

APPROVED: 31 May 2023
doi: 10.2903/sp.efsa.2023.EN-8212

Topic modelling and text classification models for applications within EFSA

Brecht Vandervoort, Geert-Jan Bex, Jonas Crevecoeur, Frank Neven
Data Science Institute, Hasselt University, Belgium

Abstract

This report presents an overview of topic modelling and classification models in relation to four case studies in the EFSA project OC/EFSA/AMU/2020/02. As adequate document embeddings have a positive influence on the effectiveness of topic modelling as well as text classification, an extensive number of different possibilities for word and document embeddings are discussed. It was found that a multitude of increasingly more complex embeddings are readily available for off-the-shelf use. But as they are trained on large but mostly general text corpora, their utility for domain specific text varies. Fine tuning or creating document embeddings from scratch is only feasible in the presence of enough data and has an associated computational cost. For some domains (like scientific articles), pretrained embeddings are available. For topic modelling, we discuss standard techniques like non-negative matrix factorization and latent Dirichlet allocation as well as more recent methods based on clustering of document embeddings like Top2Vec and BERTopic. For text classification, we consider hierarchical text classification approaches combined with established techniques for text classification via document embeddings. We propose a selection of techniques for each of the case studies justifying their choice and present a plan for evaluation. Finally, we discuss our findings after having implemented and validated the selected techniques.

© European Food Safety Authority, 2023

Key words: Natural Language Processing, Topic Modelling, Text Classification

Question number: EFSA-Q-2023-00503

Correspondence: know@efsa.europa.eu



Disclaimer: The present document has been produced and adopted by the bodies identified above as author(s). This task has been carried out exclusively by the author(s) in the context of a contract between the European Food Safety Authority and the author(s), awarded following a tender procedure. The present document is published complying with the transparency principle to which the Authority is subject. It may not be considered as an output adopted by the Authority. The European Food Safety Authority reserves its rights, view and position as regards the issues addressed and the conclusions reached in the present document, without prejudice to the rights of the authors.

Acknowledgements: The authors would like to thank Yannick Spill, Carsten Behring, Federica Barrucci, Ermanno Cavalli, Laura Martino, Angelo Cafaro, Valeria Ercolano, Spela Supej, Silvia Valtueña Martinez, Ionut Craciun, Barbara Viviani and Agnès Rortais.

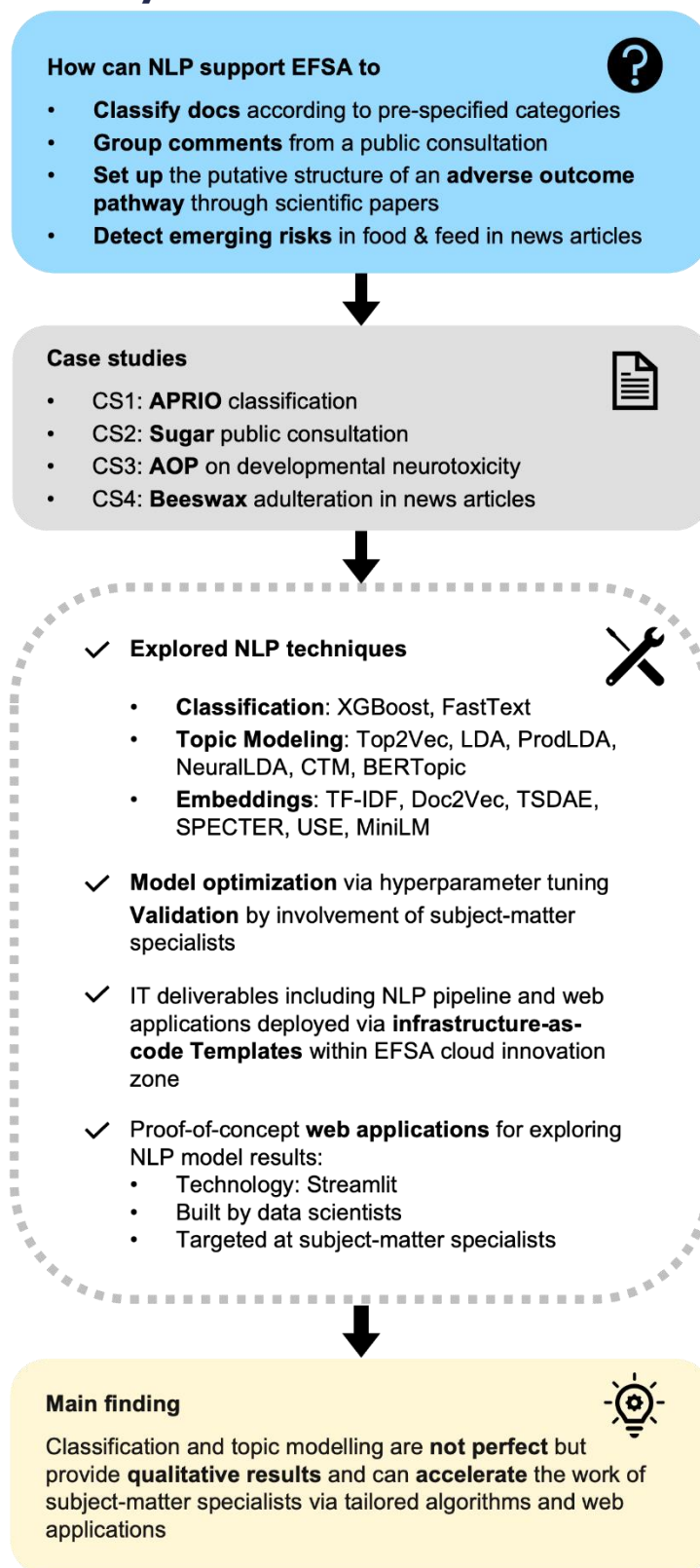
Suggested citation: Vandevoort B., Bex G. J., Crevecoeur J., Neven F., 2023. Topic modelling and text classification models for applications within EFSA. EFSA supporting publication 2023:EN-8212. 113 pp. doi:10.2903/sp.efsa.2023.EN-8212

ISSN: 2397-8325

© European Food Safety Authority, 2023

Reproduction is authorised provided the source is acknowledged.

Graphical summary





Non-technical summary

This project explores how Natural Language Processing (NLP) techniques can support subject-matter specialists in performing the following tasks:

1. Classify documents according to pre-specified categories such as for instance harmonised question classification.
2. Group similar comments from a public consultation
3. Explore scientific papers to set up the putative structure of an adverse outcome pathway
4. Explore news articles to detect emerging risks in food and feed

This project explored a variety of NLP classification and topic modelling techniques that were tested for the following case studies:

1. **CS1: APRIO classification**

A new problem formulation framework called Agent-Pathway-Recipient-Intervention-Outcome (APRIO) has been used to classify around 700 EFSA scientific opinions. The scope of this study is to assess the quality of current document classification algorithms with the aim of assisting humans at labelling the entire corpus of EFSA scientific opinions.

2. **CS2: Sugar public consultation**

About 700 comments were obtained by EFSA following public consultation on the sugar scientific opinion. Any two comments can be duplicates, related (address an overlapping set of issues than another comment), or unrelated. The scope of this study is to group the comments in order to ease the formulation of replies.

3. **CS3: AOP on developmental neurotoxicity**

The AOP (Adverse Outcome Pathway) approach requires setting a pathway to harm defined by a sequence of key events that originates from a molecular initiating event and lead to adverse outcome/s. The scope of this study is to investigate the ability of the state-of-the-art topic modelling techniques to support the exploration of the body of evidence to set up the putative structure of an AOP. The corpus of papers used for the AOP on developmental neurotoxicity will be used to train the model.

4. **CS4: Beeswax adulteration in news articles**

EFSA develops tools for the detection of emerging risks in food and feed. An example of a specific food fraud incident is beeswax adulteration: it can be adulterated for financial gain and, although it is a product from apiculture, it might enter the food chain when it is introduced as honeycomb in honey pots. However, beeswax can be used in other sectors, like cosmetic and food wrapping. Therefore, we seek to investigate the ability of the state-of-the-art topic modelling techniques to support the exploration of the body of web news, to recognize different sectors in which beeswax is used and identify the articles belonging to the food topic.

The main findings are as follows:



- CS1: The quality of the prediction of the NLP classification models is determined more by the size and the quality of the labelled training data than by the complexity of the NLP models. Classification labels that are well-represented in the labelled data set can already be predicted by simple models that have the advantage of being explainable, for instance, by providing the words most important for a prediction. A proof-of-concept web app that supports exploration of assigned labels has been developed as part of this project.
- CS2: The state-of-the-art NLP topic modeling techniques do not allow (near) perfect grouping for small datasets with highly correlated text as is the case with comments on a public opinion. This means in particular that expert involvement is still required for such a task. Manual validation of the best performing models by the domain experts did show that around half the groupings of comments were identified as coherent which indicates that the output of these models can still serve as an initial grouping, thereby facilitating the task of the domain expert in finding related comments. The best model was found to be ProLDA. A proof-of-concept web app that supports exploration of groupings has been developed as part of this project.
- CS3+CS4: Expert validation showed that the state-of-the-art NLP topic modelling technique (Top2Vec) results in qualitative topics with good topic descriptions. However, topic models are unable to do task-specific classifications. For example, they cannot differentiate between topics relevant to a specific use case and topics irrelevant to this use case. Automatic evaluation of groups therefore remains a hard task as the model might find coherent groups that are not of interest to the expert. Topic modelling should therefore be situated as a tool to aid the domain expert in finding relevant documents in a large corpus more efficiently, rather than a fully automated replacement. In particular, topic modelling is a valid tool to scope a large corpus of literature and uncover the unknown when the problem formulation is not clear since the beginning (as in the case of AOPs). A proof-of-concept web app that supports exploration of and interaction with topics has been developed as part of this project

The provided web apps are implemented in Streamlit, a lightweight framework that allows for rapid development of interactive dashboards, is tailored for data scientists and does not require any proficiency in web development. Both the webapps as well as the pipelines for training models are available within the EFSA cloud innovation zone via infrastructure as code templates facilitating rapid deployment.



Technical summary

This report presents an overview of topic modelling and classification models in relation to four case studies in the EFSA contract OC/EFSA/AMU/2020/02. We propose a selection of techniques for each of the case studies justifying their choice and presenting a plan for evaluation. The selected techniques are subsequently implemented and evaluated.

As adequate document embeddings have a positive influence on the effectiveness of topic modelling as well as text classification, an extensive number of different possibilities for word and document embeddings are discussed. More precisely, for word embeddings we discuss Word2Vec, GloVe, FastText, ELMo, and BERT. For document embeddings, we discuss TF-IDF vectors, averaging word embeddings, fine-tuning BERT for document similarity, Universal Sentence Encoders, SimCSE, Sentence-T5, TSDAE, GPL and SPECTER. It was found that a multitude of increasingly more complex embeddings are readily available for off-the-shelf use. But as they are trained on large but mostly general text corpora, their utility for domain specific text varies. Fine tuning or creating document embeddings from scratch is only feasible in the presence of enough data and has an associated computational cost. When there are more than 10,000 documents available TSDAE could be considered for self-supervised training of document embeddings. For some domains (like scientific articles), the pretrained sentence embedding SPECTER can be used.

For topic modelling, we discuss standard techniques like non-negative matrix factorization and latent Dirichlet allocation as well as more recent methods based on clustering of document embeddings like Top2Vec and BERTopic. It was found that LDA provides a good baseline and is helpful for situations that require mixed topics. Top2Vec is a very promising method and allows the possibility to use various document embedding methods. For text classification, we consider hierarchical text classification approaches combined with established techniques for text classification via document embeddings as XGBoost, support vector machines and FastText. Some case studies require segmentation of the input for which we consider a structural as well as a semantic approach called TextTiling.

To assess the effectiveness of topic modelling we consider the following metrics: topic information gain, topic coherence, topic diversity and cluster purity. Of these, topic coherence has been shown to reflect human judgment. Cluster purity is a metric that requires a ground truth to be available.

In summary, we provide an overview of our proposal for each of the case studies and report on our findings.

Case Study 1:

We propose to investigate the following techniques:

1. TF-IDF vectors for document representation combined with XGBoost for classification serving as a baseline.
2. Document embeddings based on averaging word embeddings or a pre-trained SBERT model. For hierarchical classification, a local classifier per parent node is constructed based on a combination of XGBoost and rule-based classifiers.
3. Document embeddings derived from FastText combined with rule-based classifiers.

Each model is evaluated by quantitatively comparing with the available ground truth. The most important findings are the following:

www.efsa.europa.eu/publications



- All tested NLP models performed equally well in predicting the APRIIO labels. The models are capable of retrieving 80% of the labels given to each document, and 20% of the labels attributed by the models were wrong. As a result, this task cannot be fully automated and domain experts still have to carefully review the predicted labels. Models based on TF-IDF have the additional advantage that most important features are interpretable. For this case study, we hence propose TF-IDF with XGBoost for hierarchical classification.
- Performance of the NLP model depends on the question, pillar and subquestion at hand. The models perform very well on labels which are frequently attributed and have a specific vocabulary. The models should not be used to predict infrequent labels.
- A proof-of-concept web app that supports exploration of assigned labels has been developed as part of this project.

