averaging measures semantic drift between the first and the last time step and the evaluation dataset was annotated by only considering the first and the last decade. Variation by decade also shows good results; it is a measure of the evolution of the level of variation of a word usage through time.

As can be seen in Table 1 affinity propagation on the fine-tuned BERT model yields the highest Spearman rank correlation. Results obtained using pretrained and fine-tuned models are consistent: in both runs averaging yields lower performance than clustering and affinity propagation is the best clustering method. Two-stage clustering works better than k-means but slightly worse than affinity propagation.

Fine-tuning BERT improves all models except for k-means with 3 clusters and averaging—we do not yet have a clear explanation for that exception.

To conclude, clustering fine-tuned embeddings using affinity propagation yields the best results, with a Pearson correlation with human annotation of 0.56. To evaluate the success of this result, we can use the value of the inter-rater agreement during the annotation process, which was 0.51, computed using the average of pair-wise Pearson correlations [12]. This highlights the difficulty of the task and the performance of the best method.

6 DISCUSSION

6.1 Error Analysis

We manually checked few examples by choosing the words that have less mentions in the corpus to be able to look through all sentences containing the word. One of the tricky cases for our model is the word *neutron*: according to the manual annotation,

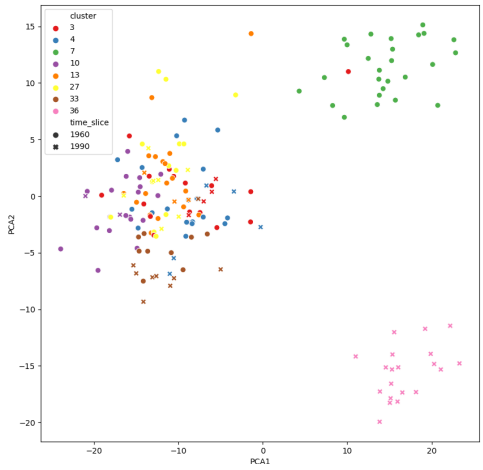


Figure 1: 2D PCA visualization for the biggest clusters obtained for word *neutron*.

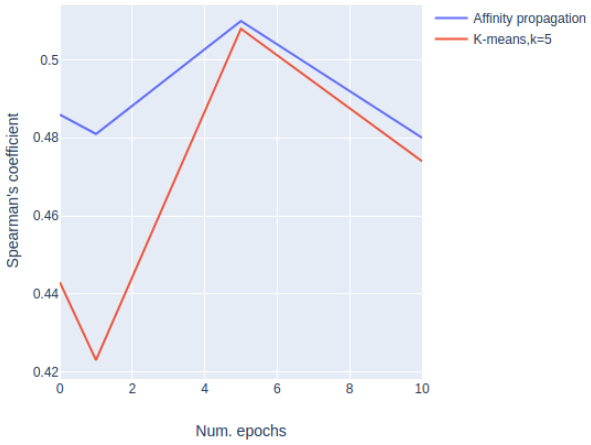


Figure 2: Impact of BERT fine-tuning on the performance of two distinct aggregation methods, affinity propagation and k-means with k=5.

it is ranked 81st and has a stable meaning, while our best model considered it one of the most changed and ranked it at 9.

We visualize the biggest clusters for *neutron* using PCA decomposition of BERT embeddings (Figure 1). There are two clearly distinctive clusters: cluster 36 in the bottom right corner, drawn with pink crosses, which consists only of instances from 1990s, and cluster 7 drawn with green dots in the top right corner, which consists only of instances from 1960s. A manual check reveals that the former cluster consists of sentences which mention *neutron stars*. Though neutron stars have been already discovered in 1960s they were probably less known[8] and are not represented in the corpus. In any case, a difference in a collocation frequency does not mean a semantic shift, since collocations often have a non-compositional meaning. Another similar example is a company called "Vector Security International" that appears only in 1990s time slice, which distorts semantic our calculations for the word *vector*. Our method could be improved by removing stable multiword expressions and named entities from the training set.

The latter distinctive cluster for *neutron*, consisting of word usages from 1960s, contains many sentences that have a certain pathetic style and elevated emotions, such as underlined in the examples below:

throughout the last several decades the dramatic revelation of this new world of matter has been dominated by a most remarkable subatomic particle – the neutron .

the discovery of the neutron by sir james chadwick in 1939. marked a great step forward in understanding the basic nature of matter .

The lack of such examples in 1990s might have a socio-cultural explanation or it could be a mere corpus artefact. In any case, this has nothing to do with semantic shift and demonstrates an ability of BERT to capture other aspects of language, including syntax and pragmatics.

[8] https://en.wikipedia.org/wiki/Neutron_star

6.2 Impact of Fine-tuning

Figure 2 shows the comparison of fine-tuning influence for two best clustering methods (affinity propagation, and k-means with k=5). Interestingly, a light fine-tuning (just for one epoch) decreases the performance of both methods (in terms of Spearman correlation) in comparison to no fine-tuning at all (zero epochs). After that, the length of fine-tuning until up to 5 epochs is linearly correlated with the performance increase.

Fine-tuning the model for five epochs appears optimal. After that, the performance for both methods starts decreasing, most likely because of over-fitting due to the reduced size of the fine-tuning dataset compared to the training data.

The impact of fine-tuning on the k-means clustering is stronger than on the affinity propagation. The difference between model's performance on 5 epochs is negligible. However, this effect holds only with k=5, other values of k do not demonstrate such a difference between original and fine-tuned models, as can be seen in Table 1.

6.3 Clustering

Results presented in Table 1 imply that most of the approaches for semantic change detection proposed in this work manage to outperform previous approaches by a large margin. We believe the differences in the numerical results should be primarily attributed to the differences in the methods, even though we can not draw a direct comparison to some of the approaches due to test set word removal and differences in the train corpora. We can however compare our results directly to the results published by [10] since they are also using BERT trained on the COHA corpus. Even more, their proposed clustering approaches are methodologically very similar to the approaches presented in this work, yet we manage to outperform their approach by a margin of about 35 percentage points when

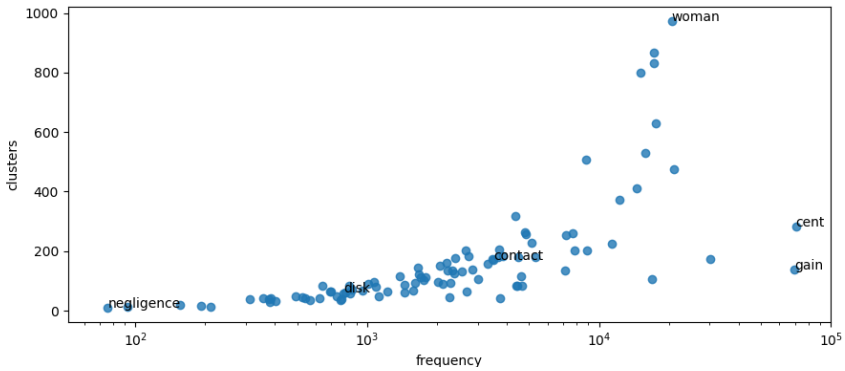


Figure 3: Number of clusters found by affinity propagation and frequency of a word in the 1960s and 1990s in COHA.

affinity propagation is used and by about 33 percentage points when k-means clustering[9], same as in [10], is used.

Unfortunately, [10] does not report a number of clusters that has been used, they only mention that the number of clusters has been optimised using the Silhouette scores. We can only speculate why their results are much lower than ours. The first hypothesis is connected with the usage of the Silhouette score, which might not be optimal for our goals. We compute the Silhouette score[10] for clusterings obtained by our methods. As can be seen in Table 1, the best Spearman correlation coefficient does not correspond to the best Silhouette score. Moreover, the Silhouette scores are quite close to zero.

The second hypothesis is connected with the difference in fine-tuning regimes employed in this research and the one conducted by [10]. We use domain adaptation fine-tuning, proving its efficiency for a certain number of epochs, for both k-means (except for a small number of clusters) and affinity propagation. However, [10] tried both diachronic fine-tuning (using the incremental fine-tuning technique first proposed by [18]) and domain-specific fine-tuning, but concluded that none led to an improvement in the results. As it was already speculated in [10], using both training regimes at the same time might lead to too extensive fine-tuning and therefore over-fitting. Further, a more thorough study on influence of incremental fine-tuning on contextual embeddings models (such as BERT) should perhaps be conducted, since the effects might differ from the ones observed for static embeddings models. Finally, the domain-specific fine-tuning is conducted only for 1 to 3 epochs, which might be too few to improve the results on some corpora.

The difference in performance between k-means and affinity propagation could be partially explained by the different number of clusters in the two approaches. Affinity propagation, which performs the best, outputs a huge amount of clusters—160 on average. The particular number of clusters found by affinity propagation for a word correlates strongly with the frequency of that word in the corpus with correlational coefficient $r = 0.875$, as is illustrated in Figure 3.

Thus, determining the optimal number of clusters for different words is not straightforward. We cannot claim that the clusters found by any of the methods we used can be interpreted as the different senses of a word or that they are even suitable for human interpretation. Most probably, affinity propagation captures subtle differences in word usages rather than global semantic shift. Nevertheless, it works better than k-means with smaller and more intuitive number of clusters, since word sense induction and semantic shift detection are not the same task.

Affinity propagation usually produces a skewed clustering, with a large number of small clusters containing only one or two data points, and can be used for outlier detection. K-means is not suitable for this task since it uses a random initialisation and if an outlier is not initially selected as a potential centroid it may never be found.

To justify this claim we conducted an additional experiment and run k-means clustering on fine-tuned embeddings using k=100 or number of instances minus one for less frequent words. As presented in Table 1, this resulted in Pearson and Spearman rank correlations of 0.315 and 0.337, respectively, which is worse than *any* other strategy we tried for fine-tuned embeddings, including averaging. At the same time, the Silhouette score for this insufficient model is almost equal to the Silhoutte score for the best model. Thus, the Silhouette score fails to discriminate between the best and the worst model.

7 FUTURE WORK

We plan to investigate how the clusters found by the methods in this work can be used to interpret the different usages of a word in a specific time slice. The initial experiments on this subject have already been conducted with the two-stage clustering, which removes the smallest clusters, containing one or two instances. Thus, it allows to focus on a smaller number of the most representative clusters, which might be more suitable for human interpretation even though it does not yield the best result. The initial check demonstrated that most of these clusters are interpretable, though some particular meaning can be spread among several clusters.

[9]Here we are referring to our best k-means configuration with five clusters and using a BERT model fine-tuned for five epochs.
[10]Using standard Scikit-learn implementation, https://scikit-learn.org/stable/modules/clustering.html#silhouette-coefficient

Our analysis hints that clustering BERT token embeddings for a word does not necessarily lead to sense-specific clusters. This conclusion is on par with [4]. Indeed, BERT ability do detect distinct word meanings has limitations. Thus, it would be interesting to extract only the semantic parts of the BERT embeddings to direct our analysis more towards word meaning and rather than word usage in general.

ACKNOWLEDGMENTS

We are grateful to Andrey Kutuzov for valuable discussions during this paper preparation. We also thank Pr. Alexandre Allauzen from ESPCI - Université Paris Dauphine and Pr. Asanobu Kitamoto from National Institute of Informatics (Tokyo) for their advises. 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] Jean Aitchison. 2001. Language Change: Progress Or Decay? In *Cambridge Approaches to Linguistics*. Cambridge University Press, Cambridge.

[2] Domagoj Alagić, Jan Šnajder, and Sebastian Padó. 2018. Leveraging lexical substitutes for unsupervised word sense induction. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[3] Asaf Amrami and Yoav Goldberg. 2019. Towards better substitution-based word sense induction. *arXiv preprint arXiv:1905.12598* (2019).

[4] Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fernanda B. Viégas, and Martin Wattenberg. 2019. Visualizing and Measuring the Geometry of BERT. In *NeurIPS*.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. (2019), 4171–4186.

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

[7] Lea Frermann and Mirella Lapata. 2016. A bayesian model of diachronic meaning change. *Transactions of the Association for Computational Linguistics* 4 (2016), 31–45.

[8] Brendan J Frey and Delbert Dueck. 2007. Clustering by passing messages between data points. *Science* 315, 5814 (2007), 972–976.

[9] Nabeel Gillani and Roger Levy. 2019. Simple dynamic word embeddings for mapping perceptions in the public sphere. In *Proceedings of the Third Workshop on Natural Language Processing and Computational Social Science*. 94–99.

[10] Mario Giulianelli. 2019. *Lexical Semantic Change Analysis with Contextualised Word Representations*. University of Amsterdam - Institute for logic, Language and computation.

[11] Yoav Goldberg and Jon Orwant. 2013. A dataset of syntactic-ngrams over time from a very large corpus of english books. In *Second Joint Conference on Lexical and Computational Semantics*. 241–247.

[12] Kristina Gulordava and Marco Baroni. 2011. A distributional similarity approach to the detection of semantic change in the Google Books Ngram corpus.. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*. 67–71.

[13] Martin Hilpert and Stefan Th Gries. 2008. Assessing frequency changes in multistage diachronic corpora: Applications for historical corpus linguistics and the study of language acquisition. *Literary and Linguistic Computing* 24, 4 (2008), 385–401.

[14] Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146* (2018).

[15] Renfen Hu, Shen Li, and Shichen Liang. 2019. Diachronic sense modeling with deep contextualized word embeddings: An ecological view. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 3899–3908.

[16] Patrick Juola. 2003. The time course of language change. *Computers and the Humanities* 37, 1 (2003), 77–96.

[17] Nobuhiro Kaji and Hayato Kobayashi. 2017. Incremental Skip-gram Model with Negative Sampling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 363–371.

[18] Yoon Kim, Yi-I Chiu, Kentaro Hanaki, Darshan Hegde, and Slav Petrov. 2014. Temporal Analysis of Language through Neural Language Models. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*. 61–65.

[19] Vivek Kulkarni, Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2015. Statistically significant detection of linguistic change. In *Proceedings of the 24th International Conference on World Wide Web*. 625–635.

[20] Andrey Kutuzov. 2020. Diachronic contextualized embeddings and semantic shifts. In *press*.

[21] Andrey Kutuzov, Elizaveta Kuzmenko, and Lidia Pivovarova. 2017. Clustering of Russian Adjective-Noun Constructions Using Word Embeddings. In *Proceedings of the 6th Workshop on Balto-Slavic Natural Language Processing*. 3–13.

[22] Andrey Kutuzov, Lilja Øvrelid, Terrence Szymanski, and Erik Velldal. 2018. Diachronic word embeddings and semantic shifts: a survey. In *Proceedings of the 27th International Conference on Computational Linguistics*. 1384–1397.

[23] Matej Martinc, Petra Kralj Novak, and Senja Pollak. 2020. Leveraging Contextual Embeddings for Detecting Diachronic Semantic Shift. In *LREC*.

[24] Hao Peng, Jianxin Li, Yangqiu Song, and Yaopeng Liu. 2017. Incrementally learning the hierarchical softmax function for neural language models. In *Thirty-First AAAI Conference on Artificial Intelligence*.

[25] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. 2227–2237.

[26] Alex Rosenfeld and Katrin Erk. 2018. 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)*. 474–484.

[27] Nina Tahmasebi, Lars Borin, and Adam Jatowt. 2018. Survey of Computational Approaches to Diachronic Conceptual Change. *arXiv preprint arXiv:1811.06278* (2018).

[28] Xuri Tang. 2018. A state-of-the-art of semantic change computation. *Natural Language Engineering* 24, 5 (2018), 649–676.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. 5998–6008.

[30] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144* (2016).

[31] Yating Zhang, Adam Jatowt, Sourav S Bhowmick, and Katsumi Tanaka. 2016. The past is not a foreign country: Detecting semantically similar terms across time. *IEEE Transactions on Knowledge and Data Engineering* 28, 10 (2016), 2793–2807.

D. Manuscript: Discovery Team at SemEval-2020 Task 1: Context-sensitive Embeddings not Always Better Than Static for Semantic Change Detection

Discovery Team at SemEval-2020 Task 1: Context-sensitive Embeddings not Always Better Than Static for Semantic Change Detection

Matej Martinc *
Jozef Stefan Institute, Slovenia
matej.martinc@ijs.si

Syrielle Montariol
Univ. Paris-Saclay, Societé Générale
LIMSI - CNRS, Univ. Paris-Sud, France
syrielle.montariol@limsi.fr

Lidia Pivovarova
University of Helsinki, Finland
lidia.pivovarova@helsinki.fi

Elaine Zosa
University of Helsinki, Finland
elaine.zosa@helsinki.fi

Abstract

This paper describes the approaches used by the Discovery Team to solve SemEval-2020 Task 1 - Unsupervised Lexical Semantic Change Detection. The proposed method is based on clustering of BERT contextual embeddings, followed by a comparison of cluster distributions across time. The best results were obtained by an ensemble of this method and static Word2Vec embeddings. According to the official results, our approach proved the best for Latin in Subtask 2.

1 Introduction

Each word has a variety of senses and connotations, constantly evolving through usage in social interactions and changes in cultural and social practices. Identifying and understanding these changes is important for linguistic research and social analysis, since it allows the detection of cultural and linguistic trends and possibly predict future changes. Detecting these changes can also be used to improve many NLP tasks, such as text classification and information retrieval.

The SemEval-2020 Task 1 — Unsupervised Lexical Semantic Change Detection (Schlechtweg et al., 2020) deals with detection of semantic change in temporal corpora containing texts from two distinct time periods in four languages: English, German, Latin and Swedish. The challenge defines two subtasks: Subtask 1 is binary classification, i.e., to determine whether a word has changed or not; SubTask 2 aims at ranking a set of target words according to their rate of semantic change.

In this paper, we present the approaches used by the Discovery Team to tackle these two subtasks. The Discovery Team qualified as 11th and 5th on Subtasks 1 and 2, respectively, and also proved the best for Latin language in Subtask 2. Our systems leverage the transformer-based BERT model to generate contextualised embeddings for each word usage. Then these embeddings are aggregated into meaningful time-specific word representations. We explore different aggregation techniques, such as clustering (k-means and affinity propagation) and averaging. We also combine BERT-based representations with static Word2Vec embeddings[1].

2 System Overview

2.1 Word Representation

In order to derive meaningful temporal representations for each target word, we adapted the methodology proposed in Martinc et al. (2020a) to the multilingual setting of the SemEval-2020 Task 1. The core component of our approach is the use of BERT (Bidirectional Encoder Representations from Transformers), a pretrained masked language model based on the transformer architecture (Devlin et al., 2019). We use specific models for each language—for English: bert-base-uncased model, for Swedish: bert-base-swedish-uncased (https://github.com/af-ai-center/SweBERT), for German: bert-base-german-cased (https://deepset.ai/german-bert), for Latin: bert-base-multilingual-uncased

*All authors contributed equally to this research.

[1]Code for the experiments is available under the MIT license at https://github.com/smontariol/Semeval2020-Task1.

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 67–73
Barcelona, Spain (Online), December 12, 2020.

model—all with 12 attention layers and a hidden layer of size 768. German is the only language for which we use a cased model since most target words are nouns, which are capitalized in German. The only model available for Latin is a multilingual BERT model trained on 104 languages, including Latin.

For each language, the model is fine-tuned for five epochs on the task's corpus, as advised by Martinc et al. (2020a). This fine-tuning is unsupervised, i.e., a masked language model objective is used in the fine-tuning step (Devlin et al., 2019) in order to adapt each model to a specific corpus.

The fine-tuned models are used to generate token embeddings. The corpus for each language is split into two periods and the fine-tuned models are fed with sentences containing one or more target words from the sub-corpus. The sentences are split into tokens, and an embedding of dimension 768 is generated for each token by summing the last four encoder output layers of BERT, as advised by recent studies which confirm the fact that semantic features are captured in higher layers of BERT (Jawahar et al., 2019).

Note that byte-pair tokenization (Kudo and Richardson, 2018) in some cases generates tokens that correspond to sub-parts of words. To generate embedding representations for the target words split into sub-parts, we concatenate the embeddings of each byte-pair token constituting a word. After this procedure, we obtain a contextual embedding representation for each target-word usage, together with the time period each word usage representation belongs to.