Case Study 2:

We propose to investigate the following techniques:

1. The first strategy applies a multi-topic topic model where each comment can be assigned to an arbitrary number of topics. The expected advantage of this approach is that both semantically separated and interwoven comments should end up in the corresponding topics.
2. The second strategy leverages text segmentation to partition comments in segments covering a single issue. Afterwards, each segment is assigned to a single topic. The expected advantage of this approach is that topic assignment of semantically separated comments is improved, since each segment covers only one issue. The disadvantage of this approach is that interwoven comments cannot be segmented properly, and are therefore expected to end up in only one topic.
3. The last strategy combines both previous strategies into a hybrid strategy: comments are segmented and both the segments and original comments are clustered using a single-topic approach. Afterwards, we use the topic assigned to the original comment, unless the topic probability for this comment doesn't meet a predefined threshold. In that case, the topics assigned to the segments derived from this comment are used instead.

Since a ground truth is available and since the focus of the case study is on grouping related comments into topics rather than finding good topic representations, cluster purity will be used as the main performance metric. Other metrics will be reported, but are considered less relevant.

The most important findings are the following:

- The majority of comments on the given opinion are highly correlated. Close relatedness between most of the comments provides an additional challenge for topic models, as the boundaries between topics will be less pronounced.
- Due to the small number of comments and high correlation between comments, baseline models (ProLDA in particular) are on par with or even outperform the more complex Top2Vec models when considering evaluation metrics relevant for this case study.
- For new datasets without a ground truth, topic information gain is a good choice for model evaluation during optimization.



- Currently, the state-of-the-art topic modelling techniques do not allow (near) perfect clustering for small datasets with highly correlated text, meaning that expert involvement is still required for such a task. However, manual validation of the best performing models by the domain experts showed that around half the considered clusters were identified as being not too broad. This indicates that the output of these models can still serve as a good initial clustering, thereby facilitating the task of the domain expert in finding related comments. A proof-of-concept web app that supports exploration of groupings has been developed as part of this project.

Case Study 3:

We propose to investigate the following techniques:

1. Topic models based on variations of LDA (LDA, neural LDA, prodLDA, CTM) serve as a baseline.
2. Top2Vec, based on both pre-trained embeddings as well as an embedding fine-tuned over the given data by using TSDAE. Since the number of documents exceeds the recommended threshold of 10,000 documents, we expect this model to potentially outperform pre-trained embeddings. An important consideration is that the computational requirements for training such a model based on TSDAE cannot be derived from the literature. Without actual experiments, it is therefore still unclear whether training is practically achievable.
3. Since the corpus consists of titles and abstracts of scientific papers, SPECTER is expected to give better results than a general-purpose pre-trained embedding. We intend to use SPECTER within Top2Vec. However, the documentation of Top2Vec is vague on the possibility of including a custom embedding that is not an SBERT model or Universal Sentence Encoder. If including SPECTER into Top2Vec is not technically possible, we plan to apply UMAP and HDBSCAN directly on top of the computed SPECTER embeddings instead.

For each strategy, we report and compare topic information gain, coherence and diversity. The most important findings are the following:

- Top2Vec generally outperforms the other models, with the chosen embedding significantly influencing the results. Document embeddings based on Doc2Vec significantly outperform embeddings based on BERT or Universal Sentence Encoders. When choosing a BERT-based model for an unlabelled dataset, pre-trained embeddings are recommended over custom embedding using unsupervised learning, as the latter require additional training without providing improved results.
- For datasets where numerous topics are expected with each topic covering a small fraction of the corpus, the removal of infrequent words is not recommended. Such a removal is expected to rule out words relevant for these smaller topics, thereby reducing topic quality.
- The state-of-the-art topic modelling algorithm results in qualitative topics with good topic descriptions. However, it is important to note that topic models are not directly suited to classify documents according to a task-specific classification. For this case study in particular, they cannot differentiate between topics relevant to a specific use case and topics irrelevant to this use case. Topic modelling should therefore be situated as a tool to aid the domain expert in exploring and classifying documents in a large corpus more efficiently, rather than a fully automated replacement. In particular, topic modelling is a valid tool to scope a large corpus of literature and uncover the unknown



when the problem formulation is not clear since the beginning (as in the case of AOPs). A proof-of-concept web app that supports exploration of and interaction with topics has been developed as part of this project.

Case Study 4:

We propose to investigate the following techniques:

1. Topic models based on a variation of LDA (LDA, neural LDA, prodLDA) serves as a baseline.
2. Top2Vec on a wide range of document embeddings (Doc2Vec, pre-trained SBERT models, pre-trained Universal Sentence Encoder models)
3. BERTopic on a wide range of document embeddings (Doc2Vec, pre-trained SBERT models, pre-trained Universal Sentence Encoder models)

For each strategy, we report and compare topic coherence and diversity. We furthermore propose an additional data-specific evaluation metric based on the classification of articles as relevant or not for the downstream task.

Our main findings are:

- The improvement of preprocessing NLP models before training is limited. The most important preprocessing step is the removal of words that either appear in almost all documents or appear in only a few documents. The optimal preprocessing parameters are used as the default values for training new models.
- Automatic evaluation of NLP clustering using evaluation metrics remains a hard task. The unsupervised model might find different groupings than the one of interest to the expert. In this study, none of the unsupervised evaluation metrics aligned perfectly with the supervised clustering by the expert.
- Expert evaluation of trained NLP models remains important. From the expert evaluation, we learned that the results obtained by the Top2Vec model align best with human interpretation.
- Automated clustering of newspapers through NLP models is a complex undertaking due to the wide variety of topics covered and the articles being written for diverse audiences by numerous authors. In this regard, we have identified three key insights that could inform future NLP models. Firstly, hyperparameter tuning and preprocessing may only yield limited improvements and should be implemented when the base model's performance is already near deployment standards. Secondly, unsupervised clustering has a high chance of finding different clusters from those intended by domain experts. Lastly, expert evaluation is crucial in selecting a model that aligns abstract numerical metrics with human interpretation.

Evaluated models

For text classification (Case study 1), we evaluate both XGBoost and Fasttext. For topic modelling (Case studies 2, 3 and 4), LDA, prodLDA, neural LDA, CTM, Top2Vec and BERTopic are evaluated. The table below summarizes for each case study the models that are evaluated.

	Case study 1	Case study 2	Case study 3	Case study 4
XGBoost	✓			
Fasttext	✓			
LDA		✓	✓	✓
prodLDA		✓	✓	✓



Topic modelling and text classification models for applications within EFSA

Vandevoort B., Bex G. J., Crevecoeur J., Neven F.

neural LDA	✓	✓	✓
CTM	✓	✓	
Top2Vec	✓	✓	✓
BERTopic			✓



Table of Contents

Abstract.....	1
Graphical summary	3
Non-technical summary	4
Technical summary	6
1 Introduction	13
1.1 Background and terms of reference as provided by the requestor.....	13
1.2 Interpretation of the Terms of Reference.....	13
2 Data and Methodologies.....	15
2.1 Data	15
2.1.1 Data Case Study 1	15
2.1.2 Data Case Study 2	16
2.1.3 Data Case Study 3	17
2.1.4 Data Case Study 4	18
2.2 Methodologies.....	18
2.2.1 Word embeddings	18
2.2.2 Document Embeddings	21
2.2.3 Text Segmentation.....	28
2.2.4 Topic Modelling.....	28
2.2.5 Hyperparameter Optimization.....	34
2.2.6 Model training architecture.....	34
3 Assessment/Results.....	36
3.1 Assessment Case Study 1	36
3.1.1 Hierarchical classification methods	36
3.1.2 Document encoding	37
3.1.3 Classifiers	37
3.1.4 Segmentation and Metadata.....	37
3.1.5 Explainability.....	38
3.1.6 Proposal	38
3.2 Results Case Study 1	38
3.2.1 Models.....	38
3.2.2 Evaluation metrics	40
3.2.3 Model training	42
3.2.4 Evaluation.....	42



3.2.5 Conclusion 51

3.3 Assessment Case Study 2 51

3.4 Results Case Study 2 53

3.4.1 Ground Truth Annotation 53

3.4.2 Models..... 56

3.4.3 Evaluation Metrics..... 57

3.4.4 Optimizing for Multitopic Cluster Purity 57

3.4.5 Expert evaluation..... 67

3.4.6 Conclusion 70

3.5 Assessment Case Study 3 71

3.6 Results Case Study 3 71

3.6.1 Models..... 71

3.6.2 Evaluation Metrics..... 73

3.6.3 Optimizing for Topic Information Gain 73

3.6.4 Expert Evaluation..... 77

3.6.5 Conclusion 79

3.7 Assessment Case Study 4 79

3.8 Results Case Study 4 80

3.8.1 Models..... 80

3.8.2 Evaluation Metrics..... 81

3.8.3 Optimizing for Penalized Entropy 82

3.8.4 Optimizing for coherence 85

3.8.5 Optimizing for expert judgement 87

3.8.6 Conclusion 87

4 Conclusion 89

References 93

Abbreviations 97

Appendix A – List of questions, pillars and subquestions for APRIO classification in case study 1 99

Appendix B – Case Study 4: Example task given to the case expert to evaluate qualitative model performance 105



1 Introduction

1.1 Background and terms of reference as provided by the requestor

This contract was awarded by EFSA to: Hasselt University

Contractor: Hasselt University

Contract title: Identification of the topic modelling and classification models more suitable for application in EFSA, training methodology for proposed models, deployment as APIs, web application development, training and testing for specific case studies

Contract number: OC/EFSA/AMU/2020/02

1.2 Interpretation of the Terms of Reference

The present section is an excerpt of the section "Requested Services" of the contract OC/EFSA/AMU/2020/02 in relation to the topic modelling and text classification requirements as well as details concerning the four case studies.

Models for topic modelling and classification

- a. Training and inference code for at least three state-of-the-art model types for topic modelling (of which, one must be top2vec) and at least three state-of-the-art model types for document classification must be implemented. Their suitability for use in the EFSA context must be assessed. They shall span from quick (e.g. approximate or rule-based) to more accurate (e.g. transformer) models. Final criteria and choice of models must be discussed with EFSA before implementation.
- b. Hyperparameters must be optimized for each use case (see below) and their influence must be explained and quantified.
- c. Each model must be accompanied with
 - i. a software documentation
 - ii. a scientific description of the model along with quantitative assessment of their parameter's influence and generalizability to different datasets, if needed through simulations.

Case study: general considerations.

- a. Modelling is performed with emphasis on future reuse by statisticians and data scientists
 - i. For each case study, at least three models should be trained on the modelling/classification task and compared for their merits.
 - ii. An ensemble model may be proposed to cover the scenario where different algorithms perform better on different subsets of the inputs.



- iii. Code for individual model training and/or ensemble modelling should use the API in a way that it can be repurposed for other use cases.
- iv. For both document classification and topic modelling, performance metrics should be proposed for each label.
- v. For each case study, the influence of hyperparameters on the specific subject matter (e.g. number of topics, relatedness of documents) should be discussed quantitatively.
- vi. A discussion must elaborate on the various performance indicators and recommend the most appropriate ones for each case study and/or model.

Case study 1: Problem Formulation. About 700 EFSA scientific opinions have been manually classified according to the new Agent-Pathway-Recipient-Intervention-Outcome (APRIO)¹ problem formulation framework. This framework labels each scientific opinion in a harmonised way irrespective of the domain according to sub-questions that have been addressed, themselves grouped into high-level questions. We seek to assess the quality of current document classification algorithms with the aim of assisting humans at labelling the entire corpus of EFSA scientific opinions.

Modelling: The influence of which sections of each document contributes most to classification accuracy for each question must be studied and reported.

Case study 2: Public consultation for sugar opinion. About 700 comments were obtained by EFSA following public consultation on the sugar scientific opinion. Any two comments can be duplicates (e.g., same comment submitted by different parties, or technical duplicates), related (address an overlapping set of issues than another comment), or unrelated. The scope of this study is to group the comments in order to ease the formulation of replies. It will be calibrated on the sugar scientific opinion dataset with the aim of reusing the calibrated topic model(s) for other public consultations.

Modelling: The trained models must be example-agnostic in order to ensure portability to public consultations unrelated to a specific mandate.

Case study 3: AOP (Adverse Outcome Pathway). Recently EFSA has started to explore the use of the AOP approach to complement the more traditional approach in risk assessment based on the hazard identification and characterization. The AOP approach requires setting a pathway to harm defined by a sequence of key events that originates from a molecular initiating event and lead to adverse outcome/s. Setting the structure of the AOP frequently implies consideration of a large corpus of papers that would need to be analysed to identify possible key events in the sequence. We seek to investigate the ability of the state-of-art topic modelling techniques to support the exploration of the body of evidence to set up the putative structure of an AOP. The corpus of papers used for the AOP on developmental neurotoxicity will be used to train the model.

Modelling: The models should be example-agnostic in order to ensure portability to AOP unrelated to DNT.

¹ We refer to the published report for more background on this case study:
<https://efsa.onlinelibrary.wiley.com/doi/epdf/10.2903/sp.efsa.2022.EN-7349>



Case study 4: Beeswax topic modelling. EFSA develops tools for the detection of emerging risks in food and feed. An example of a specific food fraud incident is beeswax adulteration: it can be adulterated for financial gain and, although it is a product from apiculture, it might enter the food chain when it is introduced as honeycomb in honey pots. A total of 2276 news articles were retrieved on EMM/MEDISYS, using “beeswax” as keyword. However, beeswax can be used in other sectors, like cosmetic and food wrapping: a human manual screening of the articles identified could be timewasting. Therefore, we seek to investigate the ability of the state-of-art topic modelling techniques to support the exploration of the body of web news, to recognize different sectors in which beeswax is used and identify the articles belonging to the food topic.