In addition to context-dependent embeddings, we generate static word representations by training a 300-dimensional Word2Vec model using the skip-gram architecture with negative sampling (SGNS) (Mikolov et al., 2013) for each time slice. We align the embeddings from the different time slices using the Orthogonal Procrustes (OP) method as in Hamilton et al. (2016). We also applied pre- and post-processing steps such as mean-centering and vector normalisation, as recommended in Schlechtweg et al. (2019) [2].

2.2 Measures of Semantic Change

We use two methods to aggregate contextual embeddings from BERT: averaging and clustering. The methods were introduced and compared in our previous work (Martinc et al., 2020a; Martinc et al., 2020b).

Averaging is a simple aggregation approach where all target-word usage representations from a given time period are averaged. A quantitative estimate of semantic change for each target word is measured by computing the cosine distance between two averaged time-specific representations of the word.

Clustering of word usage representations results in sets of word usages, where each set is expected to correspond to a single word sense or a specific context. We create two time-specific cluster distributions by counting the number of cluster members for each time period and creating a vector of cluster counts for each cluster within a time period. Then the Jensen-Shannon divergence (JSD) between two period-specific distributions is used to measure the semantic change, as in Martinc et al. (2020a).

We use two clustering techniques, *affinity propagation* (Frey and Dueck, 2007) and *k-means*[3]. Affinity propagation has been extensively used in the literature for semantic tasks such as word sense induction (Alagić et al., 2018). It works by exchanging messages between data points until a high-quality set of *exemplars*, i.e. members of the input set that are representative of clusters, is obtained. A big advantage of this method is that it considers all the data points as potential cluster centers and therefore does not require the number of clusters to be defined in advance. K-means is a very popular clustering method and has been shown to perform well for the semantic change detection task (Giulianelli et al., 2020). Contrarily to affinity propagation, it requires to define the number of clusters in advance. We try several values of k; the highest accuracy for this task is obtained with $k = 5$.

To obtain a measure of semantic change using static embeddings, we measure the cosine distance between the aligned embedding representations of the same word from two time slices.

2.3 Subtask 1: Binary Classification

In order to determine whether a target word has changed or not, we experiment with two distinct methods, *thresholding using stopwords* and *identification of period-specific clusters*.

[2]We trained Word2Vec using the code from `https://github.com/Garrafao/LSCDetection`

[3]We use the Scikit-learn implementations (`https://scikit-learn.org/stable/modules/clustering.html`) with default parameters, except for the number of clusters for k-means, for which we tried several options.

		English	Latin	Swedish	German
Number of stopwords		109	334	78	142
Mean JSD	stopwords	0.181	0.210	0.355	0.328
	targets	0.239	0.264	0.460	0.384

Table 1: Number of stopwords used and average semantic change score (JSD) for target words and stopwords.

2.3.1 Thresholding Using Stopwords

We want to find the best threshold in the ranked list of target words by relying on the assumption that stopwords—words that are very frequent in a language and play primarily auxiliary roles—undergo a low semantic change.

Though stopwords are more stable than most words of the dictionary, they can still change their meaning due to the grammaticalisation processes, i.e. when a previously meaningful word looses most of it functions except for auxiliary ones. For example, the English stopword *hence* used to have a concrete deictic meaning "from here" (e.g. "hence we go") but nowadays it is used only to connect two propositions. Since not all stopwords are stable, finding an appropriate threshold is not straightforward.

It should be noted that stopwords have extremely context-specific representations (Ethayarajh, 2019). However, high polysemy and highly variable context do not necessarily induce more semantic change (Martinc et al., 2020a). We check the difference of average semantic change between a set of stopwords and the list of target words for all languages.

First, to compute semantic change scores for a list of stopwords, we use the same procedure that was used for the target words. For all languages except Latin, we create a list of stopwords by taking the words at the intersection of the `nltk` and `Spacy` stopword lists. For Latin, we use an external resource[4]. We keep only stopwords with more than 30 occurrences in each period; the number of stopwords per language is shown in Table 1. When the number of occurrences of a word is too high, we sample 5000 sentences per period for this word. As can be seen in Table 1, the mean JSD for stopwords is sensibly lower than the one for target words.

Then, we compare stopword and target word score distributions in order to define a threshold below which a target word should be classified as unchanged.

We first divide the stopwords' semantic change score distribution into 10 bins to derive a frequency distribution in a shape of a histogram with 10 columns, as exemplified for English in Figure 1. We take the threshold as the local maximum score of the bin in the histogram containing a number of words lower than an epsilon ϵ. We exclude the first bin, which is composed of very stable words and can sometimes have a size smaller than ϵ. The frequency limit ϵ used to select the threshold depends on the number of stopwords for each language: $\epsilon = 1/10 * \textit{number-of-stopwords}$. We compute two sets of thresholds: the leftmost and the rightmost points of the border bin, as shown in the Figure 1. The higher threshold is more conservative, meaning that fewer words are classified as changed.

2.3.2 Identification of Period-Specific Clusters

The second method looks for concrete indications of semantic change, such as the appearance or disappearance of a specific word sense. Target word clusters should to some extent resemble different word senses, allowing identification of target words that obtained or lost a meaning. If one of the clusters for a target word contains word occurrences from one time period and contains less or equal than k (where k=2) word occurrences from another time period, we assume that this word has lost or gained a specific meaning.

Since clustering methods sometimes produce small-sized clusters, we consider only the clusters bigger than a threshold, in order to focus on the "main" usages of a word. Thus, for k-means we enforce a constraint that a cluster should contain at least 10 word occurrences to be considered in the analysis. For affinity propagation, we implement a dynamic threshold strategy: the threshold beyond which we consider

[4]List of Latin stopwords: https://github.com/aurelberra/stopwords

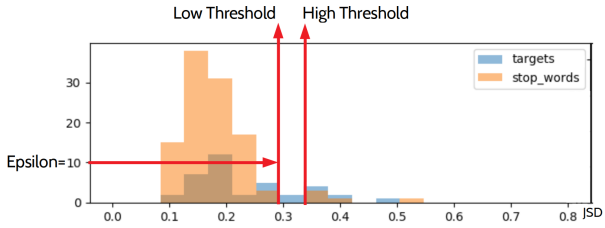


Figure 1: Distribution of semantic change scores in the English corpus: target words VS stopwords

	aff-prop	avg	kmeans_5	W2V	GS
aff-prop	1				
averaging	0.789	1			
kmeans_5	0.815	0.811	1		
word2vec	0.501	0.558	0.481	1	
Gold Standard	0.298	0.397	0.305	0.394	1

Table 2: Spearman correlation between the semantic change scores of various methods and the gold standard, averaged for all languages.

a cluster is computed for each target word as twice its average cluster size.

2.4 Subtask 2: Ranking

For Subtask2, target words were ranked according to the semantic change scores described in Section 2.2, namely divergence between cluster distributions (JSD) or cosine distance. Additional steps were performed in some of our submissions to improve this basic approach: cluster filtering and ensembling.

2.4.1 Cluster Filtering

Affinity propagation tends to produce a large number of clusters, and cluster size distribution is highly skewed. We try several heuristics to filter out the clusters that potentially contain noise and can distort the comparison between time periods. The first idea is to remove the smallest clusters (containing only one or two instances), whose appearance in a given time period is not significant. The second idea is to filter out sentences in which a target word is used as a proper noun, as in the following example: *her daddy warn everyone that rose* ***lane_nn*** *be bring home a musician with long hair*.

Finally, we noticed that some clusters contain sentences that refer to specific events. For example, one of the clusters for *attack* contains sentences about terrorist attack in Israel and consists only of sentences from the latter time period, for the obvious reasons. The sentences in this cluster contain many named entities (NEs), e.g.: *hezbollah leader hassan fadlallah defend* ***attack_nn*** *on israeli civilian target civilian be a war crime*. We filter out clusters that contain too many NEs in some of our submissions, though this "radical" NE filtering may have drawbacks: one may argue that a "terrorist attack" is a new meaning of a word *attack* that was correctly distinguished by the clustering algorithm but then discarded by filtering.

In a real-world application, NE recognition should be done on documents with preserved capitalization, preferably using a model trained specifically on historical documents. For the shared task we rely on out-of-the-box NLP pipelines.[5] Most of the tools are unable to recognize names in lowercased lemmatized text but POS-taggers are more reliable: e.g., the SpaCy NE recognition model was unable to recognize lower-cased names even if the SpaCy POS-tagger labeled the corresponding tokens as proper nouns.

We performed the NE filtering as a post-processing step, to compensate for errors in the NE recognition: we filter out a cluster if at least 80% of the target word mentions are NE. For the radical filtering, a cluster is filtered out if the number of proper nouns is 5 times larger than the number of sentences.

2.4.2 Ensembling

We ensemble different approaches for semantic change detection by multiplying the semantic change scores produced by different methods for each target word. We choose multiplication rather than the arithmetic average since the underlying distributions of the semantic shift measures are unknown, even though they produce numbers within the same range. If, for example, the numerical values of a particular measure are generally larger than values of another measure, the former measure would contribute more to the average and thus dominate the ensemble. Multiplication does not have this side effect.

We experiment with different combinations of averaging, clustering and Word2Vec based methods in order to test the hypothesis that the synergy between contextualised and static embeddings improves the

[5]We used SpaCy for English and German (`https://spacy.io/`), Polyglot for Swedish (`https://pypi.org/project/polyglot/`) and CLTK for Latin (`http://cltk.org/`).

Model	Binary method	AVG	English	German	Latin	Swedish
k-means 5	time-period specific clusters	0.600	0.649	0.542	0.500	**0.710**
aff-prop	time-period specific clusters, dynamic threshold	0.496	0.568	0.458	**0.700**	0.258
aff-prop, merging cluster	time-period specific clusters, dynamic threshold	0.545	0.514	0.542	0.575	0.548
aff-prop	stopwords, high threshold	0.573	0.622	0.604	0.550	0.516
aff-prop	stopwords, low threshold	0.552	**0.703**	0.667	0.450	0.387
ensemble: averaging + aff-prop	stopwords, low threshold	**0.621**	0.568	**0.688**	0.550	0.677

Table 3: SubTask 1 results: accuracy.

overall performance. Combinations of models that are too strongly correlated (above 0.8) are discarded. Some correlations averaged for all languages can be found in Table 2, though these values hide important disparities among languages.

3 Results

3.1 Subtask1

The results for the binary classification are shown in Table 3. We use BERT fine-tuned on the Semeval corpora for all submissions. The best official result was achieved by applying the stopword thresholding method to rankings obtained by measuring the JSD between affinity propagation cluster distributions. The stopwords thresholding method seems to work best with higher thresholds, which classify fewer words as changed.

The method of identifying period-specific clusters worked competitively when conducted on k-means clusters but performed worse with affinity propagation, since the latter method usually produces a large number of clusters. Reducing the number of clusters by merging the closest clusters together increased the performance of the method.

Looking at the average accuracy, the stopwords method seems to work better than the period-specific clusters method. However, we face high discrepancies between languages. Comparing the results for the same model, i.e. BERT with affinity propagation clustering, the latter method worked best for Latin and worse than the stopwords method for all the other languages.

3.2 Subtask2

Results for SubTask 2 are presented in Table 4. The best official result was obtained by an ensemble of Word2Vec static embeddings and fine-tuned BERT contextual embeddings, further improved with radical NE filtering as a postprocessing step—see row $\#11$ in the table. The good performance of the method can be explained by the fact that the semantic change scores outputted using static embeddings and contextualised embeddings are not highly correlated, as shown in Table 2 and we speculate that these two types of embedding capture different aspects of the semantic change.

Ensembling of four different methods—affinity propagation, K-means (k=5), averaging and Word2Vec with OP alignment—allows the merging of all the information that they gather $(\#12)$. However, it still does not out-perform the ensemble of only affinity propagation and Word2Vec $(\#10)$.

The cosine distance between averaged contextual embeddings performs much better than between Word2Vec representations for Latin but worse for other languages (rows $\#8$ and $\#9$). The affinity propagation clustering, which was the best in our previous study (Martinc et al., 2020a), did not perform well (rows $\#1$ to $\#6$), especially for Swedish, where it performed close to random. One explanation for this discrepancy could be the shuffling of sentences in the shared task corpora. BERT models cannot leverage the usual sequence of 512 tokens as a context in this setting but are limited to the number of tokens in the sentence. The correlation between larger context and better performance of the transformer-based models has been shown on some NLP tasks before (Dai et al., 2019). Therefore, the lack of context could have a detrimental effect on the quality of BERT contextual embeddings. The results however do suggest that by averaging these embeddings, a static embedding of good quality for each target token can be obtained.

The radical NE filtering has a significant impact on English and German results (compare rows $\#2$ to $\#5$), though in the opposite directions: it improves the performance on the English corpus from 0.313

	Input	Method	Post-Processing	AVG	English	German	Latin	Swedish
Clustering								
1	pretrained BERT	aff-prop, JSD	-	0.278	0.216	0.488	**0.481**	-0.072
2	fine-tuned BERT	aff-prop, JSD	-	0.298	0.313	0.436	0.467	-0.026
3	fine-tuned BERT	aff-prop, JSD	small clusters	0.302	0.327	0.440	0.472	-0.030
4	fine-tuned BERT	aff-prop, JSD	target NE	0.300	0.328	0.426	0.467	-0.023
5	fine-tuned BERT	aff-prop, JSD	NE	0.295	**0.436**	0.302	0.467	-0.025
6	fine-tuned BERT	aff-prop, JSD	NE, small clusters	0.291	0.413	0.310	0.472	-0.029
7	fine-tune BERT	kmeans k=5, JSD	-	**0.320**	0.189	**0.528**	0.324	**0.238**
Methods not using clustering								
8	fine-tune BERT	averaging, cosine dist	-	**0.397**	0.315	0.565	**0.496**	0.212
9	word2vec OP	cosine dist	(Schlechtweg et al., 2019)	0.394	**0.341**	**0.691**	0.131	**0.413**
Ensembling								
10	aff-prop (#2) + w2v (#9)	distance multiplication	-	0.417	0.357	**0.642**	0.366	0.303
11	aff-prop (#2) + w2v (#9)	distance multiplication	NE, small clusters	**0.442**	**0.361**	0.603	**0.460**	**0.343**
12	aff-prop(#2), k-means (#7), averaging (#8), w2v (#9)	multiplication, equal weights	-	0.403	0.279	0.607	0.451	0.276

Table 4: SubTask 2 results: Spearman correlation with the ground truth. Submissions made during the official evaluation phase are marked with yellow. Numbers preceded with # refer to the rows in this table, i.e. models used for the ensembling.

to 0.436 but reduces it on the German corpus from 0.436 to 0.302. Filtering as such slightly reduces the average performance (compare #2 to #6), but by removing small clusters (row #3) we gain slight improvements for all four corpora. The best performing method also uses filtering, which improves the ensemble performance for all corpora except for German (compare rows #10 and #11).

Many of the techniques that we try improved the overall method performance only for English: BERT fine-tuning, affinity propagation clustering, NE filtering. This might be related to the fact that the corpora are lemmatized, and lemmatization has a smaller effect on English, with its reduced morphology. The poor results on the Swedish corpus might be related to OCR-errors, leading to a large number of out-of-vocabulary tokens. BERT models deal with the out-of-vocabulary words by using a vocabulary of sub-word units (Kudo and Richardson, 2018). However, the vocabulary size is fixed and consists of 30,522 sub-word units, which might not be enough for a noisy corpus. This is supported by the findings of the another participant of the SemEval-2020 Task 1, which showed that character-based embeddings (ELMo) yield a significant improvement over BERT on the Swedish corpus (Kutuzov and Giulianelli, 2020).

4 Conclusion

We present the approaches employed by the Discovery team to tackle SemEval-2020 Task 1: Unsupervised Lexical Semantic Change Detection (Schlechtweg et al., 2020). While our main method was based on clustering BERT contextualised embeddings, the best official result was obtained by combining this technique with a method for semantic change detection based on static Word2Vec embeddings.

The methods based on contextualised embeddings with clustering are outperformed by averaging of contextualised embeddings and static embeddings methods. Other task participants, in particular the winning team of Subtask 2, used similar static and contextual methods and reached the same conclusion on the adequacy of static embeddings for these specific tasks and corpora (Pömsl and Lyapin, 2020). However, the discrepancy among languages is significant and the results averaged on all four corpora can be misleading. A more thorough analysis on how different embeddings perform in different settings (short or long term semantic change, type of corpus preprocessing, etc...) and different languages will be performed in the future work.

Acknowledgements

This work has been partly supported by the European Union's Horizon 2020 research and innovation programme under grant 770299 (NewsEye) and 825153 (EMBEDDIA), and Project Development of Slovene in the Digital Environment (RSDO), co-financed by the Republic of Slovenia and the European Union from the European Regional Development Fund.

References

Domagoj Alagić, Jan Šnajder, and Sebastian Padó. 2018. Leveraging lexical substitutes for unsupervised word sense induction. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. In *EMNLP/IJCNLP*.

Brendan J Frey and Delbert Dueck. 2007. Clustering by passing messages between data points. *Science*, 315(5814):972–976.

Mario Giulianelli, Marco Del Tredici, and Raquel Fernández. 2020. Analysing lexical semantic change with contextualised word representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3960–3973, Online, July. Association for Computational Linguistics.

William L Hamilton, Jure Leskovec, and Dan Jurafsky. 2016. Diachronic word embeddings reveal statistical laws of semantic change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1489–1501.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy, July. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

Andrey Kutuzov and Mario Giulianelli. 2020. Uio-uva at semeval-2020 task 1: Contextualised embeddings for lexical semantic change detection. *To appear in SemEval@COLING2020*.

Matej Martinc, Syrielle Montariol, Elaine Zosa, and Lidia Pivovarova. 2020a. Capturing evolution in word usage: Just add more clusters? In *Companion Proceedings of the Web Conference 2020*, pages 343–349.

Matej Martinc, Petra Kralj Novak, and Senja Pollak. 2020b. Leveraging contextual embeddings for detecting diachronic semantic shift. In *LREC*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Martin Pömsl and Roman Lyapin. 2020. Circe at semeval-2020 task 1: Ensembling context-free and context-dependent word representations. *To appear in SemEval@COLING2020*.

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. *To appear in SemEval@COLING2020*.

E. Manuscript: Scalable and Interpretable Semantic Change Detection

Scalable and Interpretable Semantic Change Detection

Syrielle Montariol* LISN - CNRS, Univ. Paris-Saclay Societé Générale syrielle.montariol@limsi.fr

Matej Martinc* Jozef Stefan Institute matej.martinc@ijs.si

Lidia Pivovarova University of Helsinki lidia.pivovarova@helsinki.fi

Abstract

Several cluster-based methods for semantic change detection with contextual embeddings emerged recently. They allow a fine-grained analysis of word use change by aggregating embeddings into clusters that reflect the different usages of the word. However, these methods are unscalable in terms of memory consumption and computation time. Therefore, they require a limited set of target words to be picked in advance. This drastically limits the usability of these methods in open exploratory tasks, where each word from the vocabulary can be considered as a potential target. We propose a novel scalable method for word usage-change detection that offers large gains in processing time and significant memory savings while offering the same interpretability and better performance than unscalable methods. We demonstrate the applicability of the proposed method by analysing a large corpus of news articles about COVID-19.

1 Introduction

Studying language evolution is important for many applications, since it can reflect changes in the political and social sphere. In the literature, the study of language evolution either focuses on long-term changes in the meaning of a word, or on more common short-term evolutionary phenomena, such as the word suddenly appearing in a new context, while keeping its meaning unchanged in a lexicographic sense. We refer to all types of language evolution—short- or long-term, with or without meaning change—as word usage change, a broad category that includes semantic change, but also any shifts in the context in which a word appears.

Recent studies (Giulianelli et al., 2020; Martinc et al., 2020a) show that clustering of contextual embeddings could be a proxy for word usage change: if clusters, which in theory capture distinct word usages, are distributed differently across time periods, it indicates a possible change in word's context or even loss or gain of a word sense. Thus, the cluster-based approach offers a more intuitive interpretation of word usage change than alternative methods, which look at the neighborhood of a word in each time period to interpret the change (Gonen et al., 2020; Martinc et al., 2020b) and ignore the fact that a word can have more than one meaning. The main limitation of the cluster-based methods is the scalability in terms of memory consumption and time: clustering is applied to each word in the corpus separately and all occurrences of a word need to be aggregated into clusters. For large corpora with large vocabularies, where some words can appear millions of times, the use of these methods is severely limited.