Modelling: The trained models must be beeswax-agnostic in order to ensure portability to media scans unrelated to beeswax.

2 Data and Methodologies

2.1 Data

2.1.1 Data Case Study 1

The dataset consists of 748 articles that express opinions. Articles are classified according to questions they address. In total, there are 10 questions:

1. Human or animal RA
2. Nutritional assessments
3. Surveillance
4. Animal welfare
5. Efficacy
6. Emerging risks Identification
7. Identify food vehicle of infection
8. Plant pest / microbial / animal health RA
9. Environmental RA
10. Assessment of methods

Articles address one or more of these questions. Most articles address 1 or 2 questions, but some 3 to 5 questions.

Each question concerns one or more of 4 pillars. These pillars are

1. CHARACTERISATION OF RISK OR BENEFIT
2. EFFECT CHARACTERISATION
3. EFFECT IDENTIFICATION
4. EXPOSURE ASSESSMENT



However, a document that addresses a certain question does not necessarily address all the pillars that can be associated with that question. With each question and pillar, a number of subquestions are associated. The total number of subquestions is 155. Furthermore, each subquestion is associated with a single pillar and a single question.

2.1.2 Data Case Study 2

The dataset consists of 723 comments on a scientific opinion. For each comment, the submitting organization (which can be anonymous) and country are included. Analysis on the length of comments reveals that all comments are between 21 and 1999 characters long, with an average length of 1002 characters. After removing duplicate comments, we identify 560 unique comments. For this purpose, two comments are considered duplicates if the comment string matches exactly, ignoring information about the submitter (i.e., two comments can still be identified as duplicates, even if they are submitted by different organizations).

The dataset furthermore contains a number of comments that are almost identical, but contain small differences due to spelling mistakes or differences in punctuation and structure (e.g., by adding bullet points to an enumeration). We will refer to these almost identical comments as nearly-copy-pasted comment pairs. To identify these nearly-copy-pasted comment pairs, we measure the Levenshtein distance [27] between pairs of comments, using a threshold value of 20. That is, two comments are considered nearly-copy-pasted if the text of one comment can be modified into the text of the other comments while using less than 20 character-based edits, where each character-based edit is either the removal, addition or modification of a single character. A threshold of 20 is chosen to have a good balance between false positives and false negatives. In particular, choosing a value lower than 20 might miss some nearly-copy-pasted comment pairs, whereas a higher value might falsely identify short comments as nearly-copy-pasted. **Figure 1** visualizes the distribution of edit distances between all pairs of comments, indeed showing a significant drop around a distance of 20 edits. After removing all nearly-copy-pasted comments, we obtain 493 unique comments. Note that Levenshtein distance is a syntactical metric, not taking into account the semantical meaning of comments. Therefore, two comments that are semantically different because of a minor difference (e.g., starting a comment with "Why ..." instead of "How ...") are incorrectly identified as identical by this approach. As part of this project, a web application was built to allow exploration of groups of nearly-copy-pasted comments. In particular, this web application highlights differences between nearly-copy-pasted comments, thereby facilitating manual validation. For the provided dataset, manual validation did not reveal any such false positives when using 20 as a threshold value. We refer to Section 3.3 for a more in-depth discussion.

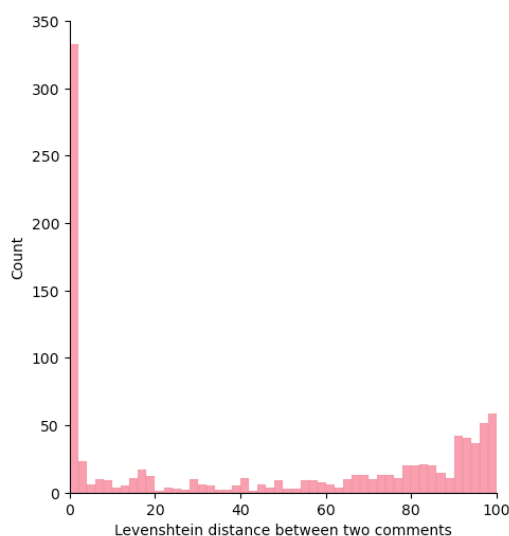


Figure 1: Distribution of Levenshtein distances for all pairs of comments (only distances up to 100 are visualized in the figure).

In addition to this dataset, a separate text document containing replies to the submitted comments is available, from which a ground truth for a topic model can be constructed. In total, this document contains 491 replies. Each reply consists of a comment, a list of organizations who submitted this comment, a textual reply and, if applicable, the changes that were made in the opinion based on this comment. Each textual reply can furthermore be separated into a number of reply points. Replies (or reply points) can refer to earlier replies (or reply points). To facilitate further analysis, we extracted this information into a machine-readable CSV file, where each row corresponds to a reply and where reply points as well as references to other replies (reply points) are extracted from the textual reply.

Analysis on these replies reveals that 162 out of the 491 replies address multiple points. Most of these replies consist of around 3 to 4 reply points, although a few of these textual replies address up to 7 points, and one reply even consists of 8 points. 372 out of the 491 replies refer to an earlier reply. Out of these replies, 216 refer to a single other reply (reply point), and the remaining replies typically refer to 2 or 3 other replies (reply points).

2.1.3 Data Case Study 3

The dataset consists of 65,983 scientific articles related to developmental neurotoxicity. For each article, metadata is available, including title and abstract. Abstracts have a length of up to 22,025 characters, with an average length of 1267 characters. Further analysis reveals that 11,283 abstracts are empty, and after removing duplicates we end up with 54,603 unique abstracts. Since only a small fraction of the dataset is identified as exact duplicates of other abstracts, we do expect only a small fraction of the abstracts to be near-duplicates of other abstracts (e.g., two identical abstracts, but with one of them missing a final newline). Analogous to the previous Case Study, we detect near-duplicates based on a Levenshtein distance with 20 edits as a threshold value, resulting in 54,491 truly unique abstracts. By choosing 20 as a threshold value, we allow for a small number of differences between duplicate abstracts, without falsely identifying shorter abstracts as identical. Note that the precise threshold value is less important here, as the purpose of this analysis is only to estimate the fraction of abstracts that appear as duplicates of other abstracts, instead of a perfect identification of all such duplicates.



A topic model based on Top2Vec (see Section 2.2.4.3) over this dataset is available as well, grouping these articles into 400 topics. Unfortunately, these topics cannot be considered a ground truth, as they are obtained through Top2Vec rather than human judgement on topic classification. Indeed, a re-run of Top2Vec with the same hyperparameter configuration would be considered a perfect topic model as it would match exactly with such a ground truth.

For each topic in the provided topic model, information on inclusion or exclusion is available (128 topics were included), obtained by a manual labelling of each topic. We recall that the main objective of this case study is not to automate this labelling, but rather to assess whether topic modelling can be used to summarize a large corpus in a reasonable number of topics, thereby facilitating the task of a domain expert to screen this corpus and to allow scoping of the literature to reveal possible topics that were unknown.

2.1.4 Data Case Study 4

A total of 2276 news articles related to beeswax are provided. For each news article, the title and text are available as part of the dataset. Analysis on the article texts reveals that articles have a length of up to 32,759 characters with an average length of 4298 characters and 7 were empty. After removing duplicates 2072 unique article texts remain. Analogous to the two previous Case Studies, we detect near-duplicates based on Levenshtein distance with 20 edits as a threshold value, identifying a small fraction of article texts as near-duplicates. After removing these, 1998 truly unique article texts remain.

Articles are manually labelled over five different possible labels:

1. Very relevant
2. Relevant
3. Not Relevant
4. Duplicates
5. Not accessible

Here, relevance is based on applicability of the article in the downstream task. Duplicates refer to articles covering the same news fact, rather than textual duplicates.

The provided data furthermore contains a topic model grouping articles into 10 topics based on LDA (see Section 2.2.4.2). This topic model and the resulting outcome is described in more detail by Rortais et al. [38]. Similar to the previous Case Study, this topic model cannot be considered a ground truth as it was obtained by LDA rather than human judgement.

2.2 Methodologies

2.2.1 Word embeddings

Word embeddings relate words to vectors that can be used in a downstream NLP task. A straightforward way to do this is by using a one-hot encoding. Assuming a vocabulary of n words, each word is encoded as an n -dimensional vector where the index corresponding to the word is set to 1 and all other values in the vector are set to 0. Such an approach clearly has a number of disadvantages, as all vectors are very sparse and a one-hot encoding cannot handle out-of-vocabulary words. Furthermore, these word embeddings do not represent semantic meaning. In particular, the distance between a pair of words is not related to their semantic similarity, since all pairs of vectors are equally distant. In this section, we present



a number of word embeddings where words are represented as dense vectors with semantic meaning. Typically, these word embeddings are derived from a large text corpus, based on the underlying idea that words with a similar meaning should appear in similar contexts (i.e., words surrounding the given word).

2.2.1.1 Word2Vec

Mikolov et al. [30] introduced Word2Vec, a tool to generate word embeddings. Word2Vec provides two models to train word embeddings from a large corpus of text. The first is called CBOW (continuous bag-of-words), and the second is called Skip-gram. Both models are trained on a prediction task, where the CBOW model predicts a word given the surrounding context, whereas the Skip-gram model predicts context-words from the given word. Through stochastic gradient descent and backpropagation, word-embeddings are changed in such a way that the outcome of the prediction task is optimized. A Python implementation of these models, including some pre-trained word embeddings on public data, is available as a part of the Gensim package.²

One of the disadvantages of these Word2Vec models is that they cannot handle out-of-vocabulary words. That is, if a word does not appear in the training vocabulary, it is not possible to derive an embedding. A second disadvantage is that, after training, every word is related to a fixed word embedding, independent of the context. Because of this, words with different possible meanings (e.g., the word "bank" in "river bank" vs "bank account") will always be represented by the same word embedding.

2.2.1.2 GloVe

The GloVe model [33] derives word embeddings from a corpus based on a word-word co-occurrence matrix. For each pair of words i and j , this matrix contains the number of times word i occurs in the context of word j . During training, this matrix is constructed first, and the actual training of word embeddings is now based on this matrix instead of the original text corpus. Because of this, the GloVe model needs to go through the entire corpus only once (to construct the matrix). Experiments furthermore show that the GloVe model outperforms the Word2Vec models on word similarity tasks. An implementation written in C as well as pre-trained GloVe embeddings are available online.³

The disadvantages of GloVe are similar to those of Word2Vec models: embeddings are not context-sensitive, and the model cannot handle out-of-vocabulary words.

2.2.1.3 FastText

Bojanowski et al. [7] present FastText, an improvement over the Word2Vec model presented by Mikolov et al. [30] by including the internal structure of words. This FastText model represents each word as a bag of n -grams (also referred to as subwords). For example, consider the word "where". Then, all n -grams for $n=3$ are "whe", "her", "ere". During training, the model learns embeddings for all these subwords, and averages them to construct an embedding for the word itself. The training task itself is similar to the Word2Vec models, with

² <https://radimrehurek.com/gensim/models/word2vec.html>

³ <https://nlp.stanford.edu/projects/glove/>



Vandevoort B., Bex G. J., Crevecoeur J., Neven F.

the notable difference that now embeddings for subwords are trained instead. After training, the model derives word embeddings for words by averaging the embeddings of the subwords. A Python implementation to train word representations from a given corpus, as well as pre-trained models over large datasets are available.⁴

The advantage of FastText over Word2Vec is that this model is able to derive word embeddings for out-of-vocabulary words. Indeed, even if the word itself was not present in the training corpus, it is still possible to derive an embedding for it by averaging the embeddings of its subwords. Similar to Word2Vec, the word embeddings produced by FastText are not context-sensitive, which is problematic for words with multiple possible meanings depending on the context.

2.2.1.4 ELMo

ELMo (Embeddings from Language Models) [34] provides word embeddings based on a pre-trained model over a large text corpus. The architecture of an ELMo model is based on a bidirectional language model (biLM), combining forward LSTMs predicting a word i from words occurring before i with backward LSTMs predicting this word i from words occurring after i . During inference, the input of an ELMo model to derive word embeddings differs from the input for models such as Word2Vec, GloVe and Fasttext. Whereas the latter models only require the word itself to derive a word embedding, ELMo expects a sequence of words and outputs a corresponding sequence of word embeddings, where word embeddings for each word are context-sensitive: they are influenced by the other words in the input. Furthermore, ELMo is able to derive word embeddings for out-of-vocabulary words as well, by taking into account the context of the word as well as subwords (cf. Fasttext) of this word. Multiple pre-trained ELMo models of different sizes are available online.⁵ Most of these models are trained on the One Billion Word Benchmark [13], a dataset containing almost one billion words of training data.

2.2.1.5 BERT

BERT (Bidirectional Encoder Representations from Transformers) [16] is a language representation model leveraging pre-training to facilitate downstream NLP tasks. In particular, a pre-trained BERT model can be used for a specific task by adding a single task-specific output layer to the model, followed by a fine-tuning step where weights are updated based on the task at hand. This should be contrasted with feature-based models such as ELMo, where the pre-trained output of the model is merely used as an additional feature for a task-specific model. The BERT model is based on a multi-layer bidirectional transformer encoder architecture [47], where self-supervised pre-training is based on the Masked Language Model (MLM). That is, the model is trained on the objective of predicting randomly masked words in a given sentence. Since the architecture is bidirectional, BERT can predict masked words based on words before as well as after it in the sentence. This deep bidirectional architecture should be contrasted with e.g. ELMo, where the model consists of a shallow concatenation of separate forward and backwards LSTMs.

⁴ <https://fasttext.cc/>

⁵ <https://allenai.org/allennlp/software/elmo>



The BERT model uses WordPiece embeddings [51] as input to the model, thereby supporting out-of-vocabulary words. Similar to ELMo, a pre-trained BERT model always expects a sentence⁶ as input and provides the corresponding context-sensitive word embeddings as output. Experiments show that BERT models pre-trained on a large text corpus (more than 3 million words) can outperform state-of-the-art task-specific architectures on a number of downstream NLP tasks. Pre-trained BERT models of different sizes are available online.⁷

2.2.2 Document Embeddings

Similar to word embeddings, a document embedding represents a document (i.e., a continuous piece of text of arbitrary length) as a fixed size vector. It is furthermore desirable that these document embeddings capture semantic information by assigning vectors to documents in such a way that the distance between these vectors represents semantic similarity between documents. These document embeddings can then be applied in a range of downstream tasks, such as document classification, topic modelling, query answering, or dense information retrieval.

In the literature, document embeddings are often referred to as sentence embeddings, where the term “sentence” should be interpreted more broadly as an arbitrary span of text (possibly consisting of multiple linguistic sentences), rather than the narrower linguistic interpretation. Throughout this document, we will use the terms document embedding and sentence embedding interchangeably.

2.2.2.1 TF-IDF vectors

Term frequency – inverse document frequency (TF-IDF) can be used to create vectors representing documents. Given a term t and document d from a corpus D , the TF-IDF score for t and d is proportional to the frequency of t in d , and inversely proportional to the number of documents in D that contain t . More formally,

$$\text{TF-IDF}(t, d, D) = \text{TF}(t, d) \cdot \text{IDF}(t, D)$$

with

$$\text{TF}(t, d) = \frac{f_{t,d}}{|d|}$$

and

$$\text{IDF}(t, D) = \log \frac{|D|}{|D_t|}$$

where $f_{t,d}$ represents the number of occurrences of t in d , $|d|$ is the total number of terms in d (with multiple occurrences of the same term included in this count), and D_t is the subset of documents in D in which t occurs.

Assuming the documents are over a vocabulary of n terms, each document can now be represented by an n -dimensional TF-IDF vector where the value on each index is the TF-IDF

⁶ A “sentence” should be interpreted as an arbitrary span of text, rather than the narrower linguistic interpretation.

⁷ <https://github.com/google-research/bert>

score for the corresponding term corresponding to this index. An immediate drawback of this approach is that it cannot handle out-of-vocabulary words.

2.2.2.2 Averaging word embeddings

A straightforward (and rather naive) approach to construct a document embedding from a document is by taking the average word embedding for all words occurring in it. The advantage of this approach is that no labelled training data is needed to infer document embeddings. One can simply choose a self-supervised word embedding model such as Word2Vec, Fasttext or GloVe and train them on unlabelled data, or simply use one of the many available pre-trained word embeddings. The disadvantage of the such obtained document embeddings is that they perform rather poorly on downstream NLP tasks. Experiments [36] show for instance that document embeddings derived by averaging GloVe embeddings are outperformed by more involved document embedding models on downstream tasks such as predicting semantic textual similarity on the STS benchmark [12].

2.2.2.3 Doc2Vec

The Doc2Vec model [25] extends the approach presented by the Word2Vec model by adding a paragraph vector during training. More specifically, Doc2Vec associates each word in the corpus with a word vector, and each document as a whole with a paragraph vector. Similar to Word2Vec, Doc2Vec presents two training models; PV-DM (distributed memory) and PV-DBOW (distributed bag of words). The former model is trained on predicting the next word in a text, using the previous words and paragraph vector as input. The latter model is similar to the Skip-gram model of Word2Vec, and is trained on the objective of predicting randomly sampled words from the text, using only the paragraph vector as input. After training, the paragraph vector for each document can be used as the desired document embedding.

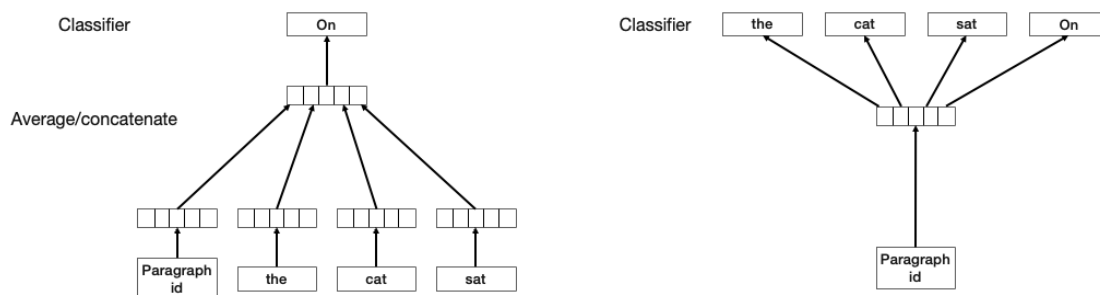


Figure 2: Architecture of the Doc2Vec PV-DM (left) and PV-DBOW (right) models during training.

A Python implementation of these models is available as part of the Gensim package.⁸ By default, the PV-DM model is provided to train word embeddings and paragraph vectors at the same time. If the PV-DBOW model is used, word embeddings can be trained simultaneously with the paragraph vectors, but this is optional. Otherwise, randomly initialized word embeddings are used to train paragraph vectors instead, resulting in a faster training.

The main advantage of Doc2Vec is that the training is self-supervised, requiring only unlabelled text. Le and Mikolov [25] evaluate Doc2Vec over datasets ranging from almost

⁸ <https://radimrehurek.com/gensim/models/doc2vec.html>



12,000 sentences to 100,000 sentences. Experiments on the STS benchmark show that the PV-DBOW model indeed outperforms methods based on averaging word embeddings [12].

2.2.2.4 Fine-tuning BERT for document similarity

As discussed in Section 2.2.1.5, a pre-trained BERT model can be fine-tuned for a number of downstream NLP tasks, including the inference of document similarity for a given pair of documents [16]. The fine-tuning of the BERT model consists of adding an additional output layer on top of the pre-trained BERT model returning the document similarity for a given pair of documents, followed by training the model on task-specific training data. In this case, the training data consists of sentence pairs annotated with a similarity score.

The main disadvantage of these fine-tuned models is that they do not provide a document embedding for each document in the corpus. Instead, they expect a pair of documents and directly infer a similarity score. As a consequence, computing the similarity for each pair of documents requires a number of inferences that is quadratic in the number of documents, which is problematic for larger text corpora. A second disadvantage of this method is that a labelled training dataset is required.

2.2.2.5 Universal Sentence Encoder

Cer et al. [11] provide two models to infer sentence embeddings that can be used as input for downstream NLP tasks, referring to these models as Universal Sentence Encoders. The first model is based on the encoder part of the transformer architecture [47], whereas the second model uses a deep averaging network (DAN) [23] to infer sentence embeddings. The former model provides higher accuracy at the cost of computational complexity, whereas the latter model trades in accuracy to obtain a compute time linear in the length of the input sequence. Both models are trained using multi-task learning, where multiple downstream tasks are used during training to improve the general applicability of inferred sentence embeddings after the model is trained.

Pre-trained Universal Sentence Encoders based on the DAN model are available online.⁹ Although these models are trained on multiple downstream tasks, it is unclear how well these models perform on domain-specific text.

2.2.2.6 Sentence-BERT

Reimers and Gurevych [36] present Sentence-BERT (SBERT), a modification of BERT models to derive semantically meaningful sentence embeddings. The crux of their approach relies on feeding sentence pairs into different BERT models with tied weights, rather than a single BERT model during fine-tuning (cf. Section 2.2.2.4). To derive fixed-size sentence embeddings from variable length text, the SBERT model adds a pooling layer on top of a pre-trained BERT model. Three different pooling strategies are explored: using the output of the CLS-token, or by computing the mean or maximum of all output vectors. The precise network structure for fine-tuning depends on the available training data:

- If sentence pairs are annotated with a class, a siamese network is used (i.e., two BERT models with tied weights), and the resulting embeddings are used as the input for a softmax classifier.

⁹ <https://tfhub.dev/google/universal-sentence-encoder/4>



- If sentence pairs are annotated with a similarity score, a siamese network is used as well, but the cosine-similarity between the inferred embeddings is used for fine-tuning instead of a softmax classifier.
- The third method assumes training data consisting of triples of sentences, referred to as anchor, positive sentence and negative sentence. The training objective for this approach requires that the distance between the inferred embeddings of the anchor and positive sentence is smaller than the distance between the embeddings of the anchor and the negative sentence.

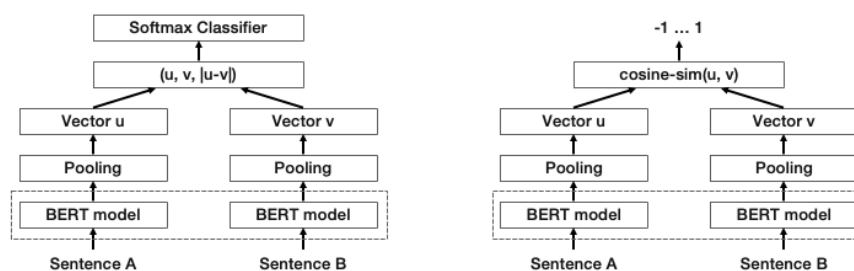


Figure 3: Sentence-BERT architecture for training based on sentence-pair classification (left) and similarity score prediction (right). In both architectures, the two BERT models in share the same weights.

Experiments [36] on the STS benchmark show that SBERT fine-tuned on a combination of the SNLI and MultiNLI datasets (consisting of 570,000 and 430,000 annotated sentence pairs, respectively) outperforms other unsupervised document embedding approaches, such as averaging GloVe embeddings or BERT embeddings, as well as the Universal Sentence Encoder model. The reported Spearman rank correlation on the STS benchmark is 79.23 for SBERT-NLI-large (i.e., based on the pre-trained BERT-large model), which should be contrasted with the results obtained by averaging GloVe embeddings (58.02), averaging BERT embeddings (46.35) or using a Universal Sentence Encoder (74.92). When SBERT is fine-tuned on a combination of NLI and the training data included with the STS benchmark, the Spearman rank correlation further increases (86.10 for SBERT-NLI-STsb-large). These results are only slightly inferior to training a BERT model more directly as described in Section 2.2.2.4. A Spearman rank correlation of 88.77 is reported when BERT-large is fine-tuned on the combination of the NLI and STS datasets. Further experiments on the different pooling strategies show that the chosen pooling strategy barely influences the final result on the NLI dataset. On the STS dataset, taking the CLS-token or computing the mean is recommended.

Different pre-trained SBERT models are available online.¹⁰ These models are fine-tuned on large sets of available training data (more than 1 billion training pairs). Their performance on domain specific text is however unclear.

2.2.2.7 SimCSE

The Simple Contrastive Sentence Embedding (SimCSE) framework [19] derives sentence embeddings by fine-tuning a BERT model. Contrasting earlier approaches, the fine-tuning technique can rely on unlabelled data as well. During fine-tuning, sentence embeddings can

¹⁰ https://www.sbert.net/docs/pretrained_models.html



be tuned by applying contrastive learning objective. This objective requires positive and negative sentence pairs, with the objective of moving embeddings of positive pairs closer together, whereas embeddings of negative pairs should be pushed further apart. The model can now either be trained in a supervised manner (assuming the training data provides these positive and negative pairs), or in an unsupervised manner (i.e., using only unlabelled sentences as input). The basic idea for the latter approach is to consider pairs consisting of two different sentences as negative pairs, and pairs where the same sentence is used twice as positive pairs. For this to work, the sentence is passed through the sentence embedding model twice, where both passes use a different (random) dropout mask (that is, a randomly chosen subset of neurons in the network is disabled). These random dropout masks lead to embeddings that are different, but still closely related to each other when the same sentence is encoded twice.

Experiments on the STS benchmark report Spearman rank correlations of 76.85 and 84.25 for the unsupervised and supervised SimCSE models, respectively. Both variants were trained starting from the pre-trained BERT-base model. The unsupervised SimCSE model is trained on 1,000,000 randomly sampled sentences from English Wikipedia, whereas the supervised SimCSE model was trained on a combination of the MNLI and SNLI datasets (314,000 labelled sentence pairs). Notice in particular that the supervised model is not trained on STS data. When comparing with SBERT it can be concluded that the unsupervised SimCSE model performs on par with the corresponding SBERT model on the STS benchmark: the SBERT model trained on the NLI datasets starting from BERT-base is reported to result in a Spearman rank correlation of 77.03 [36].