To avoid the scalability issue, cluster-based methods are generally applied to a small set of less than a hundred manually pre-selected words (Giulianelli et al., 2020; Martinc et al., 2020a). This drastically limits the application of the methods in scenarios such as identification of the most changed words in a large corpus or measuring of usage change of extremely frequent words, since clustering of all of word's contextual embeddings requires large computational resources. One way to solve the scalability problem using contextual embeddings is to *average* a set of contextual representations for each word into a single static representation (Martinc et al., 2020b). Averaging, while scalable, loses a lot on the interpretability aspect, since word usages are merged into a single representation.

The method we propose in this paper tackles scalability and interpretability at the same time. The main contributions of the paper are the following:

- A *scalable* method for contextual embeddings clustering that generates interpretable representations and outperforms other cluster-based methods.
- A method of measuring word usage change between periods with the *Wasserstein distance*. As far as we are aware, this is the first paper leverag-

* These authors contributed equally.

ing optimal transport for lexical semantic change detection.

- A *cluster filtering* step, which balances the deficiencies of clustering algorithms and consistently improves performance.
- An *interpretation pipeline* that automatically labels word senses, allowing a domain expert to find the most changing concepts and to understand *how* those changes happened.

The practical abilities of our method are demonstrated on a large corpus of news articles related to COVID-19, the Aylien Coronavirus News Dataset[1]. We compute the degree of usage change of almost 8,000 words, i.e., all words that appear more than 50 times in every time slice of the corpus, in the collection of about half a million articles in order to find the most changing words and interpret their drift[2].

2 Related Work

Diachronic word embedding models have undergone a surge of interest in the last two years with the successive publications of three articles dedicated to a literature review of the domain (Kutuzov et al., 2018; Tahmasebi et al., 2018; Tang, 2018). Most approaches build static embedding models for each time slice of the corpus and then make these representations comparable by either employing *incremental updating* (Kim et al., 2014) or *vector space alignment* (Hamilton et al., 2016b). The alignment method has proved superior on a set of synthetic semantic drifts (Shoemark et al., 2019) and has been extensively used (Hamilton et al., 2016b; Dubossarsky et al., 2017) and improved (Dubossarsky et al., 2019) in the literature. The recent SemEval Task on Unsupervised lexical semantic change detection has shown that this method is most stable and yields the best averaged performance across four SemEval corpora (Schlechtweg et al., 2020).

Yet another approach (Hamilton et al., 2016a; Yin et al., 2018) is based on comparison of neighbors of a target word in different time periods. This approach has been recently used to tackle the scalability problem (Gonen et al., 2020).

In all these methods, each word has only one representation within a time slice, which limits the sensitivity and interpretability of these techniques.

The recent rise of contextual embeddings such as BERT (Devlin et al., 2019) and ELMO (Peters et al., 2018) introduced significant changes to word representations. Contextual embeddings can be used for usage change detection by aggregating the information from the set of token embeddings. This can be done either through averaging of all vectors within a time slice and then computing averaged vector similarity (Martinc et al., 2020b), by computing a pairwise distance between vectors from different time slices (Kutuzov and Giulianelli, 2020), or by clustering all token representations to approximate its set of senses (Giulianelli et al., 2020). The analysis in this paper derives from this last set of methods, which demonstrate a higher performance than static embeddings methods at least on some datasets (Martinc et al., 2020a).

Automatic semantic shift detection has been used for text stream monitoring tasks, such as event detection (Kutuzov et al., 2017) viewpoint analysis (Azarbonyad et al., 2017) or monitoring of rapid discourse changes during crisis events (Stewart et al., 2017). None of these applications use clustering techniques and, as far as we are aware, only Martinc et al. (2020b) uses contextual embeddings for news stream analysis. In this paper we demonstrate the large potential of contextual embeddings for the *interpretable* tracking of short-term changes in word usage, which has a practical application for crisis-related news monitoring.

3 Scalability and Interpretability Limitations of Previous Methods

The main motivation for this research are the scalability or interpretability issues of previous methods for word usage change detection. The ones using contextual embeddings are either interpretable but unscalable (Giulianelli et al., 2020; Martinc et al., 2020a) or scalable but uninterpretable (Martinc et al., 2020b). The scalability issues of interpretable methods can be divided into two problems.

Memory consumption: Giulianelli et al. (2020) and Martinc et al. (2020a) apply clustering on all embeddings of each target word. This procedure becomes unfeasible for large sets of target words or if the embeddings need to be generated on a large corpus, since too many embeddings need to be saved into memory for further processing. To give an example, single-precision floating-point in Python requires 4 bytes of memory. Each contextual embedding contains 768 floats (Devlin et al.,

[1] https://blog.aylien.com/free-coronavirus-news-dataset/

[2] The code can be found at https://github.com/matejMartinc/scalable_semantic_shift

2019), leading each embedding to occupy 3072 bytes[3]. To use the previous methods on the Aylien Coronavirus News Dataset, which contains 250M tokens, about 768 Gb RAM would be necessary to store the embeddings for the entire corpus. If we limit our vocabulary to the 7,651 words that appear at least 50 times in every time slice and remove the stopwords (as we do in this work), we still need to generate contextual embeddings for 120M tokens, which is about 369 Gb of RAM.

Complexity of clustering algorithms: For the complexity analyses, we denote by d the dimension of the embedding, k is the number of clusters and n is the number of contextual embeddings, i.e., the number of word occurrences in the corpus. The time complexity of the affinity propagation algorithm (the best performing algorithm according to Martinc et al. (2020a)) is $O(n^2td)$, with t being the predefined maximum number of iterations of the data point message exchange. The time complexity of the simpler k-means algorithm[4] can be stated as $O(tknd)$, where t is the number of iterations of Lloyd's algorithm (Lloyd, 1982). As an example, consider the word *coronavirus*, which appears in the Aylien corpus about 1,2M times. For k-means with $k = 5$ and a maximal number of iterations set to 300 (the Scikit library default), about $300 * 5 * 1,300,000 * 768 \approx 1.5 \times 10^{12}$ operations are conducted for the clustering. With affinity propagation with the maximum number of iterations set to 200 (the default), clustering of the word *coronavirus* would require $1,300,000^2 * 200 * 768 \approx 2.6 \times 10^{17}$ operations, which is impossible to conduct in a reasonable amount of time on a high end desktop computer.

Contextual Embeddings Method with Interpretability Limitations: The averaging approach (Martinc et al., 2020b) eliminates the scalability problems: token embeddings for each word are not collected in a list but summed together in an element-wise fashion, which means that only 768 floats need to be saved for each word in the vocabulary. The averaged word representation is obtained for each time slice by dividing the sum by the word count. A single embedding per word is saved, leading to only 23.5 Mb of RAM required to store the embeddings for 7,651 words. These representations loose on the interpretability aspect, since all word usages are merged into a single averaged representation. It makes the method inappropriate for some tasks such as automatic labelling of word senses, and in some cases affects the overall performance of the method (Martinc et al., 2020a).

4 Methodology

Our word usage change detection pipeline follows the procedure proposed in the previous work (Martinc et al., 2020a; Giulianelli et al., 2020): for each word, we generate a set of contextual embeddings using BERT (Devlin et al., 2019). These representations are clustered using k-means or affinity propagation and the derived cluster distributions are compared across time slices by either using Jensen-Shannon divergence (JSD) (Lin, 2006) or the Wasserstein distance (WD) (Solomon, 2018). Finally, words are ranked according to the distance measure, assuming that the ranking resembles a relative degree of usage shift.

The primary contributions of this work lay in the embedding generation step, which improves the scalability of the method, and in leveraging WD to compute the distance between clusters. We also propose post-processing steps, which domain experts could use for the interpretation of results. We now describe the pipeline in more details.

4.1 Embeddings Generation

We use a pre-trained BERT model for each language of the evaluation corpora[5]. All models have 12 attention layers and a hidden layer of size 768. We fine-tune them for domain adaptation on each corpus as a masked language model for 5 epochs. Then, we extract token embeddings from the fine-tuned models. Each corpus is split into time slices. The models are fed 256 tokens long sequences in batches of 16 sequences at once. We generate sequence embeddings by summing the last four encoder output layers of BERT, following Devlin et al. (2019). Next, we split each sequence into 256 subparts to obtain a separate contextual embedding of size 768 for each token. Since one token does not necessarily correspond to one word due to byte-

[3]If we ignore the additional memory of a Python container—e.g., a Numpy list or a Pytorch tensor—required for storing this data.

[4]Here we are referring to the Scikit implementation of the algorithm employed in this work: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html.

[5]For German: bert-base-german-cased (https://deepset.ai/german-bert, for English: bert-base-uncased model, for Latin: bert-base-multilingual-uncased model from the `huggingface` library, for Swedish: bert-base-swedish-uncased (https://github.com/af-ai-center/SweBERT).

pair tokenization, we average embeddings for each byte-pair token constituting a word to obtain embeddings for each occurrence of a word.

Next, after obtaining a contextual embedding vector for each target word in a specific sequence, we decide whether this vector should be *saved* to the list or *merged* with one of the previously obtained vectors for the same word in the same time slice. To improve the scalability, we limit the number of contextual embeddings that are kept in the memory for a given word and time slice to a predefined threshold. The threshold of 200 was chosen empirically from a set of threshold candidates (20, 50, 100, 200, 500) and offers a reasonable compromise between scalability and performance. The new vector is merged if it is too similar—i.e., a duplicate or a near-duplicate—to one of the saved vectors or if the list already contains a predefined maximum number of vectors (200 in our case).

More formally, we add the new embedding e_{new} to the list of word embeddings $L = \{e_i, ..., e_n\}$ if:

$$|L| < 200 \quad \& \quad \forall e_i \in L\colon s(e_{\text{new}}, e_i) < 1 - \varepsilon$$

where s is the cosine similarity and ε is a threshold set to 0.01.

If $|L| \geq 200$ or if any vector in the list L is a near duplicate to e_{new}, we find a vector e_m in the list which is the closest to e_{new} in terms of cosine similarity:

$$e_m = \arg\max_{e_i \in L} s(e_i, e_{\text{new}})$$

This element e_m is then modified by summing it with e_{new}:

$$e_m \leftarrow e_m + e_{\text{new}}$$

The number of summed-up elements for each of the 200 groups in the list is stored besides their summed-up representations. Once the model has been fed with all the sequences in the time slice, the final summed-up vector is divided by this number to obtain an averaged embedding.

By having only 200 merged word embeddings per word per time slice, and by limiting the vocabulary of the corpus to 7,651 target words, we require up to 4.7 Gb of space for each time slice, no matter the size of the corpus. While this is still 200 times more space than if the averaging method was used (Martinc et al., 2020b), the conducted experiments show that the proposed method nevertheless keeps the bulk of the interpretability of the less scalable method proposed by Giulianelli et al. (2020), and offers competitive performance on several corpora.

4.2 Clustering

After collecting 200 vectors for each word in each time slice, we conduct clustering on these lists to extract the usage distribution of the word at each period. Clustering for a given word is performed on the set of all vectors from all time slices jointly.

We use two clustering methods previously applied for this task, namely k-means used in Giulianelli et al. (2020) and affinity propagation in Martinc et al. (2020a). The main strength of affinity propagation is that the number of clusters is not defined in advance but inferred during training. The clustering is usually skewed: a limited number of large clusters is accompanied with many clusters consisting of only a couple of instances. Thus, affinity propagation allows to pick out the core senses of a word. K-means tends to produce more even clusters. Appearance of small clusters that contain only few instances and do not represent a specific sense or usage of the word is nevertheless relatively common, since BERT is sensitive to syntax and pragmatics, which are not necessarily relevant for usage change detection. Another limitation of the k-means algorithm is that the number of clusters needs to be set in advance. This means that if the number of actual word usages is smaller than a predefined number of clusters, k-means will generate more than one cluster for each word usage.

To compensate for these deficiencies, we propose an additional *filtering and merging* step. A cluster is considered to be a legitimate representation of a usage of the word, if it contains at least 10 instances[6]. We compute the average embedding inside each cluster, and measure the cosine distance (1 - cosine similarity) between the average embeddings in each pair of legitimate clusters for a given word. If the distance between two clusters is smaller than a threshold, the clusters are merged. The threshold is defined as $avg_{cd} - 2 * std_{cd}$, where avg_{cd} is the average pairwise cosine distance between all legitimate clusters and std_{cd} is the standard deviation of that distance. This merging procedure is applied recursively until the minimum distance between the two closest clusters is larger than the threshold. After that, the merging proce-

[6]The threshold of 10 was derived from the procedure for manual labelling employed in the SemEval Task (Schlechtweg et al., 2020), where a constraint was enforced that the specific sense is attested at least 5 times in a specific time period in order to contribute word senses. We set the overall threshold of 10, which roughly translates to 5 per time period, since all of our test corpora (besides Aylien) contain two time periods.

dure is applied to illegitimate clusters (that contain less than 10 instances), using the same threshold. Illegitimate clusters could be added into one of the legitimate clusters or merged together to form a legitimate cluster with more than 10 instances. If there is no cluster that is close enough to be merged with, the illegitimate cluster is removed.

4.3 Change Detection and Interpretation

After the clustering procedure described above, for each word in each time slice, we extract its cluster distribution and normalise it by the word frequency in the time slice. Then target words are *ranked* according to the usage divergence between successive time slices, measured with the JSD or the WD[7]. If a ground-truth ranking exists, the method can be evaluated using the Spearman Rank Correlation to compare the true and the outputted ranking. In the exploratory scenario, the ranking is used to detect the most changing words and then investigate the most unevenly distributed clusters over time for the interpretation of the change.

JSD has been used for semantic shift detection in several recent papers, e.g. (Martinc et al., 2020a; Giulianelli et al., 2020; Kutuzov and Giulianelli, 2020). Since this is the first paper applying WD for this purpose, we describe it in more details.

The motivation for using the WD (Solomon, 2018) is to take into account the position of the clusters in the semantic space when comparing them. The JSD leverages semantic information encoded in the embeddings indirectly, distilled into two time-specific cluster distributions that JSD receives as an input. In addition to cluster distributions, WD accesses characteristics of the semantic space explicitly, through a matrix of cluster averages (obtained by averaging embeddings in each cluster) of size $T \times k \times 768$, where k is a number of clusters, T is a number of time slices and 768 is the embedding dimension.

This setup is a classical problem that can be solved using optimal transport (Peyré et al., 2019). We denote with μ_1 and μ_2 the sets of k average embedding points in the two vector spaces, and with c_1 and c_2 the associated clusters distributions. Thus, c_1 and c_2 are histograms on the simplex (positive and sum to 1) that represent the weights of each embedding in the source (μ_1) and target (μ_2) distributions. The task is to quantify the effort of moving one unit of mass from μ_1 to μ_2 using a chosen cost function, in our case the cosine distance. It is solved by looking for the transport plan γ, which is the minimal effort required to reconfigure c_1's mass distribution into that of c_2. The WD is the sum of all travels that have to be made to solve the problem:

$$\mathrm{WD}(c_1, c_2) = \min_{\gamma} \sum_{i,j} \gamma_{i,j} M_{i,j}$$
$$\text{with } \gamma 1 = c_1;\ \gamma^\intercal 1 = c_2;\ \gamma \geq 0$$

Where $M \in \mathbb{R}^{+}_{m \times n}$ is the cost matrix defining the cost to move mass from μ_1 to μ_2. We use the cosine similarity s, with $M = 1 - s(\mu_1, \mu_2)$.

Interpretation. Once the most changing words are detected, the next step is to understand *how* they change between two time slices by interpreting their clusters of usages.

Cluster distributions can be used directly to identify the clusters that are unevenly distributed across a time dimension. However, a cluster itself may consist of several hundreds or thousands of word usages, i.e. sentences. Interpreting the underlying sense behind each cluster by manually looking at the sentences is time-consuming. To reduce human work, we extract the most discriminating words and bigrams for each cluster: by considering a cluster as a single document and all clusters as a corpus, we compute the term frequency - inverse document frequency (tf-idf) score of each word and bigram in each cluster. The stopwords and the words appearing in more than 80% of the clusters are excluded to ensure that the selected keywords are the most discriminant. Thus, a ranked list of keywords for each cluster is obtained and top-ranked keywords are used for the interpretation of the cluster.

5 Evaluation

We use six existing manually annotated datasets for evaluation. The first dataset, proposed by Gulordava and Baroni (2011), consists of 100 English words labelled by five annotators according to the level of semantic change between the 1960s and 1990s[8]. To build the dataset, the annotators evaluated semantic change using their intuition, without looking at the context. This procedure is problematic since an annotator may forget or not be aware of a particular sense of the word.

[7] Using the POT package https://pythonot.github.io/.

[8] In order to make the proposed approach comparable to previous work, we remove four words that do not appear in the BERT vocabulary from the evaluation dataset, same as in Martinc et al. (2020a).

	COHA	SE English	SE Latin	SE German	SE Swedish	DURel	Avg. all
METHODS NOT USING CLUSTERING							
SGNS + OP + CD	0.347	0.321	0.372	**0.712**	**0.631**	**0.814**	**0.533**
Nearest Neighbors	0.310	0.150	0.273	0.627	0.404	0.590	0.392
Averaging	0.349	0.315	**0.496**	0.565	0.212	0.656	0.432
NON-SCALABLE CLUSTERING METHODS							
k-means 5 JSD	0.508	0.189	0.324	0.528	0.238	0.560	0.391
aff-prop JSD	**0.510**	0.313	0.467	0.436	-0.026	0.542	0.374
INTERPRETABLE SCALABLE METHODS							
Without filtering or merging of clusters							
k-means 5 JSD	0.430	0.316	0.358	0.508	0.073	0.658	0.390
aff-prop JSD	0.394	0.371	0.346	0.498	0.012	0.512	0.355
k-means 5 WD	0.372	0.360	0.450	0.514	0.316	0.607	0.437
aff-prop WD	0.369	**0.456**	0.397	0.421	0.264	0.484	0.399
With filtering and merging of clusters							
k-means 5 JSD	0.448	0.318	0.374	0.519	0.073	0.649	0.397
aff-prop JSD	0.403	0.348	0.408	0.583	0.018	0.712	0.412
k-means 5 WD	0.382	0.375	0.466	0.520	0.332	0.628	0.451
aff-prop WD	0.352	0.437	0.488	0.561	0.321	0.686	0.474

Table 1: Spearman Rank Correlation between system output rankings and ground truth rankings for various datasets. "SE" stands for SemEval.

The organizers of the recent SemEval-2020 Task 1— Unsupervised Lexical Semantic Change Detection (Schlechtweg et al., 2020)—employed another approach: the annotators had to decide whether a pair of sentences from different time periods convey the same meaning of the word (Schlechtweg and Schulte im Walde, 2020). For each of the four languages—German, English, Latin and Swedish—senses were manually annotated by labeling word senses in a pair of sentences drawn from different time periods. All SemEval-2020 Task 1 corpora contain only two periods and the sentences are shuffled and lemmatized. The lexical semantic change score is defined as the difference between word sense frequency distributions in the two time periods and measured by the Jensen-Shannon Distance (Lin, 2006).

The DURel dataset (Schlechtweg et al., 2018) is composed of 22 German words, ranked by semantic change by five annotators between two time periods, 1750–1799 and 1850–1899. Similarly to SemEval, the ranking was build by evaluating the relatedness of pairs of sentences from two periods.

In order to conduct usage change detection on the target words proposed by Gulordava and Baroni (2011), we fine-tune the English BERT-base-uncased model and generate contextual embeddings on the Corpus of Historical American English (COHA)[9]. We only use data from the 1960s to the 1990s (1960s has around 2.8M and 1990s 3.3M words), to match the manually annotated data. For the SemEval Task 1 evaluation set, we fine-tune the BERT models and generate contextual embeddings on the four corpora provided by the organizers of the task, English (about 13.4M words), German (142M words), Swedish (182M words) and Latin (11.2M words). Finally, we fine-tune BERT and generate embeddings on the German DTA corpus (1750–1799 period has about 25M and 1850–1899 has 38M tokens)[10].

The results are shown in Table 1. We compare our scalable approach with the *non-scalable clustering* methods used by Giulianelli et al. (2020) and Martinc et al. (2020a). *Averaging* (Martinc et al., 2020b) is the less interpretable method described in Section 3. *SGNS + OP + CD* (Schlechtweg et al., 2019) refers to the state-of-the-art semantic change detection method employing non-contextual word embeddings: the Skip-Gram with Negative Sampling (SGNS) model is trained on two periods independently and aligned using Orthogonal Procrustes (OP). Cosine Distance (CD) is used to compute the semantic change. The *Nearest Neighbors* method (Gonen et al., 2020) also uses SGNS embeddings.

[9] https://www.english-corpora.org/coha/

[10] https://www.ims.uni-stuttgart.de/en/research/resources/experiment-data/durel/

For each period, a word is represented by its top nearest neighbors (NN) according to CD. Semantic change is measured as the size of the intersection between the NN lists of two periods.

On average, the proposed scalable clustering with filtering and merging of clusters leads to a higher correlation with gold standard than the standard non-scalable clustering methods: the best method (aff-prop WD) achieving a Spearman correlation with the gold standard of 0.474 compared to the best non-scalable k-means 5 JSD achieving the Spearman correlation of 0.391. The method also outperforms averaging and NN, though it is outperformed by a large margin by the SGNS+OP+CD, achieving the score of 0.533.

The best performing clustering algorithm differs for different datasets. On average, affinity propagation only outperforms k-means when filtering and merging of clusters is employed. The effect of the filtering on k-means is positive on average but the difference is thin, as the number of clusters is low.

WD leads to better results than JSD on most of the corpora where averaging outperforms clustering, the only exception is DURel. An extreme example is the Swedish SemEval dataset, where the clustering with JSD performs particularly poorly: using the WD, which takes into account the average embeddings on top of cluster distributions, greatly increases the correlation with the gold standard. On the contrary, on COHA where averaging performs poorly in comparison to clustering, WD is under-performing.

6 Use Case: Aylien COVID-19 Corpus

The combination of scalable clustering with the interpretation pipeline opens new opportunities for diachronic corpus exploration. In this section, we demonstrate how it could be used to analyze the Aylien Coronavirus News Dataset. The corpus contains about 500k news articles related to COVID-19 from January to April 2020[11], unevenly distributed over the months (160M words in March, 41M in February, 35M in April and 10M in January). We split the corpus into monthly chunks and apply our scalable word usage change detection method.

6.1 Identification of the Top Drifting Words

The scalable method allows to perform embeddings extraction and clustering for all words in the corpus.

[11]We used an older version of the corpus. Currently the data from May are also available.

1	diamond	6	tag
2	king	7	paramount
3	ash	8	lynch
4	palm	9	developers
5	fund	10	morris

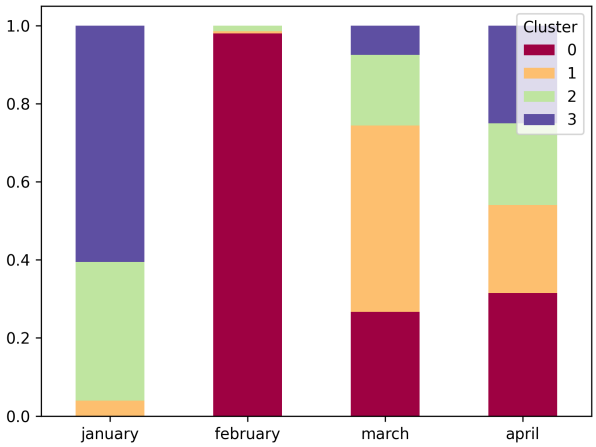
Table 2: Top 10 most changed words in the corpus according to a monthly-averaged WD of k-means ($k = 5$) cluster distributions.

We extract the top words with the highest average WD between the successive months to conduct a deeper analysis. We exclude words that appear less than 50 times in each month to avoid spurious drifts due to words having too few occurrences in a time slice. However, some drifts due to corpus artefacts remain, in particular dates such as *'2019-20'*. Thus, words containing numbers and one-letter words are also removed.

In Table 2 we present the top 10 most drifting words extracted using k-means with k=5 and ranked according to the average WD across the four months[12]. Among them, the word *diamond* is related to the cruise ship "Diamond Princess", which suffered from an outbreak of COVID-19 and was quarantined for several weeks. The word *king*, which is the second most changing word, is related to the King county, Washington, where the first confirmed COVID-19 related death in the USA appeared, and to the Netflix show "Tiger King", which was released in March. Thus, the primary context for this word changed several times, which is reflected in our results. Other words are mostly constituent words in named entities, related e.g., to an American Society of Hematology (ASH) Research Collaborative's Data Hub, which is capturing data on subjects tested positive for COVID-19.

The results suggest that the model does what it is meant to do: for most words in the list it is possible to find an explanation why its usage changed during the beginning of 2020. The list contains many proper names or proper name constituents, which could be either desirable or undesirable property, depending on research goals. Some work focuses specifically on proper names (Hennig and Wilson, 2020), since they could be a good proxy to shifts in socio-political situations. On the other hand, if

[12]This is a rather arbitrary procedure: one can imagine that a domain expert would prefer a different frequency threshold or focus more on a given month. The most time-consuming part is embedding extraction. Once this is done, clustering and keyword extraction can be done as many times as necessary.



#	Keywords
0	diamond princess, cruise ship, princess cruise, japanese, tested positive, confirm, ship diamond
1	neil diamond, comic, sweet caroline, trump, song, diamond said, comic book,
2	diamond hill, hill capital, diamond jubilee, diamond mountain, league postponed, portfolio, athletics
3	diamond industry, black diamond, jewellery, hong kong, diamond ring, surat diamond, india

Figure 1: Cluster distributions per month and top keywords for each cluster for word *diamond*.

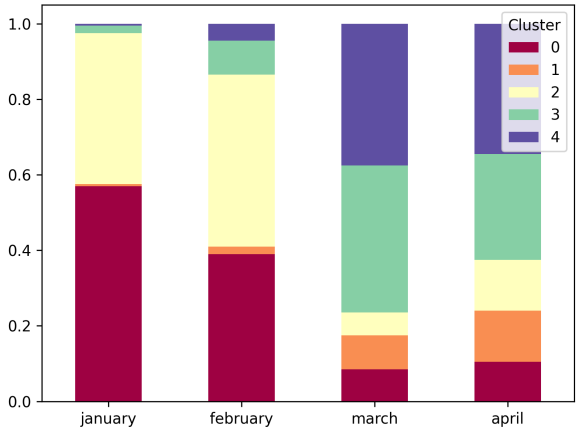
the focus of the study are shifts in more abstract concepts, then proper names could be filtered out before the embedding generation stage by employing named entity recognition tools.

6.2 Interpretation of the Usage Change

The interpretation pipeline, described in Section 4.3, is illustrated in figures 1 and 2. We focus on two words, *diamond* and *strain*, to show the various phenomena that can be detected. *Diamond* is the top drifting word in the entire vocabulary (see Table 2); it can be both a common noun and an entity, inducing usage drift when the entity appears in the newspapers after events with high media coverage. *Strain* is the 38th word with the highest drift overall, and the 15th highest between February and March 2020. It has several different senses whose usage vary across time following the events in the news. We cluster their vector representations from the Aylien corpus using k-means with $k = 5$ and apply the cluster filtering and merging step. Then, using tf-idf on unigrams and bigrams, we extract a set of keywords for each cluster to interpret the variations of their distribution.

The keywords and cluster distributions for the word *diamond* can be found in Figure 1. One of the clusters was removed at the filtering step, as it had less than 10 embeddings inside, and no other cluster was close enough. A clear temporal tendency is visible from the cluster distribution in Figure 1: a new major usage appears in February, corresponding to the event of the quarantined cruise ship (Cluster 0); this association is revealed by the keywords for this cluster. Moreover, the WD between January and February, when the outbreak happened, is 0.337; it is also very high between February and March (0.342). It reflects the large gap between the cluster distributions, first with the appearance of Cluster 0 in February that made the other usages of the word *diamond* in the media almost disappear, and then the reappearance of other usages in March, when the situation around the cruise ship gradually normalized. Cluster 1, that appears in March, is related to Neil Diamond's coronavirus parody of the song "Sweet Caroline" which was shared mid-March on the social media platforms and received a lot of attention in the US. Cluster 3 is related to the diamond industry; it is much less discussed as soon as the pandemic breaks out in February. Finally, Cluster 2 deals with several topics: Diamond Hill Capital, a US investment company, and the Wanda Diamond League, an international track and field athletic competition which saw most of its meetings postponed because of the pandemic. This last cluster shows the limitations of our clustering: it is complex to identify and differentiate all the usages of a word perfectly.

The keywords and cluster distributions for the word *strain* can be found in Figure 2. This is a polysemic word with two main senses in our corpus: as the variant of a virus or bacteria (biological term) and as "a severe or excessive demand on the strength, resources, or abilities of someone or something" (Oxford dictionary). Clusters 1, 3 and 4, which roughly match the second sense of the word (strain on healthcare systems in cluster 4, financial strain in cluster 3 and strain on resources and infrastructure in cluster 1), grow bigger across time, while clusters 0 and 2, which match the first sense of the word (e.g., new virus strain), shrink. This behavior underlines the evolution of the concerns related to the pandemic in the newspapers.



#	Keywords
0	strain coronavirus, new strain, city wuhan, novel strain, strain virus, chinese city
1	strain health, strain resources, stream, network infrastructure, international resources, likely strain
2	new strain, acute respiratory, 2019 ncov, respiratory syndrome, severe acute, identified humans
3	financial strain, feeling strain, strain coronavirus, economic strain, signs strain, strain said
4	ease strain, putting strain, strain health, reduce strain, care system, strain hospitals

Figure 2: Cluster distributions per month and top keywords for each cluster for word *strain*.

7 Conclusion

We proposed a scalable and interpretable method for word usage change detection, which outperforms the non-scalable contextual embeddings-based methods by a large margin. The new method also allows completely data-driven analysis of word sense dynamic in large corpora, which was impossible to conduct with unscalable methods. This opens new opportunities in both language change studies and text stream monitoring tasks. In this paper we focused on the latter application by analysing a large corpus of COVID-19 related news.

The method is outperformed by the state-of-the-art SGNS+OP+CD method. We hypothesise that this can be connected with the fact that the sentences in all but one evaluation corpus (COHA) are shuffled, meaning that BERT models cannot leverage the usual sequence of 512 tokens as a context, but are limited to the number of tokens in the sentence. We will explore this hypothesis in the future.

Despite achieving lower performance than the SGNS+OP+CD method, we nevertheless argue that our method offers a more fine-grained interpretation than methods based on non-contextual embeddings, since it accounts for the fact that words can have multiple meanings. The cluster-based technique returns a degree of change and a set of sentence clusters for each word in the corpus, roughly corresponding to word senses or particular usages. For this reason, the approach can be used for detection of new word usages and for tracing how these usages disappear, as we have shown in Section 6. Even more, word usages and their distributions over time could be linked with real-word events by labeling sentence clusters with a set of cluster-specific keywords.

Overall, we observe a large disparity between results on different evaluation corpora. This is in line with the results of the Semeval 2020 task 1 (Schlechtweg et al., 2020), where none of the best-performing methods was able to achieve the best result on all corpora. In practice, different methods focus on different aspects of word usage change: Averaging and SGNS+OP+CD focus on average variation of word usage, hiding the intra-period diversity. When it comes to clustering, JSD-based method detects the appearance or disappearance of a given usage, even a minor one. The WD-based method, using information from both the cluster distribution and the embeddings vectors, represents a compromise between the averaging and the JSD-based methods.

In this paper we follow the general approach in semantic shift detection literature and apply our analysis on the raw text. However, our results demonstrate that at least news monitoring applications would benefit from the application of the traditional text processing pipeline, in particular the extraction of named entities and dates. This will be addressed in the future work.

Acknowledgements

This work has been supported by the European Union Horizon 2020 research and innovation programme under grants 770299 (NewsEye) and 825153 (EMBEDDIA), the project Computer-assisted multilingual news discourse analysis with contextual embeddings (CANDAS, J6-2581), and Project Development of Slovene in the Digital Environment (RSDO).

References

Hosein Azarbonyad, Mostafa Dehghani, Kaspar Beelen, Alexandra Arkut, Maarten Marx, and Jaap Kamps. 2017. Words are malleable: Computing semantic shifts in political and media discourse. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1509–1518.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Haim Dubossarsky, Simon Hengchen, Nina Tahmasebi, and Dominik Schlechtweg. 2019. Time-out: Temporal referencing for robust modeling of lexical semantic change. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 457–470, Florence, Italy. Association for Computational Linguistics.

Haim Dubossarsky, Daphna Weinshall, and Eitan Grossman. 2017. Outta control: Laws of semantic change and inherent biases in word representation models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1136–1145. Association for Computational Linguistics.

Mario Giulianelli, Marco Del Tredici, and Raquel Fernández. 2020. Analysing lexical semantic change with contextualised word representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3960–3973, Online. Association for Computational Linguistics.

Hila Gonen, Ganesh Jawahar, Djamé Seddah, and Yoav Goldberg. 2020. Simple, interpretable and stable method for detecting words with usage change across corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 538–555, Online. Association for Computational Linguistics.

Kristina Gulordava and Marco Baroni. 2011. A distributional similarity approach to the detection of semantic change in the google books ngram corpus. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 67–71. Association for Computational Linguistics.

William L. Hamilton, Jure Leskovec, and Dan Jurafsky. 2016a. Cultural shift or linguistic drift? comparing two computational measures of semantic change. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2116–2121, Austin, Texas. Association for Computational Linguistics.

William L. Hamilton, Jure Leskovec, and Dan Jurafsky. 2016b. Diachronic word embeddings reveal statistical laws of semantic change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1489–1501.

Felix Hennig and Steven Wilson. 2020. Diachronic embeddings for people in the news. In *Proceedings of the Fourth Workshop on Natural Language Processing and Computational Social Science*, pages 173–183.

Yoon Kim, Yi-I Chiu, Kentaro Hanaki, Darshan Hegde, and Slav Petrov. 2014. Temporal analysis of language through neural language models. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, pages 61–65.

Andrey Kutuzov and Mario Giulianelli. 2020. UiO-UvA at SemEval-2020 task 1: Contextualised embeddings for lexical semantic change detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 126–134, Barcelona (online). International Committee for Computational Linguistics.

Andrey Kutuzov, Lilja Øvrelid, Terrence Szymanski, and Erik Velldal. 2018. Diachronic word embeddings and semantic shifts: a survey. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1384–1397. Association for Computational Linguistics.

Andrey Kutuzov, Erik Velldal, and Lilja Øvrelid. 2017. Tracing armed conflicts with diachronic word embedding models. In *Proceedings of the Events and Stories in the News Workshop*, pages 31–36, Vancouver, Canada. Association for Computational Linguistics.

J. Lin. 2006. Divergence measures based on the shannon entropy. *IEEE Trans. Inf. Theor.*, 37(1):145–151.

Stuart Lloyd. 1982. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137.

Matej Martinc, Syrielle Montariol, Elaine Zosa, and Lidia Pivovarova. 2020a. Capturing evolution in word usage: Just add more clusters? In *Companion Proceedings of the Web Conference 2020*, WWW '20, page 343–349, New York, NY, USA. Association for Computing Machinery.

Matej Martinc, Petra Kralj Novak, and Senja Pollak. 2020b. Leveraging contextual embeddings for detecting diachronic semantic shift. In *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, pages 4811—4819.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Gabriel Peyré, Marco Cuturi, et al. 2019. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607.

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, Florence, Italy. Association for Computational Linguistics.

Dominik Schlechtweg, Barbara McGillivray, Simon Hengchen, Haim Dubossarsky, and Nina Tahmasebi. 2020. SemEval-2020 task 1: Unsupervised lexical semantic change detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1–23, Barcelona (online). International Committee for Computational Linguistics.

Dominik Schlechtweg and Sabine Schulte im Walde. 2020. Simulating lexical semantic change from sense-annotated data. *CoRR*, abs/2001.03216.

Dominik Schlechtweg, Sabine Schulte im Walde, and Stefanie Eckmann. 2018. 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)*, pages 169–174, New Orleans, Louisiana. 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 EMNLP-IJCNLP 2019*, pages 66–76, Hong Kong, China. Association for Computational Linguistics.

Justin Solomon. 2018. Optimal transport on discrete domains.

Ian Stewart, Dustin Arendt, Eric Bell, and Svitlana Volkova. 2017. Measuring, predicting and visualizing short-term change in word representation and usage in VKontakte social network. In *Eleventh international AAAI conference on web and social media*.

Nina Tahmasebi, Lars Borin, and Adam Jatowt. 2018. Survey of computational approaches to diachronic conceptual change. *CoRR*, 1811.06278.

Xuri Tang. 2018. A state-of-the-art of semantic change computation. *Natural Language Engineering*, 24(5):649–676.

Zi Yin, Vin Sachidananda, and Balaji Prabhakar. 2018. The global anchor method for quantifying linguistic shifts and domain adaptation. In *Advances in neural information processing systems*, pages 9412–9423.

F. Manuscript: Grammatical Profiling for Semantic Change Detection

Grammatical Profiling for Semantic Change Detection

Mario Giulianelli* ILLC, University of Amsterdam m.giulianelli@uva.nl
Andrey Kutuzov* University of Oslo andreku@ifi.uio.no
Lidia Pivovarova* University of Helsinki first.last@helsinki.fi

Abstract

Semantics, morphology and syntax are strongly interdependent. However, the majority of computational methods for semantic change detection use distributional word representations which encode mostly semantics. We investigate an alternative method, grammatical profiling, based entirely on changes in the morphosyntactic behaviour of words. We demonstrate that it can be used for semantic change detection and even outperforms some distributional semantic methods. We present an in-depth qualitative and quantitative analysis of the predictions made by our grammatical profiling system, showing that they are plausible and interpretable.

1 Introduction

Lexical semantic change detection has recently become a well-represented field in NLP, with several shared tasks conducted for English, German, Latin and Swedish (Schlechtweg et al., 2020), Italian (Basile et al., 2020) and Russian (Kutuzov and Pivovarova, 2021a). The overwhelming majority of solutions employ either static word embeddings like word2vec (Mikolov et al., 2013) or more recent contextualised language models like ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019). These models build upon the distributional semantics hypothesis and can capture lexical meaning, at least to some extent (e.g., Iacobacci et al., 2016; Pilehvar and Camacho-Collados, 2019; Yenicelik et al., 2020). Thus, they are naturally equipped to model semantic change.

Yet it has long been known for linguists that semantics, morphology and syntax are strongly interrelated (Langacker, 1987; Hock and Joseph, 2019). Semantic change is consequently often accompanied by morphosyntactic drifts. Consider the English noun '*lass*': in the 20th century, its 'SWEETHEART' meaning became more dominant over the older sense of 'YOUNG WOMAN'. This was accompanied by a sharp decrease in plural usages ('*lasses*'), as shown in Figure 1.

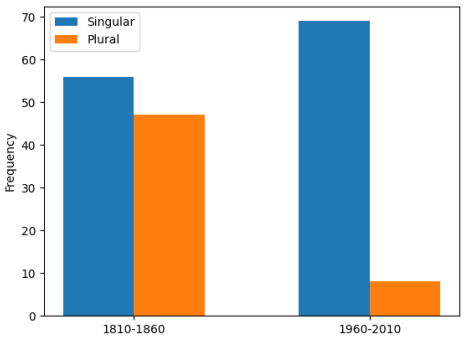


Figure 1: Changes in the number category distribution for the English noun '*lass*' over time, calculated on the English corpora of the SemEval 2020 shared task 1 (Schlechtweg et al., 2020). '*Lass*' is annotated as semantically changed in the SemEval dataset.

Exploiting distributions of *grammatical profiles*—i.e., morphological and syntactic features—to detect lexical semantic change is the focus of this paper. We investigate to what extent lexical semantic change can be detected using *only* morphosyntax. Our main hypothesis is that significant changes in the distribution of morphosyntactic categories can reveal useful information on the degree of the word's semantic change, even without help from any lexical or explicitly semantic features.