A pre-trained SimCSE model based on the supervised approach is available online.¹¹

2.2.2.8 Sentence-T5

Text-to-Text Transfer Transformer (T5) [35] is a text-to-text transformer model based on an encoder-decoder transformer architecture, which should be contrasted with BERT, an encoder-only model. Based on T5, Ni et al. [32] present Sentence T5 (ST5), a model to derive sentence embeddings from T5. They propose three different strategies:

- Encoder-only first: the sentence embedding is based on the first output vector of the encoder. The decoder is ignored.
- Encoder-only mean: the sentence embedding is based on the average of all output vectors of the encoder. Similar to the previous strategy, the decoder is ignored.
- Encoder-decoder first: the first output vector of the decoder is used to derive the sentence embedding.

Contrastive learning (see Section 2.2.2.7) is used as a training objective to fine-tune the ST5 model. Contrasting SimCSE [19], Ni et al. [32] assume labelled training data with positive and negative sentence pairs, and do not explore techniques to fine-tune ST5 from unlabelled data.

Experimental analysis shows that the encoder-only approaches have strong transfer performance, whereas the encoder-decoder approach perform better on textual similarity: Spearman rank correlations of up to 86.82 are reported for the STS benchmark when the ST5

¹¹ <https://huggingface.co/princeton-nlp/sup-simcse-bert-base-uncased>

model based on an encoder-decoder architecture and fine-tuned on the NLI dataset is used to derive sentence embeddings.

Pre-trained ST5 models are available on TensorFlow Hub.¹²

2.2.2.9 TSDAE

Wang, Reimers and Gurevych [48] present a technique for unsupervised sentence embedding learning using a Transformer-based Sequential Denoising Auto-Encoder (TSDAE). During training, the TSDAE model takes a sentence with noise (e.g. by deleting or swapping words) as input, and is tasked with reconstructing the original sentence. The architecture of TSDAE is given in Figure 4. Notice in particular how the TSDAE model first applies an encoder with pooling layer to convert the given sentence into a fixed-size sentence embedding, where a pre-trained BERT model can be used for the encoder. This sentence embedding is passed to the decoder to predict the sentence. After training, the encoder with pooling layer is used to infer sentence embeddings.

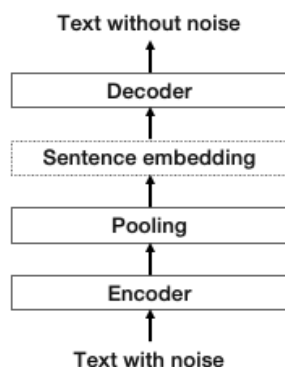


Figure 4: Architecture of the TSDAE model during training.

Experiments on multiple benchmarks explore the different possible applications of TSDAE: unsupervised learning, domain adaptation and pre-training. Unsupervised learning assumes that only unlabelled data from the target domain is used to train the TSDAE model. Domain adaptation combines this unlabelled data from the target domain with labelled data from a different source (e.g. the NLI dataset). With these two data sources, it is either possible to train the model with TSDAE over the domain-specific unlabelled data first, followed by supervised learning over the labelled data, or the other way around. The pre-training approach assumes a larger set of unlabelled data as well as a smaller set of labelled data from the target domain to pre-train the TSDAE model with the unlabelled data, followed by supervised learning over the labelled training data to further fine-tune the sentence embeddings.

For unsupervised training, they conclude that the frequently used STS benchmark is not necessarily a good performance indicator for these unsupervised models. In particular, TSDAE is outperformed by other unsupervised methods such as SimCSE on the STS benchmark (a Spearman rank correlation of 66.0 is reported for TSDAE). However, on other domain-specific benchmarks these approaches are performing notably worse than TSDAE. In fact, an out-of-the-box pre-trained supervised model (e.g. an SBERT model trained on NLI) often

¹² <https://tfhub.dev/google/collections/sentence-t5/1>

outperforms these unsupervised models. Based on further experiments on dataset size, they recommend a dataset of at least 10,000 sentences for unsupervised training. Regarding domain adaptation, experimental data suggests that training on the unlabelled domain-specific data first, followed by supervised fine-tuning on the labelled data set gives the best results.

2.2.2.10 GPL

Wang et al. [49] present Generative Pseudo Labeling (GPL), an unsupervised domain adaptation approach to improve dense retriever models. In brief, a dense retriever is a model that embeds queries and documents in a shared vector space in such a way that documents related to a query are placed close to this query. Figure 5 gives a high-level overview of the three different steps GPL uses to fine-tune a dense retriever. During the first step, T5 is used to generate potential queries from the given documents. A pre-trained dense retriever then collects a number of negative samples for each query. Afterwards, a pre-trained cross-encoder labels these pairs to indicate how related the queries and documents are. These labels are then used to fine-tune the dense retriever.

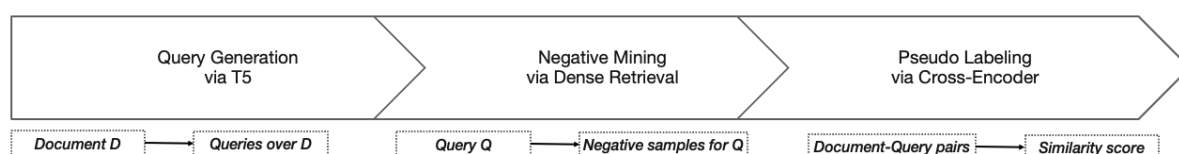


Figure 5: Schematic overview of the different steps of the GPL unsupervised domain adaptation strategy.

Although the dense retriever can be used to provide embeddings for documents, it is unclear how useful these embeddings are for detecting semantic similarity between documents. In particular, experimental analysis of GPL is focused around dense information retrieval, and does not, for example, include performance analysis on the STS benchmark.

2.2.2.11 SPECTER

SPECTER (Scientific Paper Embedding using Citation-informed TransformERs) [15] is a model that can be applied to infer document embeddings from scientific papers. The model is trained starting from SciBERT [3] (a BERT model pre-trained on scientific documents). The training data for SPECTER consists of 146,000 papers. For each paper, a number of related and unrelated papers is selected (where related papers are based on citations). These combinations are then used as the input to train the SPECTER model on the triplet loss objective, as visualized in Figure 6. Note in particular that citations are only considered during training. After training, only the title and abstract are required to infer an embedding. A pre-trained SPECTER model is available online¹³.

¹³ <https://huggingface.co/allenai/specter>

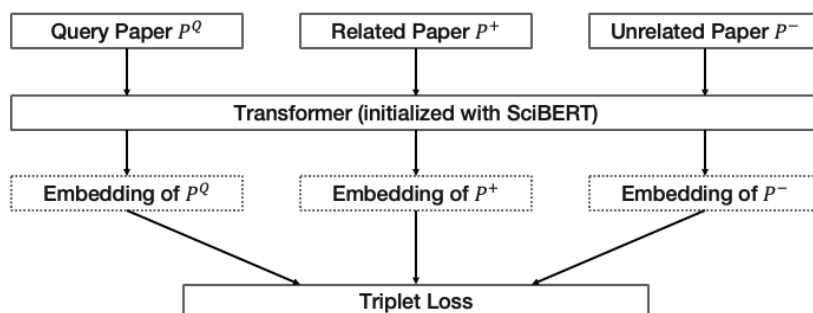


Figure 6: Schematic overview of the SPECTER model during training. Given a paper as well as a positive instance (a related paper) and a negative instance (an unrelated paper), the objective is to minimize the distance from the given paper to the positive instance while maximizing the distance to the negative instance.

2.2.3 Text Segmentation

Text segmentation is the task of splitting a larger chunk of text (e.g., a news article) into coherent parts by detecting boundaries within this text. In this section, we discuss two possible approaches: structural segmentation (based on syntactic properties of the text at hand) and semantic segmentation (based on detecting semantic boundaries within the text).

2.2.3.1 Structural segmentation

A straightforward segmentation method is splitting text on syntactic properties of the text at hand, such as sentences, line breaks or paragraphs. Depending on the corpus, more involved rule-based methods such as regular expressions can be applied to identify domain-specific segment boundaries. The disadvantage of these methods is that they do not capture semantic information.

2.2.3.2 Semantic segmentation

Semantic segmentation identifies segment boundaries by detecting a semantic shift within the text, rather than relying on syntactic features of the text itself. A popular algorithm for semantic segmentation is the TextTiling algorithm [22]. This approach is based on patterns in word cooccurrences. More specifically, the TextTiling algorithm assumes that a specific set of lexical items (i.e., words) is being used extensively while one topic is being discussed. When the topic changes, this set of frequently used lexical items changes as well. By detecting these shifts, topic boundaries within a text can be detected.

2.2.4 Topic Modelling

Given a larger corpus of text documents, topic modelling is the NLP task of grouping related documents together into topics, followed by finding a good representation for each topic. In practice, these topic representations are often a smaller group of words with associated weights, visualized via a word cloud. In this section, we will discuss several topic modelling approaches, ranging from baseline approaches such as Non-negative Matrix Factorization (NMF) and variations of Latent Dirichlet Allocation (LDA) to state-of-the-art techniques such as Top2Vec and BERTopic that leverage document embeddings to cluster related documents.

2.2.4.1 Non-negative Matrix Factorization



Vandevoort B., Bex G. J., Crevecœur J., Neven F.

Non-negative Matrix Factorization (NMF) [26] can be applied to infer topics and topic representations from documents. Assuming the corpus is over a vocabulary of n words, each document is represented as an n -dimensional TF-IDF vector (cf. Section 2.2.2.1). A corpus of d documents can now be represented as a $n \times d$ matrix V , where each column is the n -dimensional vector corresponding to a document. To derive r topics and related topic representations from this word-document matrix V , V is factorized into two smaller non-negative matrices W and H , where W is a $n \times r$ word-topic matrix and H is a $r \times d$ topic-document matrix.

A Python implementation of NMF for topic modelling is available as part of the Gensim package.¹⁴ This implementation is based on an online algorithm, allowing the retrieval of topic distributions for unseen documents, as well as iterative updates to the model based on new text corpora.

The advantage of NMF is that no labelled training data is needed. One important consideration of this approach is that the number of topics r is assumed to be given as part of the input. If the expected number of topics is not known in advance, an optimal value for r is typically determined through trial-and-error by trying different values for r and evaluating each resulting topic model (see Section 2.2.4.4 for an overview of evaluation metrics).

2.2.4.2 Latent Dirichlet allocation

The Latent Dirichlet Allocation (LDA) model presented by Blei, Ng and Jordan [5] is a generative probabilistic model based on the assumption that each document in the corpus is a random mixture of latent topics, where each topic is a probability distribution over the words in the vocabulary. Documents are generated by randomly sampling words according to these distributions. To derive topics and topic representations from a given corpus of documents, the LDA approach tries to reconstruct these latent topics and probability distributions from the corpus.

Similar to the approach based on NMF, this approach requires no labelled training data, but assumes that the number of topics is given as part of the input. The Gensim package provides an implementation¹⁵ that is able to infer topic distributions for unseen documents after the model is trained on a training corpus. This implementation furthermore allows to update a trained model based on new documents.

Srivastava and Sutton [43] present Neural LDA, a modification of LDA leveraging neural networks. This approach trains an inference network that directly maps each document (represented as a bag-of-words) to its distribution. To illustrate how straightforward it is to apply their neural network approach to other topic models, they provide a modification of Neural LDA where the distribution over words is a product of experts rather than a mixture model, referring to this model as ProLDA.

Note that Neural LDA and ProLDA still assume a bag-of-words representation as input. This representation of documents has a number of limitations: contextual information such as word order is lost, and semantic relatedness between words is not taken into account. To solve these issues, Bianchi et al. [4] use the sentence embeddings inferred by a pre-trained SBERT model (see Section 2.2.2.6) instead of a bag-of-words representation as input for a

¹⁴ <https://radimrehurek.com/gensim/models/nmf.html>

¹⁵ <https://radimrehurek.com/gensim/models/ldamodel.html>



ProdLDA model, resulting in a Contextualized Topic Model (CTM). To illustrate the advantages of CTM, they show that CTM is able to do zero-shot cross-lingual topic modelling. That is, a topic model trained on a corpus in one language can be used to infer topics for unseen documents in another language, without additional training (assuming the underlying SBERT model is a multilingual model trained on both languages).

Python implementations of NMF, LDA, Neural LDA, ProdLDA and CTM are bundled in the OCTIS ("Optimizing and Comparing Topic Models Is Simple") framework [45, 46], available online.¹⁶ This framework furthermore provides dataset pre-processing, as well as evaluation metrics and Bayesian Optimization to evaluate trained topic models and optimize hyperparameter configurations for a given dataset and evaluation metric.

2.2.4.3 Clustering Document Embeddings

More recent state-of-the-art techniques for topic modelling such as Top2Vec [1] and BERTopic [21] leverage document embeddings to cluster documents into topics and derive sensible topic representations. Before discussing the details of these models, we first give a high-level overview of the three different steps required to go from documents to topics and topic representations: inferring document embeddings, clustering and finding topic representations.

Step one: From documents to document embeddings

The first step leverages a document embedding model to convert documents to vectors in an n -dimensional space, where the distance between each pair of document vectors is related to the semantic similarity between these two documents. Depending on the size of the corpus and the presence or absence of labelled training data (e.g., document pairs annotated with a similarity score), one might opt to train a document embedding from scratch or fine-tune an existing embedding using either a supervised or self-supervised training approach, or simply opt for a pre-trained off-the-shelf document embedding. We refer to Section 2.2.2 for an overview of document embedding models and techniques to train them.