Due to the interdependence of semantics and morphosyntax, it is often difficult to determine which type of change occurred first, and whether it triggered the other. Establishing the correct causal direction is outside the scope of this study; it is sufficient for us to know that semantic and morphosyntactic changes often co-occur.

By proposing this functionalist approach to lexical semantic change detection, we are not aiming at establishing a new state-of-the-art. This

*Equal contribution, the authors listed alphabetically.

Proceedings of the 25th Conference on Computational Natural Language Learning (CoNLL), pages 423–434

is hardly possible without taking semantics into account. But what exactly *is* possible in such a functionalist setup?

We investigate this question experimentally[1] using standard semantic change datasets for English, German, Swedish, Latin, Italian and Russian. Our main findings are the following:

1. Tracing the changes in the distribution of dependency labels, number, case, tense and other morphosyntactic categories outperforms count-based distributional models. In many cases, prediction-based distributional models (static word embeddings) are outperformed as well. This holds across six languages and three different datasets.
2. Morphological and syntactic categories are complementary: combining them improves semantic change detection performance.
3. The categories most correlated with semantic change are language-dependent, with number being a good predictor cross-linguistically.
4. The predictions derived from grammatical profiling are usually interpretable (as in the '*lass*' example above), which is not always the case for methods from prior work based on word embeddings, either static or contextualised. This makes our method suitable for linguistic studies that require qualitative explanations.

2 Related work

Behavioural profiles were introduced in corpus linguistics by Hanks (1996) as the set of syntactic and lexical preferences of a word, revealed by studying a large concordance extracted from a corpus. The behavioural profile of a word consists of corpus counts of various linguistic properties, including morphological features, preferred types of clauses and phrases, collocates and their semantic types (Gries and Otani, 2010). Subtle distinctions in word meaning are reflected in behavioural profiles. Indeed this technique, which combines lexical and grammatical criteria for word sense distinction, was used to study synonymy and polysemy (Divjak and Gries, 2006; Gries and Divjak, 2009) as well as antonymy (Gries and Otani, 2010).

One of the theoretical roots for profiling is the theory of *lexical priming* (Hoey, 2005). According to this theory, words trigger a set of grammatical and lexical constraints, referred to as *primings* and stored in a mental concordance. The theory states that 'Drifts in priming ... provide a mechanism for temporary or permanent language change' (Hoey, 2005, p. 9), and since primings are thought to be organised in the mental concordance in the form of behavioural profiles (Gries and Otani, 2010), it is theoretically plausible that diachronic word meaning change is reflected in a change of behavioural profiles. As far as we are aware, this idea has not been further developed in corpus linguistics.

In spite of its theoretical validity, behavioural profiling as a practical data analysis technique has serious limitations. Profiles include a large variety of word properties and some of them, especially those related to semantics, cannot be easily extracted from a corpus automatically. Usually, a particular subset of word properties is selected based on researchers' intuition and background knowledge, and statistical tests are sometimes used for feature selection at later stages of the analysis (Divjak and Gries, 2006). Moreover, the variety of properties comprised in a behavioural profile makes statistical analysis difficult due to correlations between language phenomena of different levels and sparsity of the data (Kuznetsova, 2015, section 2.2.2). For these reasons, some studies (Janda and Lyashevskaya, 2011; Eckhoff and Janda, 2014) reduce a word's possibly very broad behavioural profile to a more compact *grammatical profile*, i.e. a set of preferred morphological forms for the word. These studies too, however, rely on an *a priori* selection of relevant morphological tags.

These technical difficulties may explain why profiling has not been used in computational approaches to lexical semantic change detection. Most attempts to tackle word meaning change in NLP are based on distributional patterns of *lexical* co-occurrences, starting from early count-based approaches (Juola, 2003; Hilpert and Gries, 2008), continuing with dimensionality reduction techniques (Gulordava and Baroni, 2011), and later accelerated by embeddings-based models (Kutuzov et al., 2018). More recently, contextualised embeddings were also applied to this task (Giulianelli et al., 2020; Montariol et al., 2021).

As far as we are aware, there is one exception to this trend: Ryzhova et al. (2021) employed grammatical profiles to detect the semantic change of Russian nouns. In their work, a profile of case and number frequency distributions is collected separately for each time period, and the degree of

[1] Our code is available at https://github.com/glnmario/semchange-profiling

semantic change is measured as the cosine distance between the two distributions. The results obtained with this method are close to the results yielded by word2vec embeddings, but lower than those of contextualised embeddings. Inspired by Ryzhova et al. (2021), we further investigate the ability of grammatical profiles to capture word meaning change. We propose a number of improvements and evaluate them on datasets in six different languages. Most importantly, we use *all* available morphological tags, without any manual pre-selection, and we conduct an in-depth analysis of our results to understand why grammatical profiling works for this task and what are its limitations.

3 Data and tasks

Following the standard evaluation approach adopted for automatic lexical semantic change detection, we cast the problem as either binary classification (Subtask 1, using the terminology of the SemEval 2020 Unsupervised Lexical Semantic Change Detection shared task (Schlechtweg et al., 2020)) or as a ranking task (Subtask 2). In Subtask 1, given a set of target words, a system must determine whether the words lost or gained any senses between two time periods. In Subtask 2, a system has to rank a set of target words according to the degree of their semantic change.

Annotating data for word meaning change detection is a non-trivial process because it requires taking into account numerous word occurrences from every time period of interest. The current practice adopted in the community is to annotate pairs of sentences containing a target word used either in the same or in a different sense; then pairwise scores are aggregated to obtain a final measure of change, either binary or continuous (Schlechtweg et al., 2018). This procedure has been used by organizers of three recent shared tasks: the SemEval 2020 Unsupervised Lexical Semantic Change Detection shared task (Schlechtweg et al., 2020), EvaLita (Basile et al., 2020) and RuShiftEval (Kutuzov and Pivovarova, 2021a). We use the data from these three shared tasks, allowing to compare our approach with the state-of-the-art results obtained by distributional models.

The SemEval dataset consists of target words in four languages—37 English, 48 German, 40 Latin, and 32 Swedish—that are manually annotated for both subtasks. The EvaLita dataset consists of 18 Italian words annotated for Subtask 1 only. Finally, the RuShiftEval dataset consists of 99 Russian nouns annotated for Subtask 2. All datasets are accompanied by diachronic corpora. Most of the corpora are split in two time periods, except for the RuShiftEval corpus, which is separated into three time bins: *Russian1* and *Russian2* are annotated with semantic shifts between the pre-Soviet and Soviet periods, and between the Soviet and post-Soviet periods respectively; *Russian3* is annotated with semantic shifts between the pre-Soviet and post-Soviet periods (Kutuzov and Pivovarova, 2021b).

In sum, we have at our disposal several dozens words from three Indo-European language groups: Italic, Germanic and Slavic. Though our results may not generalize to other language families or to other languages within the families analysed, these are the most diverse data that are currently available for this kind of study.

4 Methods

4.1 Basic procedure

To obtain grammatical profiles, the target historical corpora are first tagged and parsed with UDPipe (Straka and Straková, 2017).[2] Then we count the frequency of morphological and syntactic categories for each target word in both corpora. More precisely, we count the FEATS values of a corpus's CONLLU file and store the frequencies in two data structures—one for each time period. For example, `{'Number=Sing': 338, 'Number=Plur': 114}` is the morphological dictionary obtained for an English noun in a single time period. We store syntactic features in an additional dictionary, where keys correspond to the labels of the dependency arc from the target word to its syntactic head (as found in the DEPREL field of a CONLLU-formatted corpus).

For each target word and for both morphological and syntactic dictionaries, we create a list of features by taking the union of keys in the corresponding dictionaries for the two time bins. The feature list will be `['Number=Sing', 'Number=Plur']` for the example above. Then, we create feature vectors $\vec{x}_1$ and $\vec{x}_2$, where each dimension represents a grammatical category and the value it takes is the frequency of that category in the corresponding time period. If a feature does

[2] We use the following models: *english-lines-ud-2.5*, *german-gsd-ud-2.5*, *latin-proiel-ud-2.5*, *swedish-lines-ud-2.5*, *russian-syntagrus-ud-2.5*, *italian-isdt-ud-2.5*.

not occur in a time period, its value is set to 0. The resulting feature vectors represent grammatical profiles for a word in the corresponding periods. Since the feature list is produced separately for each word, the size of the vectors varies across words.

Finally, we compute the cosine distance $cos(\vec{x}_1, \vec{x}_2)$ between the vectors to quantify the change in the grammatical profiles of the target word. This is done separately for morphological and syntactic categories, yielding two distance scores d_{morph} and d_{synt}. They are used directly to rank words in Subtask 2: the larger is the distance, the stronger is the semantic change. To solve the binary classification task (Subtask 1), we classify the top n target words in the ranking as 'changed' (1) and the rest of the list as 'stable' (0). The value of n can be either set manually or inferred from the ranking using off-the-shelf algorithms of change point detection (Truong et al., 2020).

We also combine the scores obtained separately for morphological and syntactic tags by averaging d_{morph} and d_{synt} for each target word (rounding to the nearest integer in the case of binary classification) and then re-rank the words according to the resulting values. In the end, we have three solutions for each task: 'morphology', 'syntax' and 'averaged'. In the next subsections, we describe a number of improvements that we use to amend this basic procedure.

4.2 Filtering

To reduce noise that could be introduced due to rare word forms and possible tagging errors, we exclude rare grammatical categories from the analysis. A feature is filtered out from a feature vector $\vec{x}$ if the sum of the feature occurrences in the two time slices amounts to less than five percent of the total word usages. It is possible to optimise this threshold, but we do not tune any numerical parameters to avoid over-fitting to the target datasets.

4.3 Category separation

In the basic procedure described above, we extract exactly one morphological feature for each word occurrence; this type of morphological feature is a combination of morphological categories that exhaustively describes a word form. For example, this is an excerpt from a grammatical profile of the English verb '*circle*' in the 1810-1860 time period:

```
Tense=Pres|VerbForm=Part : 50
Mood=Ind|Tense=Past|VerbForm=Fin : 24
Tense=Past|VerbForm=Part|Voice=Pass : 17
VerbForm=Inf : 9
Mood=Ind|Tense=Pres|VerbForm=Fin : 1
Tense=Past|VerbForm=Part : 1
```

This representation is very sparse—some features appear only once in the corpus—and it conflates categories of different nature, such as verb form and tense. We therefore introduce a category separation step, where feature vectors are created separately for each morphological category. Thus, we transform a distribution of *word forms* into a distribution of *morphological categories* and obtain a denser and more meaningful representation:

```
Tense : {Past 42, Pres 51}
VerbForm : {Part 68, Fin 25, Inf 9}
Mood : {Ind 25}
Voice : {Pass : 17}
```

Then cosine distance is computed for each category separately. In the example above, we obtain separate distance values for `Tense`, `VerbForm`, `Mood`, and `Voice`; the number of distances differs across words and languages. We take the maximum distance value as the final change score, assuming that a significant change in the distribution of a single category indicates semantic change, regardless of the other categories.[3]

When separation is combined with filtering, filtering is performed *after* feature separation to preserve maximum information. Continuing with the previous example: in the basic procedure, the word form `Tense=Past|VerbForm=Part` is filtered out, as it appears once in the first corpus and it is rare in the second corpus as well. In the category separation strategy this form is taken into account, separately contributing to the `Tense` and `VerbForm` distances.

4.4 Combination of morphology and syntax

Category separation opens new possibilities for taking syntactic categories into account. We can average morphological and syntactic distances, as in our basic procedure, or append the syntactic distance value to the array of morphological distances, and then choose the maximum. In the first strategy, morphological and syntactic rankings are weighted equally regardless of the number of morphological categories for a given word. In the second strategy,

[3] We also experimented with averaging category distances. This improves the results compared to using categories without separation, but it is not as effective as taking the maximum.

syntactic labels are weighted down depending on the richness of the morphological profile.

5 Results

We evaluate our method on both subtasks of the SemEval 2020 Unsupervised Lexical Semantic Change Detection shared task (Schlechtweg et al., 2020). As described in Section 3, Subtask 1 is a binary classification task, evaluated with accuracy. Subtask 2 is a ranking task, evaluated with Spearman's rank correlation.

Basic procedure Using only morphological features, we obtain an average correlation of 0.181 across the four SemEval languages, as can be seen in Table 1. Syntactic features yield a +0.017 increase, and after averaging d_{morph} and d_{synt} (see Section 4.1) we reach a correlation score of 0.208. This is already substantially higher than the SemEval baseline which employed count-based distributional models (see Table 1).

Frequency threshold Filtering out rare features as described in Section 4.2 has a small but positive impact on all three setups: +0.011 for morphological features, +0.033 for syntactic features, and +0.065 for the combination of the two.

Category separation Measuring distance between morphological categories separately (see Section 4.3) produces an additional significant boost: we obtain a correlation score of 0.278 using these refined morphological representations. In combination with syntactic features (Section 4.4), this approach yields an average correlation of 0.369 with human judgements. This is our best result on Subtask 2, more than twice higher than a correlation obtained by the SemEval count-based baseline (see Table 1); for Latin, a language with rich morphology, grammatical profiles actually outperform even the *best* SemEval 2020 submission. These scores are particularly impressive given that, unlike those based on distributional vectors, our method has no access to lexical semantic information.

As can be seen in Table 1, our category separation approach does not extend well to the Russian test sets, obtaining an average correlation score of 0.130.[4] A possible explanation for the lower correlation may be related to smaller distances between Russian time bins as compared to the SemEval setup: *Russian1* and *Russian2* are annotated with semantic shifts between pre-Soviet and Soviet and between Soviet and post-Soviet periods respectively, while *Russian3* measures the change between pre-Soviet and post-Soviet periods, with a significant time gap in between. Indeed we obtain much higher scores on *Russian3*. In addition, the annotation procedures for the RuShiftEval dataset differ in some details from those for SemEval'20.

Another observation is that morphological category separation does not improve results for English. The best method for English relies only on syntactic features. The most plausible explanation is that English morphology is rather poor and it tends to mark grammatical categories with separate words. Our method can be potentially improved by taking into account multi-word forms, e.g. to determine English verb mood.

Subtask 1 Following our basic procedure (Section 4.1), we assign a classification score of 1 to the top 43% of the target words[5] for each language, ranked according to their grammatical profile changes. This yields an accuracy close to that of the SemEval count-based baseline (see Table 2).[6] Filtering rare features hardly yields any improvement here, but once combined with morphological category separation and automatic change point detection it produces an accuracy of 0.603. We also observe that using change point detection with dynamic programming (Truong et al., 2020) does not cause any significant accuracy decrease in comparison to using the hard-coded 43% ratio, showing that our method does not require knowledge of the test data distribution. On the Italian test set, we correctly classify 3 more words (out of 18) than the collocation-based baseline (Basile et al., 2019b), obtaining an accuracy of 0.778.

6 Qualitative analysis

In Section 5, we showed that grammatical profiling alone can detect a word meaning change better than count-based distributional semantic models which exploit lexical co-occurrence statistics. This is a remarkable finding: it confirms that meaning change leaves traces in grammatical profiles and it demonstrates that these traces can be used as effective predictors of a word's meaning stability. In this Section, to better understand when change in grammatical profiles is a good indicator of lexical

[4] At the same time, in the basic procedure, morphological features yield a much higher correlation score of 0.225.

[5] Average ratio of changed words across SemEval datasets.

[6] Note that the SemEval'20 count baseline also uses a manually defined threshold value in Subtask 1.

Categories	SemEval 2020 languages					Russian			
	English	German	Latin	Swedish	Mean	Russian1	Russian2	Russian3	Mean
	Basic procedure								
Morphology	0.234	0.043	0.241	0.207	0.181	**0.137**	0.210	**0.327**	**0.225**
Syntax	0.319	0.163	0.328	-0.017	0.198	0.060	0.101	0.269	0.143
Average	0.293	0.147	0.304	0.088	0.208	0.101	0.191	0.294	0.195
	5% filtering								
Morphology	0.211	0.080	0.285	0.191	0.192	0.127	0.185	0.264	0.192
Syntax	**0.331**	0.146	0.265	0.184	0.231	0.056	0.111	0.279	0.149
Average	0.315	0.171	0.345	0.263	0.273	0.094	0.183	0.278	0.185
	Category separation and 5% filtering								
Morphology	0.218	0.074	0.519	0.303	0.278	0.028	**0.241**	0.293	0.187
Average	0.321	0.227	0.523	**0.381**	0.363	0.002	0.179	0.278	0.153
Combination	0.320	**0.298**	**0.525**	0.334	**0.369**	0.000	0.149	0.242	0.130
	Prior SemEval results					**Prior RuShiftEval results***			
Count baseline	0.022	0.216	0.359	-0.022	0.144	0.314	0.302	0.381	0.332
Best shared task system	0.422	0.725	0.412	0.547	0.527	0.798	0.803	0.822	0.807
(Ryzhova et al., 2021)	-	-	-	-	-	0.157	0.199	0.343	0.233

Table 1: Performance in graded change detection (SemEval'20 Subtask 2 and RuShiftEval), Spearman rank correlation coefficients. Note that RuShiftEval features three test sets for three different time period pairs.
*The RuShiftEval baseline relies on CBOW word embeddings and their local neighborhood similarity. (Ryzhova et al., 2021) used an ensemble method with much higher performance, we report the results obtained solely with profiling. While SemEval results are fully unsupervised, the best RuShiftEval results are supervised and not directly comparable to our setting.

Categories	English	German	Latin	Swedish	Mean	Italian
	Basic procedure					
Morphology	0.595	0.521	0.525	0.581	0.555	0.722
Syntax	0.541	**0.646**	0.575	0.645	0.602	0.611
Average	0.568	0.583	0.475	**0.710**	0.584	0.722
	Automatic change point detection					
Morphology	**0.622**	0.479	**0.625**	0.548	0.569	0.722
Syntax	0.514	0.625	0.500	0.677	0.579	0.611
Average	0.595	0.542	0.525	0.677	0.585	**0.778**
	Category separation, change point detection and 5% filtering					
Morphology	0.622	0.583	**0.625**	0.581	**0.603**	0.500
Average	0.595	0.625	0.450	**0.710**	0.595	0.667
Combination	0.541	0.583	0.575	0.645	0.586	0.500
	Prior SemEval results					**Prior EvaLita results***
Baseline	0.595	0.688	0.525	0.645	0.613	0.611
Best shared task system	0.622	0.750	0.700	0.677	0.687	0.944

Table 2: Performance in binary change detection (SemEval'20 Subtask 1 and EvaLita), accuracy. Note that in this paper we mostly focus on ranking (Subtask 2). All the binary change detection methods here are entirely based on the scores produces by the ranking methods.
*The Italian baseline relies on collocations (Basile et al., 2019a): for each target word, two vector representations are built, with the Bag-of-Collocations related to the two different time periods. Then, the cosine similarity between them is computed.

semantic change, we analyse the characteristics of the target words to which our method assigns the most and least accurate rankings.

6.1 When is grammatical profiling enough?

We begin by analysing the most accurately ranked words (see Appendix A). The Italian word '*lucciola*', for example, is ranked 1st out of 18 by our method due to the singular usages of the word disappearing after 1990. The singular usage is indeed much more likely for the dying sense of the word (an euphemism for 'PROSTITUTE'), whereas the plural form '*lucciole*' is more likely used for the stable sense of the word ('FIREFLIES') or in the idiomatic expression *prendere lucciole per lanterne* (*getting the wrong end of the stick*), which makes up for most of the occurrences between 1990 and 2014. Another example of correctly identified semantically shifted words is the Latin '*imperator*' (ranked 1st out of 40). In the second time period—ranging from 0 to 2000 A.D.—nominative usages become predominant. A possible explanation for this change is that the more frequent agentive usages of the word correspond to the new role of the 'EMPEROR' in the imperial Rome (27 B.C. to A.D. 476) rather than that of a generic 'COMMANDER'—the older sense of the word.[7]

For English, the noun '*stab*' is ranked 4th out of 37, mostly because of syntactic changes: 27% of its occurrences in the 20th century are used as oblique arguments, compared to only 13% in the 19th century. This is arguably associated with the emergent sense of 'SUDDEN SHARP FEELING' ('*...left me with a sharp stab of sadness*'). The German word '*artikulieren*' correctly receives a high rank (9th out of 48): it occurs only 3 times in the 19th century and 210 times between 1946 and 1990, shifting towards a much richer grammatical profile. Sharp changes in frequency are reflected in the diversity of grammatical profiles and can also help detect lexical semantic change.

Our qualitative analysis reveals that the successful examples are often cases of broadening and narrowing of word meaning. These kinds of semantic change seem to be easily picked with profiling. However, some examples of broadening and narrowing fail to be detected, as will be shown in Section 6.2, especially if they involve metaphorical extensions of word meaning. A consistent characterisation of the kinds of semantic change detected and overlooked by our method would require diachronic corpora where both the degree and the type of semantic changes are annotated.

6.2 When it is not enough?

Although it largely outperforms simple distributional semantic models, our grammatical profiling approach is still not on par with state-of-the-art semantics-based algorithms. To find out when changes in morphosyntactic profiles are not sufficient to detect a word's meaning change, we analyse *false positives* and *false negatives*: i.e., target words that are assigned an erroneously high or low semantic change score, respectively.

False positives are words whose change in grammatical profile does not correspond to semantic change. An example of a false positive is the Italian word '*cappuccio*' ('HOOD'). The increase from 9% to 41% of plural usages causes our method to assign this word a relatively high change score—6th out of 18 (6 words are annotated as changing in the Italian dataset). Inspecting the Italian corpora, we notice that between 1945 and 1970 the word is mainly used to describe the pointed hood of the robes typically worn by Ku Klux Klan members; after 1990, the word's context of usage becomes much less narrow. The meaning of the word, however, does not change. This type of errors is, at least to a certain extent, an artifact of the source data: grammatical profiles are less accurate when the set of domains covered by a corpus is limited.

Another type of false positives is also partially related to corpus imbalance. We have seen in the previous section that sharp frequency increases correspond to significant changes in grammatical profiles, and that this information can be exploited by our method to detect changing words. However, frequency change can be an unfaithful indicator of meaning change. This is the case, for example, for the German words '*Lyzeum*' ('LYCEUM'; ranked 1st out of 48), and '*Truppenteil*' (a 'UNIT OF TROOPS'; ranked 11th), and for the Latin word '*jus*' (a 'RIGHT', the 'LAW'; ranked 4th out of 40).

False negatives, on the other hand, are words whose semantic change is not reflected in changes in grammatical profile. The German word '*ausspannen*' ('TO REMOVE', 'TO UNCLAMP') is used across the 19th and 20th century only in its infini-

[7] We are aware that the current separation of the Latin corpus into two time periods can be controversial. Still, we follow the splits defined by the SemEval 2020 organisers (Schlechtweg et al., 2020) for consistency and comparability with prior work.

tive form, so our method assigns it a relatively low change score (23rd out of 48). Most of the occurrences in the 19th century, however, are literal usages of the word (e.g., *die Pferde ausspannen*, *to unhitch the horses*), whereas in the (second part of the) 20th century the novel metaphorical usage of the word (e.g., *für fünf Minuten ausspannen*, *to relax for five minutes*) is the most frequent one. Another example of a German word whose novel metaphorical sense remains undetected (ranked 31st) is '*Ohrwurm*' ('EARWORM'): the grammatical profile of this word remains stable (except for the accusative case becoming slighlty more frequent), but the word acquires the meaning of *catchy song*, or *haunting melody*. Similarly, the singular usages of the Latin word '*pontifex*' increase from 63% to 83%, signalling the semantic narrowing of the word occurred in medieval Latin (from a 'BISHOP' to the 'POPE'), but the case distribution remains similar; this results in a rather low change score (ranked 22nd out of 40). The last two examples show that taking the maximum distance across categories (see 4.3) is a correct strategy, yet sometimes the changes in that grammatical category are still insufficient for our method to detect change.

7 Category importance

In this Section, we conduct an additional experiment to find out which grammatical categories are most related to semantic change. To this end, we train logistic regression classifiers for binary classification using English, German, Latin, Swedish and Italian data. The classifier features are cosine distances between frequency vectors of each particular category from different time bins. Before fitting the classifier, each feature is independently zero-centered and scaled to the unit variance. Then, regression coefficients are estimated for each feature: we consider positive weights as an indication of usefulness of a feature for classification. The outcome of this analysis is shown in Table 3. We list English nouns and verbs separately since the SemEval'20 dataset explicitly annotates part-of-speech tags for the English target words. This is not the case for the other languages in this dataset.

In line with the results presented in Section 5, Swedish and Italian classifiers yield the highest accuracy and F-score. Latin, a highly inflectional language, has by far the largest set of categories contributing positively to semantic change detection (interestingly, excluding syntax). English, a highly analytical language, is on the other end of the spectrum.

Language	Top categories	Accur.	F1
English nouns	number	0.576	0.523
English verbs	verb form, syntax	0.750	0.733
German	number, syntax, gender	0.542	0.541
Swedish	syntax, mood, voice, definiteness, number	0.839	0.797
Latin	voice, number, degree, case, gender, mood, aspect, person, tense	0.650	0.649
Italian	number, tense, syntax	0.778	0.723

Table 3: Categories with positive weights in binary classifiers of semantic change (logistic regression). 'Syntax' stands for dependency relation to the syntactic head of the word. Evaluation scores are calculated on the train data, F1 is macro-averaged.

Additionally, we estimate the relative importance of morphosyntactic categories by calculating the Spearman's rank-correlation of their respective cosine distance values (across all target words) with the gold semantic change rankings. In other words, we single out each category, e.g. verbal mood, and test whether diachronic change in its frequency distribution is correlated with manually annotated semantic change scores.

In Table 4, we show the categories with statistically significant ($p < 0.05$) correlations for each language and dataset. In English, as expected given its analytical nature, only changes in syntactic roles yield such a correlation; other categories are either non-existent in this language, or are not linked to semantic change strongly enough. For an inflection language such as Latin, number and adjectival degree are highly predictive (the latter is arguably because Latin has the highest ratio of adjectives among all SemEval 2020 Task 1 datasets: about 20%). Not surprisingly for a synthetic language, the morphological categories of number and case show strong correlations for Russian. In the case of the larger time gap between pre-Soviet and post-Soviet periods (Russian 3), syntactic relationships also become a good predictor.

What *is* surprising, however, is that changes in gender are also correlated with semantic change in the Russian case. This result is hard to inter-

	Number	Mood	Degree	Gender	Case	Syntax
English	-	-	-	-	-	0.331
German	-	-	-	-	-	-
Latin	0.304	-	0.301	-	-	-
Swedish	0.402	0.397	-	-	-	-
Russian 1	-	-	-	0.218	0.196	-
Russian 2	-	-	-	0.231	0.324	-
Russian 3	0.246	-	-	0.218	0.327	0.279

Table 4: Spearman rank correlations between diachronic grammatical profile distances for different categories and manually annotated semantic change estimations. '-' stands for no significant correlation.

pret, since grammatical gender is a lexical feature of Russian nouns and does not change from occurrence to occurrence; even diachronically, such cases are quite rare. The reason for this is slightly erroneous morphological tagging: our tagger mixes up homographic inflected forms, which abound in Russian, and assigns feminine gender to masculine nouns, and vice versa. The reliance on the tagger performance can be seen as a limitation of our grammatical profiling approach. However, the existence of the correlation hints that these errors are not entirely random, and their frequency is influenced by word usage: gender is ambiguous only in certain case and number combinations, and the frequency of these combinations seems to change diachronically. For example, for the form '*cheki*' ('cheques/grenade pin'), the masculine lemma licenses the accusative plural reading, while the feminine lemma licenses the genitive singular reading. Thus, even the tagger errors are in fact informative.

Interestingly, for German, no single category changes are significantly correlated with semantic change. This is in line with our weak—although still higher than the count-based baseline—results for German described above, but is somewhat surprising, given the fusional nature of the language, with its rich spectrum of inflected word forms.[8] Some peculiarities of the employed tagger model might be responsible for this finding, which should be further tested and explained in future work.

8 Conclusion

Semantic change is inextricably tied to changes in the distribution of morphosyntactic properties of words, i.e. their grammatical profiles. In this paper, we showed that tracking these changes is enough to build a semantic change detection system which, without access to any lexical semantic information, consistently outperforms count-based distributional semantic approaches to the task. Grammatical profiling yields surprisingly good evaluation scores across different languages and datasets, without any language-specific tuning. For Latin, a language with rich morphology, our methods even establish a new SOTA in Subtask 2 of SemEval'20 Task 1.

These results indicate that grammatical profiling cannot compete with state-of-the-art methods based on large pre-trained language models, since they have the potential to encode both semantics and grammar. Yet reaching the highest possible scores on the task was not our goal. Instead, the aim of our study was to demonstrate that more attention should be paid to the relation between morphosyntax and semantic change. Whether morphosyntactic and semantic features are complementary and can be successfully combined is a interesting question to be addressed in future work.

We performed an extensive quantitative and qualitative analysis of our semantic change detection methods, showing that profiling yields interpretable results across several languages. Nevertheless, we still lack an understanding of some aspects of the interaction between semantics and morphosyntax. Finding the reasons behind the relatively poor performance on some datasets, e.g. German, is an important direction for future studies.

Another interesting question is how to incorporate full dependency trees into grammatical profiles, rather than only dependency relations to the syntactic head of a word. This is particularly important for analytical languages, where grammatical markers are presented in more than one word, such as with English verb mood and aspect. Moreover, dependency structure can be crucial for languages from families other than the Indo-European, e.g. to take into account detached counters in Japanese or plural markers in Yoruba.

In light of our experimental results, we argue that grammatical profiling should become one of the standard baselines for semantic change detection.

Acknowledgements

We thank the anonymous CoNLL-2021 reviewers for their helpful comments. This work has been partly supported by the European Union's Horizon 2020 research and innovation programme under grants 770299 (NewsEye), 825153 (EMBEDDIA), and 819455 (DREAM).

[8] We computed correlations for German nouns and verbs separately, but did not find any significant correlation either.

References

Pierpaolo Basile, Annalina Caputo, Tommaso Caselli, Pierluigi Cassotti, and Rossella Varvara. 2020. DIACR-Ita @ EVALITA2020: Overview of the EVALITA2020 Diachronic Lexical Semantics (DIACR-Ita) Task. *Proceedings of the Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020) CEUR Workshop Proceedings (CEUR-WS.org)*.

Pierpaolo Basile, Annalina Caputo, Seamus Lawless, and Giovanni Semeraro. 2019a. Diachronic analysis of entities by exploiting Wikipedia page revisions. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 84–91, Varna, Bulgaria. INCOMA Ltd.

Pierpaolo Basile, Giovanni Semeraro, and Annalina Caputo. 2019b. Kronos-it: a Dataset for the Italian Semantic Change Detection Task. In *CLiC-it*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dagmar Divjak and Stefan Th Gries. 2006. Ways of trying in Russian: clustering behavioral profiles. *Corpus Linguistics and Linguistic Theory*, 2:23J60.

Hanne M Eckhoff and Laura A Janda. 2014. Grammatical profiles and aspect in old church slavonic. *Transactions of the Philological Society*, 112(2):231–258.

Mario Giulianelli, Marco Del Tredici, and Raquel Fernández. 2020. Analysing lexical semantic change with contextualised word representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3960–3973, Online. Association for Computational Linguistics.

Stefan Th Gries and Dagmar Divjak. 2009. Behavioral profiles: a corpus-based approach to cognitive semantic analysis. *New directions in cognitive linguistics*, 57:75.

Stefan Th Gries and Naoki Otani. 2010. Behavioral profiles: A corpus-based perspective on synonymy and antonymy. *ICAME Journal*, 34:121–150.

Kristina Gulordava and Marco Baroni. 2011. A distributional similarity approach to the detection of semantic change in the Google Books ngram corpus. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 67–71, Edinburgh, UK. Association for Computational Linguistics.

Patrick Hanks. 1996. Contextual dependency and lexical sets. *International journal of corpus linguistics*, 1(1):75–98.

Martin Hilpert and Stefan Th Gries. 2008. Assessing frequency changes in multistage diachronic corpora: Applications for historical corpus linguistics and the study of language acquisition. *Literary and Linguistic Computing*, 24(4):385–401.

Hans Henrich Hock and Brian D Joseph. 2019. *Language history, language change, and language relationship: An introduction to historical and comparative linguistics*. Walter de Gruyter GmbH & Co KG.

Michael Hoey. 2005. *Lexical Priming: A New Theory of Words and Language*. Routledge.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2016. Embeddings for word sense disambiguation: An evaluation study. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 897–907, Berlin, Germany. Association for Computational Linguistics.

Laura A Janda and Olga Lyashevskaya. 2011. Grammatical profiles and the interaction of the lexicon with aspect, tense, and mood in Russian. *Cognitive linguistics*, 22(4):719–763.

Patrick Juola. 2003. The time course of language change. *Computers and the Humanities*, 37(1):77–96.

Andrey Kutuzov, Lilja Øvrelid, Terrence Szymanski, and Erik Velldal. 2018. Diachronic word embeddings and semantic shifts: a survey. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1384–1397, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Andrey Kutuzov and Lidia Pivovarova. 2021a. RuShiftEval: a shared task on semantic shift detection for Russian. *Computational linguistics and intellectual technologies: Papers from the annual conference Dialogue*.

Andrey Kutuzov and Lidia Pivovarova. 2021b. Three-part diachronic semantic change dataset for Russian. In *Proceedings of the 2nd International Workshop on Computational Approaches to Historical Language Change 2021*, pages 7–13, Online. Association for Computational Linguistics.

Julia Kuznetsova. 2015. *Linguistic profiles: Going from form to meaning via statistics*. Walter de Gruyter GmbH & Co KG.

Ronald W Langacker. 1987. *Foundations of cognitive grammar: Theoretical prerequisites*, volume 1. Stanford university press.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems 26*, pages 3111–3119.

Syrielle Montariol, Matej Martinc, and Lidia Pivovarova. 2021. Scalable and interpretable semantic change detection. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4642–4652, Online. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the Word-in-Context Dataset for Evaluating Context-Sensitive Meaning Representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1267–1273, Minneapolis, Minnesota. Association for Computational Linguistics.

Anastasiia Ryzhova, Daria Ryzhova, and Ilya Sochenkov. 2021. Detection of semantic changes in Russian nouns with distributional models and grammatical features. *Computational linguistics and intellectual technologies: Papers from the annual conference Dialogue*.

Dominik Schlechtweg, Barbara McGillivray, Simon Hengchen, Haim Dubossarsky, and Nina Tahmasebi. 2020. SemEval-2020 task 1: Unsupervised lexical semantic change detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1–23, Barcelona (online). International Committee for Computational Linguistics.

Dominik Schlechtweg, Sabine Schulte im Walde, and Stefanie Eckmann. 2018. 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)*, pages 169–174, New Orleans, Louisiana. Association for Computational Linguistics.

Milan Straka and Jana Straková. 2017. Tokenizing, POS tagging, lemmatizing and parsing UD 2.0 with UDPipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada. Association for Computational Linguistics.

Charles Truong, Laurent Oudre, and Nicolas Vayatis. 2020. Selective review of offline change point detection methods. *Signal Processing*, 167:107299.

David Yenicelik, Florian Schmidt, and Yannic Kilcher. 2020. How does BERT capture semantics? a closer look at polysemous words. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 156–162, Online. Association for Computational Linguistics.

G. Manuscript: Benchmarks for Unsupervised Discourse Change Detection

Benchmarks for Unsupervised Discourse Change Detection

Quan Duong, Lidia Pivovarova, Elaine Zosa
University of Helsinki
firstname.lastname@helsinki.fi

ABSTRACT

The paper tackles a novel task of automatic extraction of discourse trends from large text corpora. The main motivation for this work lies in the need in digital humanities to track discourse dynamics in diachronic corpora. In many real use cases ground truth is not available and annotating discourses on a corpus-level is incredibly difficult and time-consuming. 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.

ACM Reference Format:
Quan Duong, Lidia Pivovarova, Elaine Zosa. 2021. Benchmarks for Unsupervised Discourse Change Detection. In ,. ACM, New York, NY, USA, 10 pages. https://doi.org/xxxxxxxxx

1 INTRODUCTION

Large collections of text, such as news archives, reflect valuable information on *discourse dynamics*—the change in prevalence of certain topics, opinions, and attitudes over a period of time. This is a valuable source of information in digital humanities and computational social sciences. Various NLP methods, from keyword extraction to topic modelling, have been established to facilitate discourse analysis. However, studying discourse dynamics is a novel and challenging research area that still needs to be developed.

This paper tackles a problem of automatic detection of discourse change in news streams. Our focus is the development of reliable methodology rather than investigating a particular use case. Thus, evaluation is our primary concern. In digital humanities, research questions are generally complex and involve a lot of uncertainty, thus the ground truth needed for quantitative evaluation is usually unavailable. Moreover, quite often digital humanities research deals with a specific use case, focusing on a single non-annotated dataset without a proper split into training and test subsets.

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. Synthetic datasets are widely used for a related task of lexical semantic change detection, but we are unaware of any similar work performed *on the discourse level* or exploiting *news categories* for a similar purpose.

The main contributions of this paper are the following:

- We draw attention to a discourse dynamic detection task that is relevant for humanities and computational social sciences but has been less studied within the NLP field.
- We establish a novel evaluation framework for discourse change detection and perform a large-scale experiment on a set of thousand synthetic datasets created to emulate six different patterns of discourse change.
- We propose several benchmark methods to tackle the problem. The best-performing method yields 78% accuracy.
- Finally, we perform a qualitative evaluation on a separate (unannotated) news corpus and demonstrate that a proposed method is able to find discourse change in a large news stream.

The rest of the paper is organized as follows. We start with presenting the background and related work for our paper in Section 2. Then in Section 3 we present a formal definition for the task at hand. Section 4 describes construction of synthetic datasets. Section 5 presents methods we tried to solve the problem. Section 6 describes evaluation metrics, while section 7 shows results obtained on synthetic datasets. Finally, Secion 8 describes our experiments on realistic data.

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 [11, 13, 22, 24], though others use techniques, such as language models and clustering [6, 7]. 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 have been several attempts within the NLP field to model discourse change, by the means of unsupervised topic models, such as dynamic topic models [1] or Topics over Time [23]. More recent models make use of word embeddings and neural inference networks to learn topics from data streams [3, 10]. However, even papers proposing these models often rely on use cases rather than numerical evaluations. 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 [12]; quite often a basic LDA model is preferred to more sophisticated models [5].

Another task relevant to diachronic change is lexical semantic change detection, which recently got a boost from by leveraging

Submitted for review,

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/xxxxxxxxx

word embeddings [8, 19]. In this task, manual data annotation is extremely challenging [15], and the datasets are rather small and scarce. Thus, synthetic datasets commonly used [14, 16, 18, 21]. Currently the usual approach is to merge two words with different meanings into one pseudo-word and then sample from their contexts according to some predefined distribution. A model is then evaluated by its ability to recognise the distribution.

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.

3 PROBLEM STATEMENT

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. The discourse change could only be found in *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 **subtasks**:

(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.

Finding training and evaluation data for this task is currently not possible because, as far as we know, there does not yet exist diachronic corpora annotated with discourses. Moreover, it is difficult to produce an annotated corpus for several reasons. First, annotation on a discourse level requires a lot of effort and can only be done by someone with a thorough knowledge of the corpus. Second, it is difficult for a human to distinguish between an actual change in the data from noise, and especially difficult to find a concrete pivot point, apart from some obvious cases. Third, since the task is defined on a corpus level, supervised learning would require annotation of multiple corpora, which is not feasible. Thus, all our models are trained and tested on synthetic data, while their applicability to real-world use cases is demonstrated qualitatively.

4 SYNTHETIC DATASETS

4.1 Yle News Corpus

The synthetic datasets are created from a corpus of news articles published from 2011 to 2018 by the national Finnish broadcasting company Yle. The corpus is distributed through Finnish Language Ban (Kielipankki)[1] and is freely available for research use[2].