Step two: Clustering document embeddings

Once document vectors are inferred, a conventional clustering technique can be applied to group these data points into clusters. Important considerations when choosing an appropriate clustering algorithm are performance on large datasets, the ability to handle outliers or noise, the ability to detect the optimal number of clusters (rather than expecting this number as a parameter), as well as the assumptions on the underlying data (e.g., globular clusters vs. arbitrarily shaped clusters). For example, the popular K-Means approach is able to handle large datasets, but requires the number of clusters to be known upfront, assumes globular clusters and does not handle outliers. For topic modelling, K-Means is therefore expected to be a poor choice: the number of topics in a dataset is usually not known upfront, and the vectors of documents belonging to a single topic do not necessarily have a globular shape. Depending on the dataset, outliers (i.e., documents that are not closely related to other documents in the corpus) are often expected.

Based on these observations, HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) [9, 10] is a clustering method more suited to cluster document embeddings. HDBSCAN is an extension of DBSCAN [18], building on the assumption that

¹⁶ <https://github.com/MIND-Lab/OCTIS>

www.efsa.europa.eu/publications



clusters are dense regions, thereby inherently dealing with arbitrary shaped clusters as well as outliers. Contrasting DBSCAN, HDBSCAN is furthermore able to detect clusters of varying density. The documentation of HDBSCAN provides a more detailed comparison of different clustering algorithms.¹⁷

Since document embeddings have a high number of dimensions, it is recommended to reduce the dimensionality before clustering. UMAP (Uniform Manifold Approximation and Projection) [29] is a dimensionality reduction algorithm that can be applied before clustering such that the clustering algorithm is applied over a low-dimensional space instead. It is furthermore possible to apply UMAP after clustering to reduce to two dimensions, thereby facilitating visualizations that can be used for manual validation of cluster quality.

Step three: Deriving topic representations

The previous step groups documents into topics, but does not provide an explanation why these documents are related. The third step therefore consists of finding qualitative topic representations. These topic representations are often a set of words with associated weights. One approach to derive topic representations is based on the words appearing in documents belonging to each topic. It should be noted that words frequently occurring in documents belonging to the same topic are not necessarily good topic representations (think for example of stop words such as "the", "a", "is" ...). Instead, more qualitative topic representations can be obtained by considering words that appear frequently within the topic, but are not frequently occurring in documents outside this topic (cf. BERTopic, see below for a detailed discussion). Alternatively, if the chosen document embedding shares the embedding space with a word embedding, topic representations can be derived more directly by including words with a word embedding close to the document embeddings of documents in the topic (cf. Top2Vec, see below for a detailed discussion).

Top2Vec

Angelov [1] presents Top2Vec, an algorithm for topic modelling based on a joint word and document embedding. The Doc2Vec model (see Section 2.2.2.3) is used to jointly learn word embeddings and document embeddings. Before clustering using HDBSCAN, a dimensionality reduction based on UMAP is applied. Based on experimental analysis, 5 dimensions is recommended to obtain the best results. Afterwards, the centroid of each topic in the original document embedding space is calculated, and this vector is referred to as the topic vector. A topic representation of each topic is constructed by taking the words with an embedding close to the topic vector (recall that the document embedding space is shared with a word embedding space). The weight of each word relative to the topic is based on the distance between the word embedding and the topic vector, where words with a shorter distance to the topic vector have a higher contribution to the topic representation.

An implementation of Top2Vec is available online.¹⁸ It should be noted that this implementation allows document embeddings different from Doc2Vec as well, including pre-trained Universal Sentence Encoders (see Section 2.2.2.5) and SBERT models (see Section 2.2.2.6).

BERTopic

¹⁷ https://hdbscan.readthedocs.io/en/latest/comparing_clustering_algorithms.html

¹⁸ <https://github.com/ddangelov/Top2Vec>

Grootendorst [21] presents BERTopic, a framework for topic modelling. Similar to Top2Vec, BERTopic leverages a document embedding model to infer document vectors from documents, followed by UMAP and HDBSCAN to cluster semantically related documents into topics. Topic representations are derived from the words appearing in each document based on class-based TF-IDF. The crux of this metric is that words frequently occurring in the topic documents, but not frequently occurring in the rest of the corpus, are good candidates for the topic representation. More formally, each topic is represented by a class, which is a single document obtained by concatenating all documents belonging to the topic. Then, the score $W_{t,c}$ for a term t in a class c is calculated as follows:

$$W_{t,c} = tf_{t,c} \times \log\left(1 + \frac{A}{tf_t}\right)$$

Where $tf_{t,c}$ is the frequency of term t in class c , A is the average number of words in a class and tf_t is the frequency of term t across all classes.

An implementation of BERTopic is available online.¹⁹ On top of the topic modelling algorithm discussed above, this implementation includes other features such as different visualizations and dynamic topic modelling techniques to detect topic evolutions over time. Similar to Top2Vec, this implementation includes a number of pre-trained document embedding models, but allows to include custom document embeddings as well.

2.2.4.4 Evaluation metrics

To quantitatively assess the quality of topics for a given set of documents, an evaluation metric is needed. In this section, we give an overview of different evaluation metrics proposed in literature.

Topic information Gain

To quantitatively measure whether topic representations correctly represent the documents assigned to each topic, Angelov [1] introduces topic information gain. The metric is based on probability-weighted amount of information (PWI), and is calculated as follows:

$$PWI = \sum_{d \in D} \sum_{t \in T} \sum_{w \in W_t} P(d|w) P(t|d) \log\left(\frac{P(d,w)}{P(d)P(w)}\right)$$

Where D is the set of documents, T is the set of topics, W_t is the set of words in the topic representation of topic t , $P(d|w)$ is the conditional probability of document d given word w , $P(t|d)$ is the probability of document d belonging to topic t , $P(d,w)$ is the joint probability of document d and word w , and $P(d)$ and $P(w)$ are the probabilities of document d and word w , respectively. When a hard clustering is performed (i.e., each document is either assigned to a topic or not), $P(t|d)$ is 1 if the document d is assigned to topic t , and 0 otherwise.

At the time of writing, a reference implementation of PWI is not yet included in the Top2Vec framework.²⁰

¹⁹ <https://github.com/MaartenGr/BERTopic>

²⁰ <https://github.com/ddangelov/Top2Vec/issues/158>

Topic Coherence

Topic coherence evaluates how well words within a topic representation are coherent. A frequently used metric for topic coherence is normalized pointwise mutual information (NPMI) [8], since experimental analysis revealed that NPMI closely reflects human judgement [24]. For a pair of words w_i and w_j , the NPMI score for this pair is calculated as follows:

$$\text{NPMI}(w_i, w_j) = \frac{\log \frac{P(w_i, w_j)}{P(w_i)P(w_j)}}{-\log P(w_i, w_j)}$$

where $P(w_i)$ and $P(w_j)$ are the probabilities of respectively words w_i and w_j to occur, and $P(w_i, w_j)$ is the probability of words w_i and w_j to occur together. To obtain an NPMI score for a topic, this pairwise NPMI score can be summed or averaged over all pairs of words in the topic representation.

Contrasting topic information gain, topic coherence only considers the words in each topic representation, thereby ignoring the assignment of documents to topics. An implementation is available as part of the Gensim package.²¹

Topic Diversity

Topic diversity expresses how diverse different topics are. A straightforward metric proposed by Dieng et al. [17] is the percentage of unique words over all topic representations combined. A score of 1 indicates that no two topic representations share the same word, whereas a lower score indicates that different topics share common words, and are therefore covering the same underlying concept.

Similar to topic coherence, topic diversity only considers the words in each topic representation, thereby ignoring the assignment of documents to topics. An implementation is available as part of the OCTIS framework.²²

Cluster Purity

Assuming a ground truth, cluster purity [52] indicates how pure the given clusters are, measuring up to what level each cluster represents only a single actual class. More formally, assume a ground truth partitions the dataset consisting of n data points in ℓ different classes. Then, the purity of a cluster S_r of size n_r is calculated as follows:

$$\text{Purity}(S_r) = \frac{1}{n_r} \max_{i \in [1, \ell]} (n_r^i)$$

with n_r^i the number of datapoints in S_r that are in class i . Intuitively, the cluster purity of S_r is the maximal fraction of datapoints in S_r assigned to the same class. The overall purity for an obtained clustering is now obtained by taking a weighted sum:

$$\text{Purity} = \sum_{r=1}^k \frac{n_r}{n} \text{Purity}(S_r)$$

²¹ <https://radimrehurek.com/gensim/models/coherencemodel.html>

²² <https://github.com/MIND-Lab/OCTIS#available-metrics>



where k is the number of clusters obtained.

Contrasting most of the previous evaluation metrics for topic modelling, cluster purity only measures how well a topic model partitions the documents into groups relative to a ground truth, and is not influenced by the chosen topic representation.

2.2.5 Hyperparameter Optimization

Each of the NLP models trained in this study has a number of parameters that can be tweaked to the data set at hand. These tweakable parameters are called hyperparameters and the process of finding the optimal hyperparameters is called hyperparameter optimization [50]. In hyperparameter optimization, first a grid of candidate parameters is defined, and the model is trained multiple times each time with a different candidate for the hyperparameters. An evaluation metric from Section 2.2.4.4 is used to select the best model. For the NLP models considered in this project the number of tuning parameters can be large, which makes evaluating all parameter combinations practically unfeasible. We, therefore, opt for a Bayesian search strategy. In Bayesian optimization [2, 42, 50], the model is first evaluated on a random sample of parameters. Subsequently, when enough information is collected, consequent parameters are chosen by a Bayesian algorithm based on the results of the previous points. Regions that previously produced good results are further explored, while regions with known bad performance are skipped. As a result, this method converges more rapidly to a good solution without having to evaluate the full grid.²³ We use the implementation in scikit-learn²⁴ with gaussian process as the learning process.

As NLP models are stochastic, refitting the same model with the same parameters will still result in a different result and metrics. We minimize this effect in our search strategy by averaging the metrics over 5 model fits with the same parameter set.

2.2.6 Model training architecture

An NLP platform has been developed on Azure to facilitate the training and maintenance of NLP models for this project. The Azure ML service was utilized for model training, while model tracking was performed using MLFlow. The NLP models for case studies 1, 2, and 4 were trained on a four-core, 14GB RAM instance (Standard_DS3_v2). However, due to insufficient memory capacity, the AOP models for case study 3 were trained on an eight-core, 56GB RAM instance (Standard_D13_v2). In particular, the large dataset for case study 3 (cf. Section 2.1.3) requires more memory to be available, as the Python packages used for training (e.g. Top2Vec) assume all documents to be loaded in memory, and do not provide functionality to process the data in small batches to reduce the memory footprint.

To speed up training, multiple models were trained in parallel on the available cores where possible (e.g., when training based on the same hyperparameters multiple times to average the obtained metrics). For case study 3, attempts to train multiple models in parallel always resulted in memory-related errors due to each parallel process requiring a complete copy of the complete dataset. Because of this, all models for case study 3 were trained in a sequential fashion during hyperparameter optimization to keep the memory footprint acceptable. Model

²³ For an in-depth technical introduction to Bayesian Optimization, see <https://www.cs.cornell.edu/courses/cs4787/2019sp/notes/lecture16.pdf>

²⁴ <https://scikit-optimize.github.io/stable/modules/generated/skopt.Optimizer.html>



Topic modelling and text classification models for applications within EFSA

Vandevoort B., Bex G. J., Crevecoeur J., Neven F.

training, including hyperparameter optimization, for case studies 1, 2, and 4 were completed in less than 1.5 hours per model, while model training for case study 3 took up to 5 hours.

The custom domain-specific sentence transformer for case study 3 trained on the dataset itself using TSDAE was trained once beforehand. During model training/hyperparameter optimization, this sentence transformer is loaded and embeddings are derived from this transformer analogous to other online available pre-trained transformers. Training the transformer using TSDAE required 25 hours, and retraining this model during hyperparameter optimization is therefore not recommended.



3 Assessment/Results

This section is structured as follows. For each case study, we assess the practical applicability of the different methodologies for text classification and topic modelling presented in Section **Error! Reference source not found.** and propose a number of concrete experiments to quantitatively validate the effectiveness of these methodologies. We then discuss our findings after having performed these experiments in a corresponding results subsection.

For each case study, we will apply data cleaning techniques in line with common practice, such as duplicate removal and removal of stop-words as well as words that appear only once in the corpus.

3.1 Assessment Case Study 1

We can formalize the problem as follows. Let $Q = \{q_1, \dots, q_{10}\}$ be the set of questions. Let $P = \{p_1, \dots, p_4\}$ be the set of pillars. Each question q_i has a set of associated pillars $P_i \subseteq P$ where $i \in \{1, \dots, 10\}$. We can now define the set of subquestions S_{ij} for pillar j in question i . These sets are disjoint, so $S_{ij} \cap S_{kl} = \emptyset$ for $i \neq k$ or $j \neq l$ and $S_{ij} \subset S = \{s_1, \dots, s_{155}\}$. In addition, we consider the set of training documents $D = \{d_1, \dots, d_{755}\}$. Given a document $d \in D$, the task is to determine the sets of relevant questions Q_d , pillars P_d and subquestions S_d for this document.