The Yle corpus contains more than 700,000 articles written in Finnish published from 2011 to 2018. 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, which is important for the quality of the data. 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. This is then used for generating the synthetic datasets described in the following section.

4.2 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. In this work, we approximate a discourse as a category.

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, where there is no significant change in discourse prevalence over time. The precise formulation for the patterns are presented in Section 4.3

In our experiments, we use 100 time points, but this number can be flexible for different usages. Out of the 12 categories 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.

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

[2] 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.

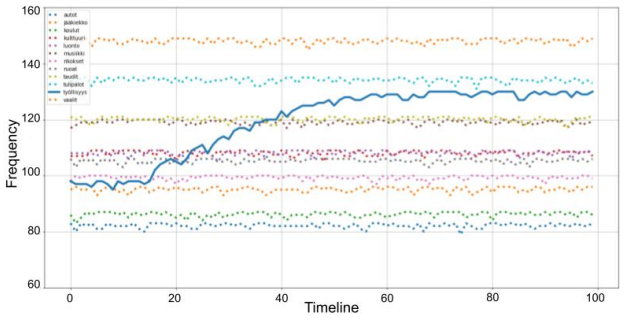


Figure 1: A sample experiment with 1 increasing category (Up-Down) and 11 stable categories.

While generating these sequences, we also randomly assign the pivot points when the non-stable pattern starts and ends, which is necessary for sub-task 3. Then we sample data from the remaining 11 categories, which all follow the stable pattern. Thus, sub-task 2 could be reformulated as finding documents that belong to a non-stable category among all documents in a given dataset.

An example dataset is presented in Figure 1. In this example the Up-Down pattern is used. We can see from the figure that random noise is added to all categories, so small spikes are visible for all categories, including stable ones. Note that the input to our trend detection model, described in Section 5.3 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.

4.3 Pattern Definitions

We now present formal definitions for the patterns. Two functions are used as fundamental components for discourse change: *sigmoid* or *Gaussian*. Each function with its adjustable parameters can create a typical shape, which we discuss in more details.

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 an S-curve, which is a natural shape in many language-change processes [2].

More concretely, a number of articles for each time point in Up and Down patterns follows this formula:

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

where T_{start} and T_{end} are the time points where the pattern starts and ends, respectively; X_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, selected randomly, 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 from the start to the end.

In the same way, the Gaussian function is suitable 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:

$$X \sim \mathcal{N}(\mu, \sigma^2)$$

$$\mu = (T_{end} - T_{start})/2, \sigma = (T_{end} - T_{start})/k$$

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

The variable X is drawn from the Gaussian distribution, where μ is the middle point in time range, and σ is estimated with 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 X sampled in the bell shape, we can calculate the number of articles for each time point, however, X needs to be rescaled using min-max scaling as in the last equation.

Another pattern that uses the Gaussian distribution is *periodic*– 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.

The stable pattern is a constant plus randomly sampled noise. The same noise is added to all other patterns to simulate a natural distribution of documents in the collection. We are aiming at 100-200 documents from each category at each time point, though obviously this number depends on the pattern.

4.4 Data Sampling

For each synthetic dataset only one category follows a non-stable pattern, while 11 additional categories follow a stable pattern. The role of stable patterns is to add noise for the next training steps, which helps create more generalised models. The way we construct data points for each pattern is as follows:

For the up, down, up-down and down-up patterns, we randomly assign two points in the timeline as the *pivot points*, denoted by T_{start} and T_{end} in the formulas above. Before and after the pivot points, the data is sampled according to the stable pattern. The data between the two pivots are generated by either the sigmoid or Gaussian functions. Note that each part will receive the number of articles $X_{start-1}$ of the last time point before the pivot point, so it can continue to generate $X_{start}....X_{end}$ for the new time range. Then a stable pattern is formed using X_{end} as a starting value.

Similarly, the periodic patterns share the same idea of generation, with the difference that there are more than two pivot points in the timeline. For the periodic sign, we randomly generate p intervals evenly in the timeline. The time points separating the intervals are considered as pivots. Using the Gaussian function, we sample the data points around these pivots.

As will be discussed in the next section, the approach we use to tackle the problem consists of two steps: unsupervised clustering of the documents followed by a supervised classification of the time series for each cluster. Since only the second step requires training data and each time series is processed independently of the others, for *training* we use the synthetic time series generated according to the formulas described above. In other words, for training we generate a number of documents from a given category at each time step but do not bother to sample real documents for this category

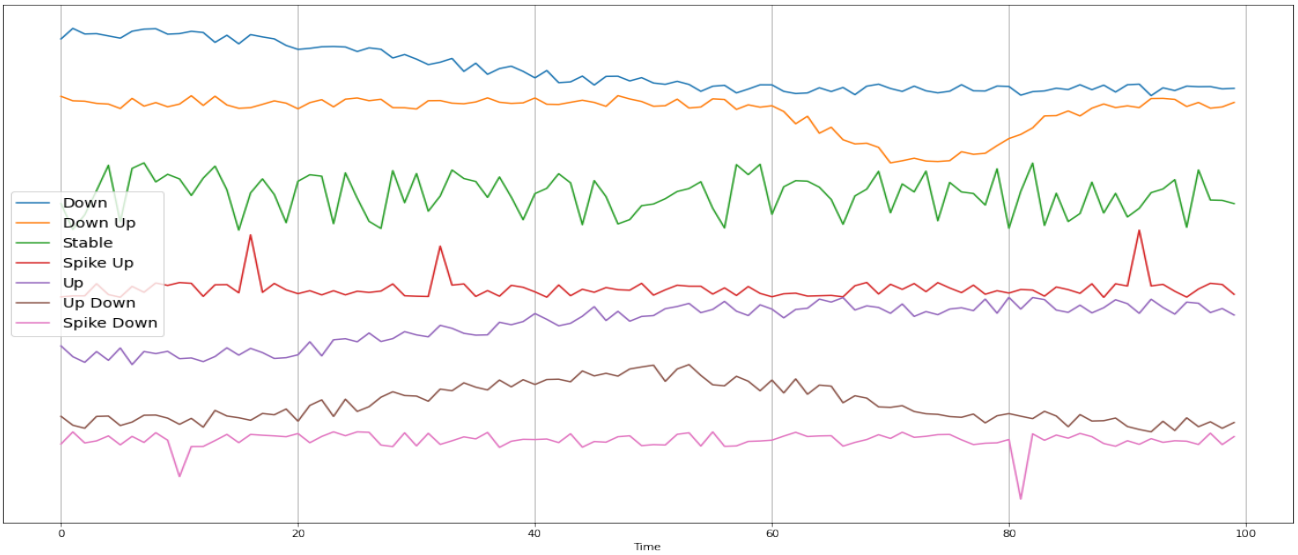


Figure 2: Seven patterns used to emulate discourse dynamics in the synthetic datasets.

because they are not used as an input for the model. Because the training data does not require to be clustered in step 1, the artificial noise is added to this data to mimic the noise from the clustering process. The purpose of this strategy is to provide a large number of samples for the training task and ensure the quality of the labeled pivot points which is hard to control if go through clustering step.

However, we would like to test the model performance in a realistic setting. Thus, *for testing* we generate an actual corpus by sampling from several categories according to the same patterns, then run this dataset through the clustering step and then apply the trained model. Thus, data used for testing is noisy. Again, we deem that to be a desirable property since our goal is not to build a dataset that would be easy to classify. On the contrary, we want the task to be sufficiently complex, allowing us to discriminate between various methods.

5 METHOD

5.1 Method overview

In all our experiments we use two major steps:

- building a timeseries from textual data;
- 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. Since regression requires an additional step for sequence segmentation, we utilize the sliding window approach for this purpose.

5.2 Building Timeseries

5.2.1 Clustering. **K-means** Clustering requires a dense document representation to compute the distances between articles. We use doc2vec model [9] to compute document representations[3]. The inferred document vectors are then clustered using k-means[4].

K-means clustering is run independently for each of the 1000 datasets. Thus, 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 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. The number of stable trends found by the model does not matter; it does not affect the measures of system performance, which we will describe in Section 6.

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 clustering. We use the Gensim implementation of LDA [5] with asymmetric priors learned from the data. We trained one topic model for each synthetic dataset. Topic model training was done in parallel. We set the number of topics to 20 to align with k-means.

The timeline on top of LDA is built using soft clustering instead of hard clustering, since an article can have more than one topic. The

[3]We use the Gensim implemetation of doc2vec (https://radimrehurek.com/gensim/models/doc2vec.html), with a dimensionality to 128, negative sampling of 5 words and train for 30 epochs.
[4]We used scikit-learn implementation with default parameters.
[5]https://radimrehurek.com/gensim/models/ldamodel.html

LDA output is usually unbalanced with a few large topics containing most of the articles, and many much smaller ones. To count 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. Thus each document contributes to at least one topic timeline. Similar to k-means, topic timelines are scaled to [0, 1] range.

5.2.2 Training Data. The cluster-based timeseries, described in the previous section, are used only to construct the validation set. As has been mentioned before, we are not using real articles as training data for a neural network. Instead, we directly sample the patterns with noise to mimic the sequence of frequency in the clustered set.

There is no guarantee that the article distribution obtained by clustering still maintains the shape of the pattern used to produce the dataset. It depends on the quality of the clustering method. When noise is introduced directly into the generated timeline, the shape of the pattern is not modified. Thus, the data used to train a neural network is cleaner than those used for validation. This way, we are able to generate more samples for training and avoid an unexpected behavior from the unsupervised process.

We generate a training set of 100K artificial timelines with noise. The noise is a product $n \times r$, where n is uniformly sampled from [0,1] range and r is a noise ratio sampled from 0.0001, 0.001, 0.002 with probabilities 0.1, 0.3, 0.6 respectively. To add even more variation, the pattern change rate is also sampled uniformly in range from 0.5 to 0.8.

The input for our models is a sequence of frequencies in a timeseries. There are 100 timepoints, each has the frequency normalized to [0,1] range.

The model produces two outputs: a binary prediction of whether a timeseries is stable or non-stable and a sequence of the same range, where the value at each time point is the probability that the time point belongs to a non-stable pattern. To generate this sequence 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.

Stable and non-stable timeseries are sampled equally for the training. Out of 100k samples in the training set, 5k are used as a development set and the rest as a training set.

5.3 Sequence to sequence model

We propose a novel neural network-based architecture that is trained jointly to solve two tasks: (1) to detect whether a timeseries has a non-stable pattern; and (2) to detect the pivot points in the non-stable timeseries.

The model is a combination of Recurent Neural Network (RNN) and Convolutional Neural Network (CNN). In addition to the combined model, we experiment wth CNN and RNN separately. In this section, we first present each model separately and then describe how they are combined.

5.3.1 Recurrent Neural Network. The RNN and its variations are designed for the temporal problem by memorizing the important information for each timestep [17]. The input to this model is a matrix with the shape $(N, 100)$ where N is the batch size and 100 is the length of the timeseries. Each example is a sequence of numbers in the range $[0, 1]$.

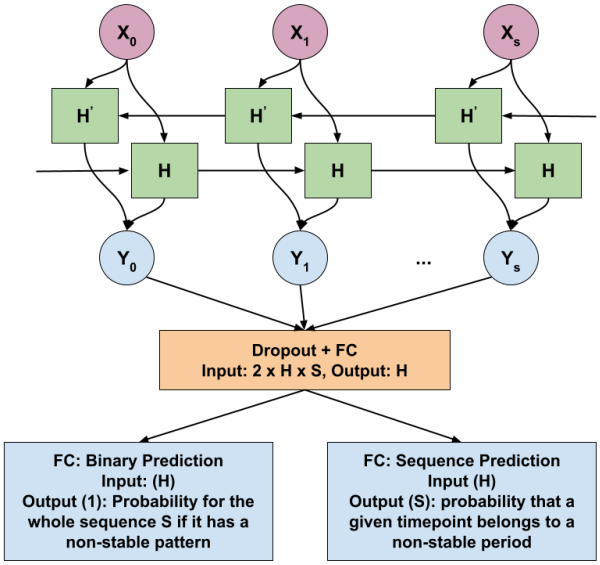


Figure 3: RNN network architecture with biLSTM layer. The prediction Y from all timesteps are used for the FC layer.

The model structure is presented in Figure 3. We use an RNN variant—bidirectional Long Short Term Memory (bi-LSTM)— stacked with one fully connected (FC) layer. Bi-LSTM is capable of learning long-term dependencies as well as capturing the features from both directions of the sequence. The bi-LSTM layer has 256 hidden units.

The following FC layer takes the all outputs from the LSTM layer and flatten them as input. A dropout layer is introduced to reduce the 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 pivot probabilities. Both output layers use the sigmoid activation function to get probability values.

5.3.2 Convolutional Neural Network. The CNN is intended for capturing locals features for image recognition [17]. 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.

5.3.3 Combined Model. While RNN is good at handling sequence information, CNN has the strength at local pattern detection. However, in a local region, if a non-stable shape is spawned accidentally due to the noise, the CNN model might mistake it as a

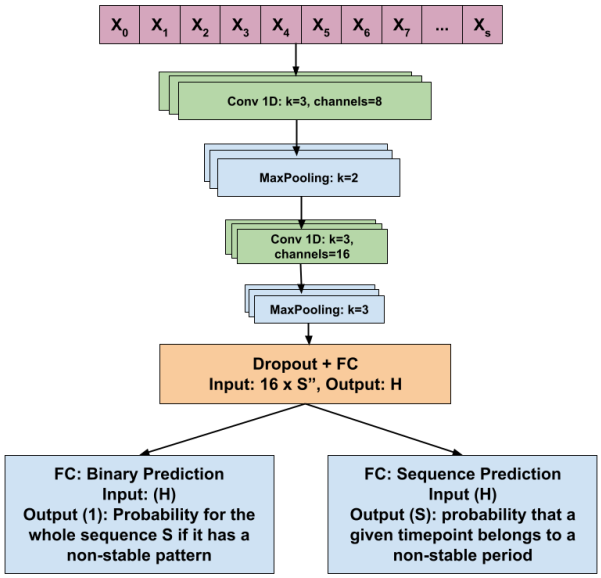


Figure 4: CNN network architecture. Where k is the kernel size, H is the hidden size, and S'' is the length of sequence after going through the convolutional layers.

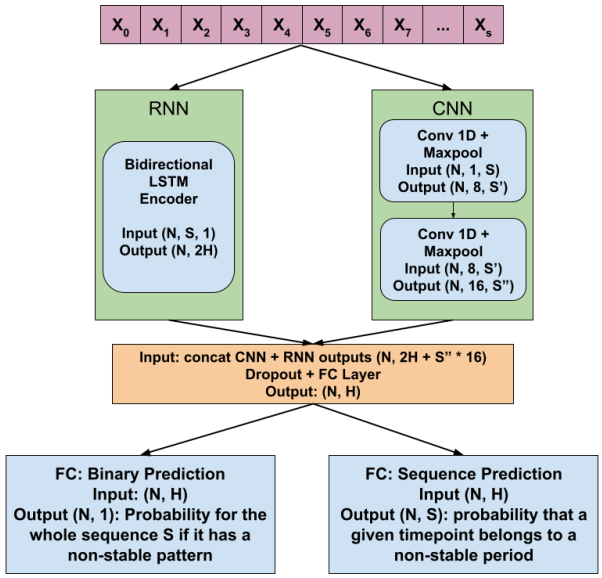


Figure 5: Combined (RCNN) network architecture. Where N is the batch size, H is the hidden size. S is the length of input sequence, S'' is the length of convolution output. The hidden states from LSTM are used for the next layer instead of the predicted outputs.

valid pattern. RNN can handle longer sequences due to it ability to "memorize" the sequence state. 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.

Note that when RNN is used alone we use all the prediction outputs of timesteps from LSTM layers. However, the combined model only takes the hidden state for the next step and all the timestep predictions are discarded.

After concatenating RNN and CNN outputs, the rest of the model is organized identical to the previous cases.

In all three neural network models, we set the dropout probability to 0.5. We train the models for 30 epochs with using a batch size of 64, Adam optimizer with learning rate $1 \cdot e^{-4}$. The loss is calculated by summarising two binary cross entropy (BCE) loss functions, one for binary classification and other for sequence of pivots prediction.

5.4 Baseline

5.4.1 Linear regression. Our baseline approach is based on linear regression. Unlike neural model, it is not independent for each cluster within dataset.

We fit a linear regression model to each of the 20 clusters obtained for the dataset. The slope 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. Depending on the threshold and slope value, there could be more than one non-stable clusters. 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.

5.4.2 Sliding Window Segmentation. Timeseries identified as non-stable in the previous step are processed using the sliding-window segmentation method to identify pivot points. We use the implementation in Rupture library [20].

The algorithm uses two windows, which slide along the timeline. These windows are used to measure the discrepancy between the left and right context at a given timepoint. The idea is that if both sliding windows fall in to the same segment, the discrepancy will be low. If the discrepancy is significantly higher, it can be considered as a pivot point.

The formula for calculating the discrepancy is taken from the Rupture documentation[6]:

$$d(y_{u..v}, y_{v..w}) = c(y_{u..w}) - c(y_{u..v}) - c(y_{v..w})$$

where c is the cost function, y_u, y_v, y_w are the input value at timepoints of sliding windows $u..v$ and $v..w$. Follow the documentation of Rupture, we set the window size 5 units, $jump = 5$, use L1 as cost function and a penalty of 0.5 to prevent overfitting.

[6]https://centre-borelli.github.io/ruptures-docs/user-guide/detection/window/

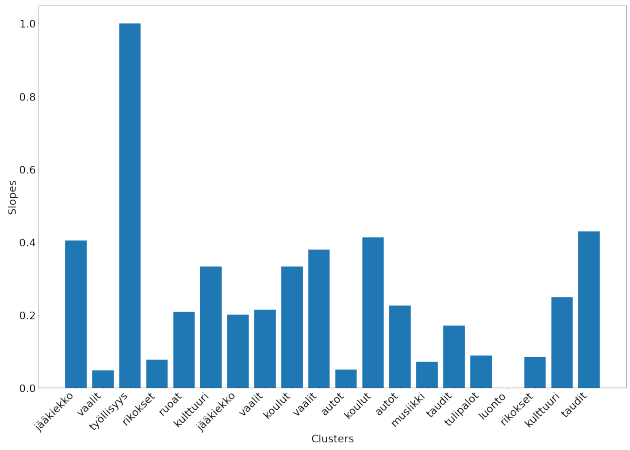


Figure 6: An example dataset, where each cluster, obtained from k-means, is fitted with linear regressions. The normalized slopes are shown in the histogram, with one pattern having significantly higher slope than the others, which indicates non-stable discourse dynamic. Bars are labelled with the major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

6 EVALUATION

We evaluate system performance at different levels:

- on the *dataset level* we calculate a percentage of datasets, where a model found at least one non-stable pattern;
- on the *category level* we use accuracy to measure how often if the major category within a detected non-stable cluster is the true non-stable category;
- on the *document level* we measure the proportion of true category documents within clusters detected as non-stable. For this we use recall, precision, and F-measure;
- finally on the *time-point level* we apply Rand-index to identify how close the predicted pivot points are to the real pivot points.

Since evaluation is done on synthetic datasets the ground truth for all these measures is directly available by the generation process. We now discuss each of these measures on more details.

6.1 Dataset-level Evaluation

Even though all synthetic datasets contain exactly one non-stable category, our models do not make any assumption on the number of non-stable patterns. Thus we hope the methods will generalize to real-world use cases where the number of changing discourses is not known in advance. As a consequence, an output can contain and arbitrary number of non-stable patterns, or none. As a first rough estimation of the model performance we compute a percentage of datasets, where a model predicted at least one non-stable dataset.

6.2 Category-level Evaluation

Category-level accuracy measures how well a model can detect a non-stable category. For each of 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 considered to be correct. For each dataset we calculate a ration 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 set 0. Thus, the model is punished for non-finding any changing trends but also punished for finding to many of them. However, it is not affected if a non-stable category is split into two clusters or if a non-stable cluster contains many documents from other categories.

6.3 Document-level Evaluation

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 clusters:

$$precision = \frac{C \cap N}{N}$$

, where C is a target category and N is a non-stable cluster.

The dataset precision is the mean of all non-stable cluster precisions. A model is penalized for splitting a target category in two clusters even if each of them does not contain any noise. On the other hand, precision for a cluster could be non-zero even if its major category is incorrect.

Recall is the proportion of documents from a non-stable cluster in a target category:

$$precision = \frac{C \cap N}{C}$$

Similar to precision, recall for all non-stable clusters is averaged, and a split of the target category leads to decrease of this measure.

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 all three measures are averaged across datasets.

Note that evaluation is focused on non-stable clusters and a target category. The presence of all other categories among clusters does not affect any of the measures. The reason for that is that our task is to extract dynamic trends from the data, rather than describe a collection as a whole.

6.4 Time-point level Evaluation

For each cluster that is classified as non-stable, a model must output also pivot points, i.e. time points where 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.

Rand index is computed as a proportion of time-point pairs correctly put either in the same or in the different periods. Since each timeline consists of 100 time points, shifting a pivot point by 1-2 positions from the true point slightly decreases Rand index. Radical misplacement or finding an incorrect number of pivot points, however, results in a large performance drop.

Rand index is averaged for all non-stable clusters in the dataset. If all clusters are classified as stable, Rand index is zero. This measure is then averaged across all datasets.

Method		Dataset	Category	Precision	Recall	F1	Rand
STEP 1	STEP2	coverage	accuracy				index
k-means	Regression	90.55	52.78	43.98	34.73	37.04	42.52
	RNN	95.33	73.63	60.55	46.33	50.43	73.17
	CNN	**96.59**	75.17	61.46	46.56	51.49	67.79
	RCNN	95.10	**78.43**	**63.77**	**51.69**	**55.22**	**73.26**
LDA	Regression	88.81	41.88	31.56	31.26	27.14	41.04
	RNN	89.51	38.65	30.48	31.84	27.53	65.04
	CNN	92.07	47.73	36.41	33.26	31.87	53.27
	RCNN	90.05	51.46	37.22	43.94	36.03	60.43

Table 1: Result obtained on 1000 synthetic 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.

7 RESULTS

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 only exception is the dataset coverage metrics, which is highest for the CNN model, though the difference is not significant. 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.

Comparing k-means and LDA, k-means works better for most of the models and measurements. In both cases, we can see CNN is better than RNN at non-stable pattern detection. However, RNN yields much higher Rand index, which means better at pivot points detection.

The difference between LDA and k-means would need deeper investigation in the future. The models applied on top of LDA yield low document-level F-measure, and especially low precision. The Rand index is also lower than for the k-means results though higher than could be expected judging from the document-level performance. For example, the LDA+RNN model yields a Rand index close to the k-means+CNN one, even though for LDA+RNN F-measure is much lower. This confirms our assumption that Rand index is independent of the document-level performance.

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 come to rich morphology language like Finnish, where words have many variants and compounds are frequently used [4]. This makes LDA hard to handle the semantic relationship, even with the lemmatized texts. Thus, the quality of clustering result from LDA might be affected.

8 EXPERIMENTS WITH REAL DATA

For a qualitative assessment, we use another Finnish corpus: The Finnish News Agency (STT) Archive, which is freely available for research use via Kielipankki[7]. The corpus consists of the STT newswire articles for the period between 1992-2018. We limit our experiments to the data from years 2007-2008, which does not overlap in time with YLE dataset. The data consist of approximately 250,000 documents.

For our experiments we split the two year data into weeks, excluding the first and the last two weeks, which gives us 100 weeks. Thus the timeline has the same length as the synthetic datasets and we could directly apply models trained on synthetic data.

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 briefly scanned the documents within these clusters and found a couple for which we could find interpretation.

Figure 7 shows a cluster, which contains articles about sport competitions. The periods of non-stability—between red vertical line in the plot—roughly correspond to two major sport events: the 2007 Hockey World Championships [8] and the 2008 Olympic games. We also found another sport-related cluster, where a period of instability roughly coincides with the Olympic games.

Another cluster, shown in Figure 8 is associated with party politics. The date of the Finish parliamentary elections is shown with green vertical line. This date is positioned two automatically determined pivot points—it seems natural that elections is actively discussed in the news some time before and after the event.

We use this experiment to demonstrate that a model trained on synthetic datasets, which are 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 computation social science research is yet to be found in collaboration with domain specialists. Most probably they would be interested in less obvious cases than sport events. It is not unlikely that cases difficult for us to interpret would be the most interesting for the experts.

Since training is done using synthetic data nothing prevents us from using more or less than a hundred datapoints. The number of clusters can also vary since each cluster is processed independently. This allows us to use different levels of granularity in discourse

[7] http://urn.fi/urn:nbn:fi:lb-2019041501
[8] https://en.wikipedia.org/wiki/2007_IIHF_World_Championship

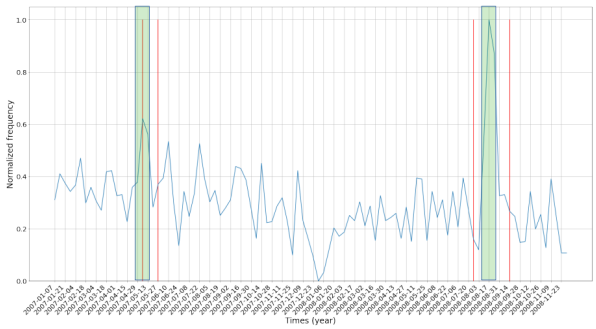


Figure 7: A non-stable cluster obtained on the STT data. Automatically detected pivot points show with red vertical lines. The documents within this cluster are about sport and competitions. The green rectangles show dates of the Hockey World Championship (left) and the Olympic games (right).

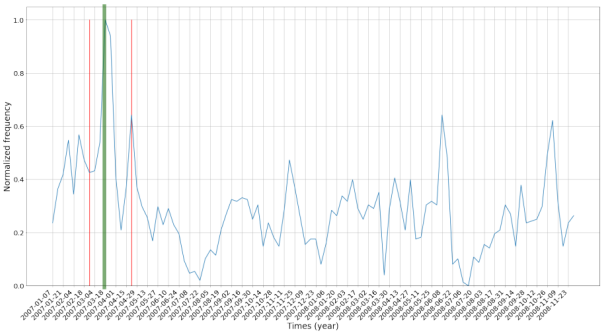


Figure 8: A non-stable cluster obtained on the STT data. Automatically detected pivot points show with red vertical lines. The documents within this cluster are mostly about politics and parties. The date of the Finnish Parliamentary elections is shown with green vertical line.

analysis, which would impact applicability and interpretability of results.

In our experiments with the STT data one cluster contains hundreds of documents, even if we limit our attention with articles that lie between pivot point. The clusters themselves are quite noisy, which could be expected from relatively low document-level F-measure for synthetic data. Thus, finding interpretations for non-stable behaviour is a tedious task. However, we could combine our method with other automatic description techniques, such as finding the most prominent keywords for each cluster or generating a summary.

CONCLUSION

In this paper we presented the novel task of automatic detection of discourse change in large text collections. This task is relevant for many research questions in digital humanities and computational social science and could potentially have applications in media monitoring. However, computational methods to tackle this type of problem is 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 lean to discourse change simulation with synthetic data.

Synthetic data allows us to train supervised models. Moreover, the procedure which we proposed in this paper generates sufficiently complex datasets so that the problem cannot be solved by simple means, such as regression. This allows for evaluation, comparison, and improvement of the methods, impossible on most typical use cases where ground truth in not accessible.

We proposed a combination of clustering with a neural sequence-to-sequence model to extract non-stable trends and find periods of instability in the data. The best-performing method yields 78% accuracy in non-stable trend detection and 73% Rand index for pivot time points detection. Nevertheless, the complexity of the task leaves much space for improvement even on synthetic data.

Finally, we demonstrated that a model trained on synthetic data is able to find change in real news content, without fine-tuning or any other adjustments for the data. This is a valuable property, since digital humanities use cases often involve a single unique dataset, which makes it difficult to optimise models or tune hyper-parameters.

Further work will continue in two directions in parallel. First, we will collaborate with humanities and social science specialists to test applicability of the proposed method in practice. Previous collaborations indicate that there is an actual need for a model able to track discourse change in textual data, though we have not yet had a chance to apply the models developed in this paper to a humanities use case.

Second, we will continue improving our models using synthetic datasets.This does not need any manual evaluation and to a large extent could be done independently from domain specialists. One obvious—though not trivial—next step could be using texts as an input, in addition to one-dimensional timeseries utilized in the current implementation. We will also investigate how to utilize full LDA output rather than hard topic assignments.

REFERENCES

[1] David M Blei and John D Lafferty. 2006. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*. 113–120.
[2] Richard A Blythe and William Croft. 2012. S-curves and the mechanisms of propagation in language change. *Language* (2012), 269–304.
[3] Adji B Dieng, Francisco JR Ruiz, and David M Blei. 2019. The dynamic embedded topic model. *arXiv preprint arXiv:1907.05545* (2019).
[4] Quan Duong, Mika Hämäläinen, and Simon Hengchen. 2020. An Unsupervised method for OCR Post-Correction and Spelling Normalisation for Finnish. arXiv:2011.03502 [cs.CL]
[5] David Hall, Dan Jurafsky, and Christopher D Manning. 2008. Studying the history of ideas using topic models. In *Proceedings of the 2008 conference on empirical methods in natural language processing*. 363–371.
[6] Simon Hengchen, Ruben Ros, and Jani Marjanen. 2019. 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*.
[7] Mike Kestemont, Folgert Karsdorp, and Marten During. 2014. 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*. 62.
[8] Andrey Kutuzov, Lilja Øvrelid, Terrence Szymanski, and Erik Velldal. 2018. Diachronic word embeddings and semantic shifts: a survey. In *Proceedings of the 27th International Conference on Computational Linguistics*. 1384–1397.

[9] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*. 1188–1196.

[10] Yue Li, Pratheeksha Nair, Zhi Wen, Imane Chafi, Anya Okhmatovskaia, Guido Powell, Yannan Shen, and David Buckeridge. 2020. Global Surveillance of COVID-19 by mining news media using a multi-source dynamic embedded topic model. In *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*. 1–14.

[11] Ryan Light and Jeanine Cunningham. 2016. Oracles of peace: Topic modeling, cultural opportunity, and the Nobel peace prize, 1902–2012. *Mobilization: An International Quarterly* 21, 1 (2016), 43–64.

[12] Jani Marjanen, Elaine Zosa, Simon Hengchen, Lidia Pivovarova, and Mikko Tolonen. 2021. Topic modelling discourse dynamics in historical newspapers. (2021).

[13] David J Newman and Sharon Block. 2006. Probabilistic topic decomposition of an eighteenth-century American newspaper. *Journal of the American Society for Information Science and Technology* 57, 6 (2006), 753–767.

[14] Alex Rosenfeld and Katrin Erk. 2018. 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)*. 474–484.

[15] Dominik Schlechtweg, Sabine Schulte im Walde, and Stefanie Eckmann. 2018. 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)*. 169–174.

[16] Dominik Schlechtweg and Sabine Schulte im Walde. 2020. Simulating Lexical Semantic Change from Sense-Annotated Data. In *The Evolution of Language: Proceedings of the 13th International Conference (EvoLang13)*.

[17] Jürgen Schmidhuber. 2015. Deep learning in neural networks: An overview. *Neural Networks* 61 (Jan 2015), 85–117. https://doi.org/10.1016/j.neunet.2014.09.003

[18] 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)*. 66–76.

[19] Xuri Tang. 2018. A state-of-the-art of semantic change computation. *Natural Language Engineering* 24, 5 (2018), 649–676.

[20] Charles Truong, Laurent Oudre, and Nicolas Vayatis. 2020. Selective review of offline change point detection methods. *Signal Processing* 167 (2020), 107299.

[21] Adam Tsakalidis and Maria Liakata. 2020. 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)*. 8485–8497.

[22] Lorella Viola and Jaap Verheul. 2020. Mining ethnicity: Discourse-driven topic modelling of immigrant discourses in the USA, 1898–1920. *Digital Scholarship in the Humanities* 35, 4 (2020), 921–943.

[23] Xuerui Wang and Andrew McCallum. 2006. 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*. 424–433.

[24] Tze-I Yang, Andrew Torget, and Rada Mihalcea. 2011. Topic modeling on historical newspapers. In *Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*. 96–104.

H. Manuscript: Visual Topic Modelling for NewsImage Task at MediaEval 2021

Visual Topic Modelling for NewsImage Task at MediaEval 2021

Lidia Pivovarova, Elaine Zosa
University of Helsinki, Finland
first.last@helsinki.fi

ABSTRACT

We present the Visual Topic Model (VTM)—a model able to generate a topic distribution for an image, without using any text during inference time. The model is applied to an image-text matching task at MediaEval 2021. Though results for this specific task are negative (the model works worse than a baseline), we demonstrate that VTM produces meaningful results and can be used in other applications.

1 INTRODUCTION

We present a novel approach for Visual Topic Modelling (VTM), i.e. assigning to an image a topic distribution, where 2-3 topics are the most probable ones. A topic is represented as a list of words, so an image is labeled with a set of predefined keywords.

VTM is an extension of Contextualized Topic Models (CTM) [1]. For training it requires pairs of images and texts. During inference, it takes as an input only an image. Thus, the model is capable of assigning topics to an image without any textual description.

In this paper, we apply VTM for MediaEval 2021 NewsImage Task 1, i.e. matching news articles with corresponding images [4]. In short, our approach consists of training two *aligned* topic models: one model takes as an input text, another takes as an input image, both produce as an output, a topic distribution. During training, we use aligned texts and images and train models in such a way that they have a similar output distributions. During inference time, to find images corresponding to a given text, we apply visual and text models independently and then sort images by topic distribution similarity to the text topic distribution. Since each topic can be represented as a set of keywords, results are intertpretable.

To train aligned visual and text topic models we use *knowledge distillation* approach [3], i.e. first training a *teacher* and then training a *student* model that should produce an output similar to those produced by the teacher.

Our experiments with text to image matching produced *negative results*: a solution based on VTM works worse than a baseline, based on cosine similarity between out-of-the-box text and image embeddings [5]. Nevertheless, we believe that topic modelling for images can have many other applications. It can also be possible to improve the current solution with hyperparameter tuning or by using a larger training set.

2 METHOD

2.1 Visual Topic Model

VTM is an extension of CTM [1]. CTM is a family of neural topic models that is trained to take as an input, text embeddings and to produce as an output the bag-of-words reconstruction. The model trains an inference network to estimate the parameters of the topic distribution of the input. During inference time this topic distribution is used as the model output to describe texts unseen during training.

Thus, to train a model, each input instance has two parts: text embeddings and bag-of-words representation (BoW). Our main contribution is that we replace text embeddings with visual embeddings and demonstrate that they can be used to train a topic model. The ZeroShot CTM model uses the BoW representation only to compute loss, i.e. this information is not needed during inference time. Since we have a training set that consists of aligned text and image pairs we can use the texts to produce the BoW representation and use it to train a model.

To obtain image embeddings we use CLIP—a pretrained model that produces text and image embeddings in the same space [5]. CLIP representations for text and image are already aligned. However, this is not a requirement for VTM: in our preliminary experiments we used ViT [2] for image and German BERT for texts (https://huggingface.co/bert-base-german-cased). The results obtained using non-aligned embeddings were only slightly worse than those with CLIP embeddings. Topic models converge to similar results because they use the same BoW to compute loss; alignment of embeddings simplifies this process but is not necessary.

This basic procedure, i.e. training image and text models independently, produces similar but not aligned topic models. Topics could be slightly different and even similar topics are organized in different (random) order. To increase similarity between text and image models we use knowledge distillation. In this approach a student model uses a different input than a teacher (e.g. image instead of text) but should produce the same result.

CTM uses a sum of two losses: reconstruction loss and divergence loss. The reconstruction loss ensures that the reconstructed BoW representation is not far from the true one. The divergence loss, measured as KL-divergence between *priors* and *posteriors*, ensures a diversity property, that is desired for any topic model: a topic has large probabilities only for a small subset of words and a document has high probabilities only for a small subset of topics.

In the knowledge distillation approach we leave the reconstruction loss intact but replace divergence loss with KL-divergence with regards to the *teacher output*. The assumption here is that since a teacher model is already trained to be diverse and a student model is trained to mimic the teacher, the student does not need priors. Experiments supported this assumption.

We use knowledge distillation in two versions: *joint model* and *text-teacher*. In the joint approach we first train a joint model that takes as an input a concatenation of text and image embeddings, then train two student models for image and text separately. In the second approach, we first train a text model and then train an

MediaEval'21, December 13-15 2021, Online

Table 1: Results

Model	Correct in Top100	MRR@100	Recall@5	Recall@10	Recall@50	Recall@100
baseline (CLIP)	1225	0.169	0.22	0.30	0.53	0.64
joint 120 topics	767	0.043	0.06	0.09	0.26	0.40
joint 60 topics	698	0.030	0.04	0.07	0.24	0.36
text teacher 120 topics	816	0.042	0.05	0.09	0.30	0.43
text teacher 60 topics	757	0.037	0.05	0.08	0.26	0.39

(a) CLIP 1st

(b) VTM 1st

(c) CLIP 2nd

(d) VTM 2nd

Figure 1: Images, most close to the story about the trial of Anna Semenova according to the baseline (a,c) and VTM (b,d) models.

image model as a student. We try 60 and 120 topics with both joint and text-teacher approaches.

2.2 Baseline

As a baseline, we use raw cosine similarities between CLIP embeddings, without any domain adaptation for the text. We use an implementation provided as a part of Sentence Bert package (https://www.sbert.net/examples/applications/image-search/README.html).

3 RESULTS

The results are presented in Table 1. As can be seen from the table, the best results are obtained with CLIP embeddings, that are used without any fine-tuning to the training set. They are able to find the correct image in 1225 cases out of 1915 and has a Mean Reciprocal Rank (MRR) of 0.17. The best VTM model finds correct image in 816 cases out of 1915 and yields an MRR of 0.03.

These results to some extent correspond to our previous observation that topic modelling is not the best method for document linking [6]. The probable explanation for that might be that topic modelling produces a sparse representation of the data. While CLIP embeddings are continuous vectors and could represent an almost infinite amount of information, in topic modelling dimensions are not independent due to the diversity requirement, described above. It can be seen from Table 1 that models that have more topics yield better performance.

Another interesting observation is that models that use the text model as a teacher for a visual model work better than joint models. This is an unexpected result, since one would expect that a model that has access to full information could serve as a better teacher. It is possible that text bears less noise: a text model uses the same text for contextual and BoW representation, while an image could be completely random.

The fact that embeddings and topic modelling work on different principles is illustrated in Figure 1, where we reproduce images found by the model for the text about the Anna Semenova trial. CLIP model finds photos of Anna Semenova, probably due to the huge text and image base used to train the embeddings. VTM returns images with a statue of Themis, a personification of Justice, which represent the text *topic* rather than specific facts. Though according to our results, CLIP embeddings outperform VTM, the ability to illustrate text topic might be a desirable property for some applications, as well as topic interpretability.

Our code is available at https://github.com/lmphcs/media_eval_vctm.

ACKNOWLEDGMENTS

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

REFERENCES

[1] Federico Bianchi, Silvia Terragni, Dirk Hovy, Debora Nozza, and Elisabetta Fersini. 2021. Cross-lingual Contextualized Topic Models with Zero-shot Learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Association for Computational Linguistics, Online, 1676–1683. https://www.aclweb.org/anthology/2021.eacl-main.143

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, and others. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.

[3] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision* 129, 6 (2021), 1789–1819.

[4] Benjamin Kille, Andreas Lommatzsch, Özlem Özgöbek, Mehdi Elahi, and Duc-Tien Dang-Nguyen. 2021. News Images in MediaEval 2021. CEUR Workshop Proceedings.

[5] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and others. *Learning Transferable Visual Models From Natural Language Supervision*. Technical Report.

[6] Elaine Zosa, Mark Granroth-Wilding, and Lidia Pivovarova. A Comparison of Unsupervised Methods for Ad hoc Cross-Lingual Document Retrieval. In *LREC 2020 Language Resources and Evaluation Conference 11–16 May 2020*. 32.