3.1.1 Hierarchical classification methods

Given this hierarchy of questions, pillars and subquestions and its nature, if the subquestions for a document can be identified, the questions and pillars for that document are also known. Conversely, if a question can be determined for a document, the relevant sets of pillars and subquestions are also known, and the number of possibilities drops drastically. This implies that the classification problem can be tackled using hierarchical text classification. Four paradigms have been proposed for hierarchical classification [41]: flat classification, local classifier per node, local classifier per parent node and global classifier.

Given that once the subquestions have been identified, the pillars and questions are known, one could attempt a flat classification, i.e., predicting simply the subquestions. However, given that there are 155 subquestions for a training corpus of 755 documents, it is questionable whether such an approach would be successful as subquestions can be semantically related, but pertain to different pillars and questions. The local classifier per node method is well suited for binary tree hierarchies, which for this data set would introduce many artificial categories, making the approach impractical.

In theory, it would be possible to train a global classifier using, e.g., Clus-HMC algorithm that infers predictive cluster trees [6]. In this case, a single classifier is trained that predicts all the nodes in the class hierarchy the document is associated with. It is checked whether the result is consistent: for any node, its parent node should be included in the result if it has one. Although the classification task is non-trivial, the constraints imposed by the class hierarchy simplify the task at hand.

Local classifier per parent node is a top-down approach, for each parent class in the hierarchy, a multiclass classifier is trained. In general, the same type of classifier is used for all nodes, but it has been suggested [40] that using different types of classifiers for various nodes may



be beneficial. The latter approach is preferable for this problem due to the imbalance in the data set.

3.1.2 Document encoding

TF-IDF vectors (cf. Section 2.2.2.1) can be used to represent the documents. A drawback of TF-IDF is that it has to be constructed on the corpus that is provided and given its small size, it might be hard to deal with new data, i.e., this might not generalize adequately. However, TF-IDF typically makes for a good baseline.

Since the number of documents is fairly small, using a pre-trained word embedding seems the best option. Several candidates can be considered: word2vec, Fasttext and GloVe (cf. Section 2.2.1). All three methods have been evaluated in the context of text classification [44], with similar results for word2vec and GloVe. It is an option to attempt to fine-tune the pre-trained embeddings based on the available documents. In order to reduce the word vectors to a vector representing the document, the word vectors could be averaged.

A document embedding could also be directly constructed using for example a pre-trained Sentence-BERT model (cf. Section 2.2.2.6) or sup-FastText (cf. Section 3.1.3).

3.1.3 Classifiers

There are a number of options for the specific classifiers to use. A widely used non-parametric model is boosted trees. These are an improvement on decision trees by growing multiple trees sequentially. Each tree can be rather small with only a few terminal nodes (a so-called stump) and is fitted to the residuals of the previous tree [20]. The first prominent boosting methods (e.g. AdaBoost) were based on finding general rules-of-thumb (algorithm driven) to minimize the prediction error [39]. A next iteration of boosting was the gradient-descent based approach, also termed gradient boosting machines (GBM). The principle is that during the training phase, a chosen differentiable loss function (e.g., logarithmic loss) is minimized using gradient descent [31]. Extreme Gradient Boosting (XGBoost) builds upon this idea and introduces further improvements to the formulation such as advanced regularization and computational enhancements [14]. Another candidate is Support Vector Machines (SVM). For very small classes, a rule-based approach might be more appropriate.

In [44] a number of approaches have been compared in the context of hierarchical text classification. XGBoost is reported to perform well and outperforms SVM. sup-FastText, the variant of FastText that trains word embeddings relative to a supervised classification task, is an excellent candidate as well since it outperformed alternatives (including XGBoost and SVM) by an, admittedly, small margin [44]. The experiments described in [44] were performed on a corpus of 800,000 documents in 103 categories, at most 4 levels deep. So, these recommendations should be considered with the necessary caution.

3.1.4 Segmentation and Metadata

As input, the entire text of a document could be used, pre-processed via a word-embedding into a feature vector [44].

Alternatively, it might be useful to segment the document (cf. Section 2.2.3), and use the segments as input for the classifier. Segmentation can be done semantically using the TextTiling algorithm [22]. Alternatively, structural segmentation can also be done based on the layout of the PDF document. The output of the OCR step provides coordinates of word sequences, a straightforward way to recognize paragraphs is to determine them based on the extra space between lines.



In addition to the textual data, the documents have metadata such as authors and their affiliations as well as other relevant information, such as the EFSA unit which contributed to the opinion. Given that researchers or research groups typically concentrate on a limited number of domains, this can be valuable attributes for the classification as well.

3.1.5 Explainability

In addition to obtaining a label for a given document, the underlying motivation why this specific label was assigned by the classifier is often desired as well. For text classifiers in particular, labels can be explained by providing the text passages (e.g. words or sentences) contributing the most to the obtained label. Since classifiers act as a “black-box”, explainability is typically a nontrivial task.

If the documents are segmented, it will be relatively easy to attribute subquestions to document segments, regardless of the algorithms used. Alternatively, or in addition to that, LIME [37] or SHAP [28] could be used to identify the most relevant text passages that contributed to the classification. LIME and SHAP approach explainability in a way that is in large part model-agnostic and it has been used in the context of NLP tasks. Both approaches calculate the contribution of each feature to the prediction.

3.1.6 Proposal

For Case Study 1 we suggest the following approaches:

1. TF-IDF vectors for document representation combined with XGBoost for classification serving as a baseline.
2. Document embeddings based on averaging word embeddings or a pre-trained SBERT model. For hierarchical classification, a local classifier per parent node is constructed based on a combination of XGBoost and rule-based classifiers.
3. Document embeddings derived from sup-FastText combined with rule-based classifiers.

In addition, we would propose to compare the results of using text segments as an alternative to using the entire document as input. For classes with only a few data points, we propose to use a rule-based classifier.

3.2 Results Case Study 1

3.2.1 Models

Five NLP models were trained on the articles provided for case study 1:

1. XGBoost using TF-IDF word vectors
2. XGBoost using TF-IDF word vectors with a hierarchical classification strategy
3. XGBoost using a Doc2Vec embedding
4. XGBoost using a Doc2Vec embedding with a hierarchical classification strategy
5. FastText

The non-hierarchical XGBoost approaches (1 and 3) train a separate Gradient Boosting Machine. FastText (5) trains a single model which immediately predicts all labels. These approaches (1, 3 and 5) do not include the hierarchical structure of the data and as a result might attribute subquestions to a document without including the corresponding questions

and pillars. The hierarchical approaches (2 and 4) do not have this drawback as models are fitted on conditional data, i.e. pillars are modelled conditional on the presence of questions, and subquestions are modelled conditional on the presence of both questions and pillars. However, this feature comes at the cost of having a more complex model with more complicated relationships between the input data and the attributed label. In the hierarchical models the probability of a subquestion being present in a new document is computed by combining the predictions of the question, pillar and subquestion model as follows:

$$P(\text{Subquestion}) = P(\text{Question}) \cdot P(\text{Pillar} | \text{Question}) \cdot P(\text{Subquestion} | \text{Question}, \text{Pillar}).$$

This relationship guarantees that also in the predictions a subquestion can only be attributed when both the question and pillar have already been attributed to the document.

The model specific parameters are:

- XGBoost with TF-IDF:
 - Remove stopwords
 - Apply stemming
 - Retain word fraction 5% -- Top 5% of words with the highest TF-IDF score are retained
 - Max tree depth : 2
 - Eta (learning rate) : 0.2
 - Trees : 100

The hierarchical version is trained with the same hyperparameters

- XGBoost with Doc2Vec:
 - Remove stopwords
 - Apply stemming
 - Doc2Vec vector size : 50
 - Doc2Vec epochs : 40
 - Max tree depth : 2
 - Eta (learning rate) : 0.2
 - Trees : 100

The hierarchical version is trained with the same hyperparameters

- FastText
 - Remove punctuation
 - Remove stopwords
 - Stemming
 - Epochs : 25
 - Learning rate : 0.5

The default values for the preprocessing (e.g. removing stopwords) and the model parameters (e.g. number of trees) listed above were chosen based on the hyperparameter tuning strategy outlined in Section 2.2.5. The number of trees (100) used in XGBoost is relatively low compared to general advice, but higher values resulted in significant overfitting given that there are only 752 documents. In TF-IDF 5% of all (stemmed) words are used to train the classifier, which amounts to 4178 words. A large number of words is required because this

vector is sparse as there are many topics and diversity among documents is large. In Doc2Vec a smaller embedding of only 50 features is chosen as the Doc2Vec vector is dense.

The APRIO labels have a hierarchical structure (Question – Pillar – Subquestion). This hierarchical structure can be preserved in the output at the cost of using more complex NLP models. Alternatively, consistency of predictions with the hierarchical structure can be obtained via simpler non-hierarchical models followed by an additional filtering step. The web application that was built for this case study applies such filtering when necessary.

3.2.2 Evaluation metrics

We distinguish two types of evaluation metrics. The first type of evaluation metrics is probabilistic and is based on the probability of assigning a label to a given document. The second type is classification based and assigns to a document all labels for which the probability exceeds a given threshold. A threshold of 50% was used in this study. The classification based metrics evaluate the similarity between the actual and predicted labels.

3.2.2.1 Probabilistic metrics

These measures use the probability $p_{i,j}$ of assigning label j to document i .

Loglikelihood

For each label the binomial loglikelihood is computed and these loglikelihoods are summed across all labels. This metric is also called binary crossentropy. The loglikelihood is computed as

$$\sum_{\text{document } i} \sum_{\text{label } j} \log(p_{ij}) \cdot \text{label}_{ij} + \log(1 - p_{ij}) \cdot (1 - \text{label}_{ij}),$$

where label_{ij} is one when document i has label j and zero otherwise. This evaluation metric is maximized when training the NLP models. The main disadvantage of the loglikelihood is that it cannot be interpreted directly.

Geometric probability

The geometric probability applies a monotonic transformation to the binomial loglikelihood to facilitate its interpretation. It is calculated as

$$\text{Exp} \left(\frac{1}{\text{num documents} \cdot \text{num labels}} \cdot \text{loglikelihood} \right).$$

This transformation results in a value between 0 and 1, which can be interpreted as the geometric average of the probability assigned to the correct answer. A model performs well when the geometric probability is close to one.

3.2.2.2 Classification metrics



These metrics are computed by first assigning to a document all labels for which the predicted probability p_{ij} exceeds 1. The metrics are based on

- TP : true positives, i.e., the number of documents that are correctly classified as belonging to a class.
- FP: false positives, i.e., the number of documents classified as belonging to the class while they do not.
- TN: true negatives, i.e., the number of documents correctly classified as not belonging to the class.
- FN: false negatives, i.e., the number of documents that was classified as not belonging to the class while they do.

If the test set has N documents, $TP+FP+TN+FN=N$.

Accuracy

The accuracy is the ratio of correct predictions versus the total number of predictions, i.e.,

$$\frac{TP + TN}{TP + TN + FP + FN}$$

True positive rate (recall)

The true positive rate is the probability that an actual member of the class will be classified as such. It is calculated as

$$\frac{TP}{TP + FN}$$

False positive rate

The false positive rate is the probability that a non-member of the class will be classified as a member. It is calculated as

$$\frac{FP}{FP + TN}$$

Positive predictive value (precision)

The positive predictive value is $P(\text{document}_i \text{ has label}_j \mid p_{ij} > 0.5)$, i.e. the probability that the document actually has the label given that it has been attributed. This is calculated as

$$\frac{TP}{TP + FP}$$

Negative predictive value

The negative predictive value is $P(\text{document}_i \text{ has not label}_j \mid p_{ij} < 0.5)$, i.e. the probability that the document doesn't have the label when it is not attributed. This is calculated as

$$\frac{TN}{TN + FN}$$

F1 score



The F1 score combines the precision and recall metric into a single new metric. This metric is often used for measuring the performance of imbalanced classification problems as is the case with the APRIO labels. It is calculated as

$$2 \cdot \frac{\textit{precision} \cdot \textit{recall}}{\textit{precision} + \textit{recall}}$$

3.2.3 Model training

3.2.3.1 Training data

We investigated training the model only on a part of the document (abstract, metadata), but in the end chose to use the full document as training input.

Using only the abstract the training error was twice as high. Although abstracts are informative, they are not sufficiently rich to assign all labels.

The most useful metadata for the documents are the keywords assigned to each document. Using these keywords XGBoost models could reach an accuracy of 60%, which is significantly lower than the 95% and higher performance reached using the full document.

Since both the abstract and the keywords are part of the text, their content is also used by the model when training on the full document text.

The TextTiling algorithm implemented in NLTK has been tested on the text representation of the documents. The algorithm is only able to split on sections of the articles, not taking into account any semantics of the text. As segmentation was intended to help provide feedback to the analyst on the relevance of sentences or paragraphs, using TextTiling will not provide any useful information. We also investigated splitting into paragraphs based on the basis of the spacing between the lines. This approach provides more useful opportunities to provide feedback, but suffers from the presence of tables in the documents as the OCR output has too little information to identify these as such.

Cross validation

All reported metrics are evaluated on out-of-sample data using 5-fold cross validation. The data is split into five equally sized folds. Each of these folds is selected once as the hold-out fold, with the model being trained on the remaining four folds and evaluated on the out-of-sample hold-out fold. The evaluations on the hold-out folds are combined to compute the evaluation metrics. This cross-validation strategy makes optimal use of the data available, which is important as there are only 752 documents.

3.2.4 Evaluation

3.2.4.1 Evaluation across all labels

Table 1 evaluates the previously listed evaluation metrics for each strategy across all questions, pillars and subquestions. All selected models perform roughly equally well on the task. Therefore, we favour the non-hierarchical XGBoost TF-IDF since it is the simplest model providing word importances which can be used to obtain more insight in the model. In



applications where preserving the hierarchical structure is essential, this hierarchical structure can be restored by applying a filtering step on the results from the non-hierarchical model.

	<i>FastText</i>	<i>XGBoost TF-IDF</i>	<i>XGBoost TF-IDF Hierarchical</i>	<i>XGBoost Doc2Vec</i>	<i>XGBoost Doc2Vec Hierarchical</i>
<i>Geometric probability</i>	91.5%	91.2%	91.6%	90.9%	91.2%
<i>Accuracy</i>	96.6%	96.8%	96.8%	96.6%	96.6%
<i>True positive rate</i>	74.4%	75.4%	78.8%	74.5%	77.4%
<i>False positive rate</i>	1.4%	1.3%	1.6%	1.5%	1.8%
<i>Positive predictive value</i>	81.9%	83.6%	81.3%	81.3%	79.3%
<i>Negative predictive value</i>	97.8%	97.9%	98.2%	97.8%	98.0%
<i>F1 score</i>	78.0%	79.3%	80.0%	77.8%	78.3%

Table 1: Model performance on Case study one for a range of models and metrics

During model training the (in-sample) geometric probability is optimized. Hence, using geometric probability for model selection creates a consistency between the training and selection process. Moreover, geometric probability provides the best view on how well the model performs statistically and allows making statistical statements about the performance.

The other metrics are classification metrics and can be optimized to improve the predictive performance of the model. The accuracy and F1-score measure the overall performance of the model. Selecting a model based on accuracy or F1-score is interesting when the model is used as a black box and the predictions (attributed labels) without the probabilities are most important.

For true positive rate (resp. positive predictive value) there is a trade-off with the false positive rate (resp. negative predictive value). One can be increased, with a reduction of the other, by changing the classification threshold. When it is better to detect too many labels rather than miss some, the attribution threshold can be lowered, which results in attributing more labels and hence increase the true positive rate, while simultaneously increasing the number of false positives. Similarly, a lower threshold corresponds to a lower true predictive value and a higher false predictive value.

All tested NLP models performed equally well in predicting the APRIO labels. The models are capable of retrieving 80% of the labels given to each document (true positive rate), and 80% of the labels attributed by the models were correct (positive predictive value). As a result, this task cannot be fully automated and domain experts still have to carefully review the predicted labels.

3.2.4.2 Evaluation by hierarchical level

Table 2 computes the same performance metrics for the XGBoost TF-IDF model per hierarchical level. Performance is lower on subquestions, which are less prevalent in the data and are therefore more difficult to learn.

<i>Level</i>	<i>Question</i>	<i>Pillar</i>	<i>Subquestion</i>
<i>Accuracy</i>	97.7%	96.8%	96.8%
<i>True positive rate</i>	90.5%	87.4%	69.4%
<i>False positive rate</i>	1.1%	1.8%	1.2%
<i>Positive predictive value</i>	93.3%	88.2%	80.3%
<i>Negative predictive value</i>	98.4%	98.1%	98.0%

Table 2: Performance per hierarchical level for the XGBoost TF-IDF model

3.2.4.3 Evaluation by label

As the number of labels is large, we cannot discuss in detail the performance for each label separately. The following two figures provide some insight into the performance at the level of individual labels.

Figure 7 shows for each label the average probability assigned to out-of-sample documents with the given label and without the given label. Overall, probabilities are higher on average for new documents with the given label. This proves that the model has learned useful patterns for classifying new documents. There are also a large number of labels which always predict very low probabilities. This is the case for labels which appear infrequently in the data. These labels are mostly subquestions, but there are also a few rare questions and pillars.

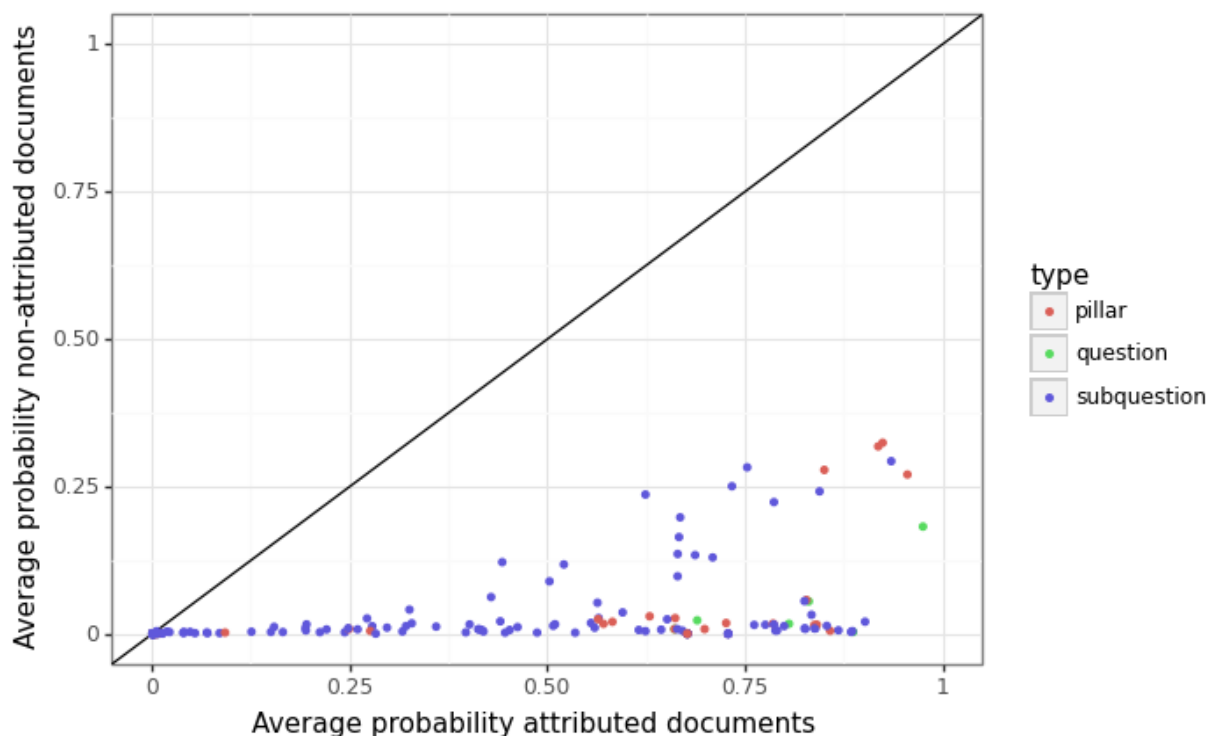


Figure 7: Average probability assigned to out-of-sample documents with the given label and without the given label.

For Figure 8 we compute the true positive rate for each label and plot these true positive rates from low to high. The bottom left of the figure shows a large number of labels with a true positive rate of zero. These are infrequent labels for which the predicted probability out-of-sample never exceeds 50%. As a result these labels are never attributed to new documents. This is a significant portion of the number of labels, but as these are the infrequently used labels, they account for only a small portion of all attributed labels. On the other end, we see that for many labels most of the actual documents are identified when using a threshold of 50%.

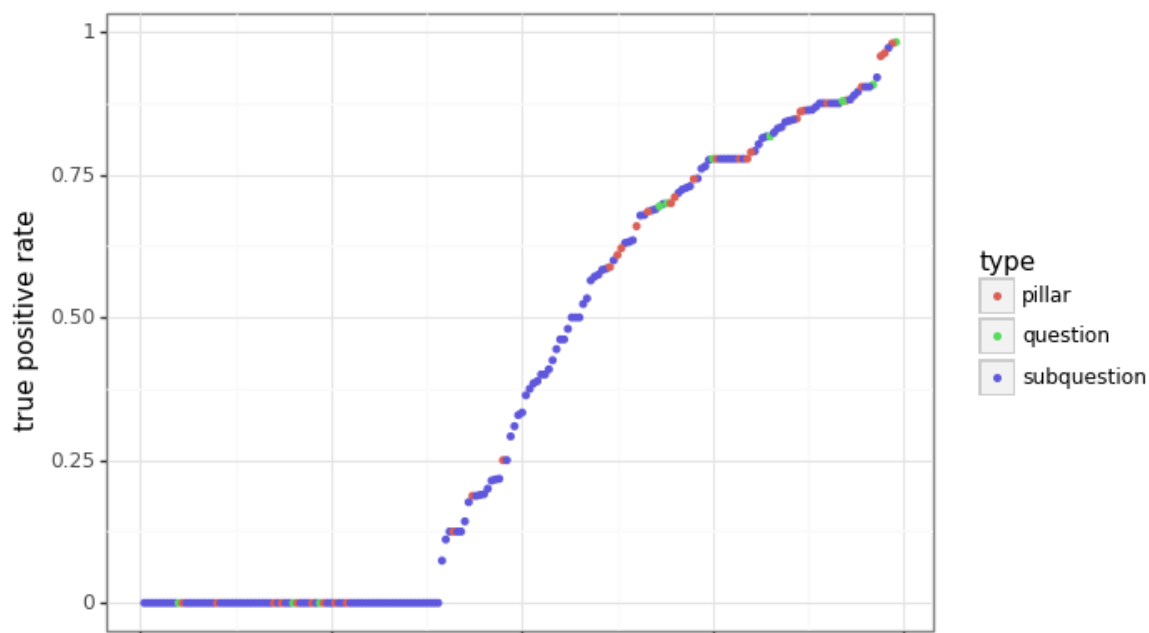


Figure 8: True positive rate per label ordered from low to high.

In Figure 9 we visualize the relation between label frequency on the log scale and the F1-score of the label. When labels are too infrequent, the label will never exceed the attribution threshold of 50% and as a result the F1-score will be zero. We see that in general pillars and questions can be predicted with high accuracy when they appear more frequently. Subquestions can remain difficult to predict even at higher frequencies. This could happen when the label has no distinct vocabulary which can be used to distinguish it from other labels.

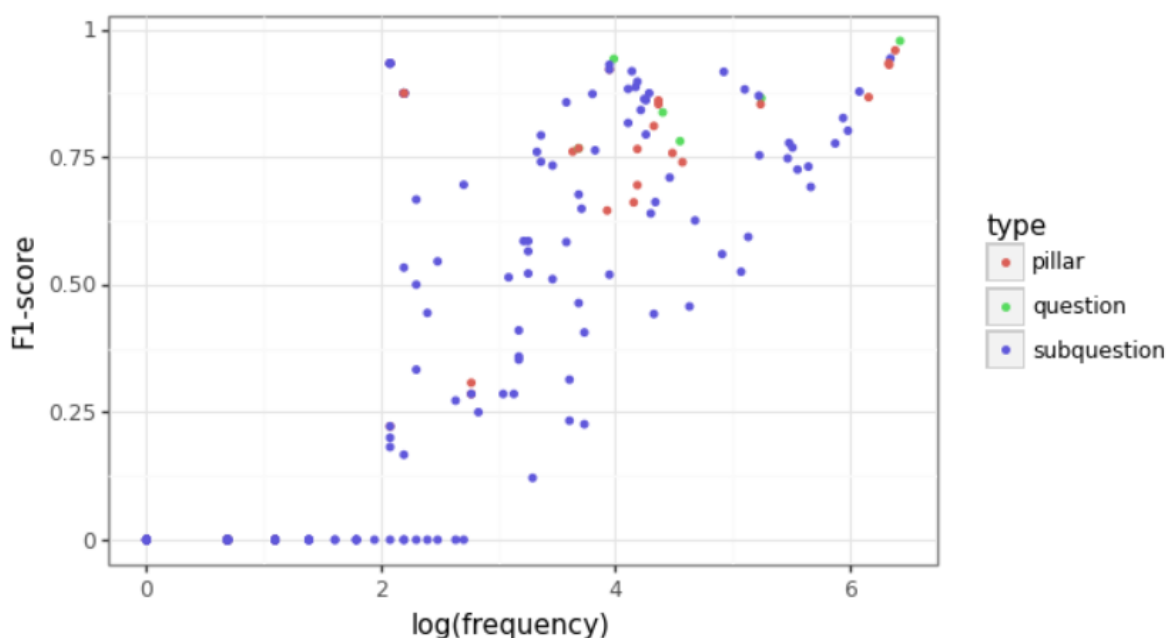


Figure 9: Relation between F1-score and frequency of the label in the dataset on the log-scale.



Performance of the NLP model depends on the question, pillar and subquestion at hand. The models perform very well on labels which are frequently attributed and have a specific vocabulary. The models should not be used to predict infrequent labels.

We continue by looking in detail at three specific labels:

- Question: Human or animal RA.
- Pillar: Exposure assessment, under question "assessment of methods".
- Subquestion: Monitoring occurrence of foodborne outbreaks, under question "Surveillance" and pillar "Exposure assessment"

These labels were chosen to show the different patterns in the data. Table 3 shows basic properties for these labels. The question « Human and animal RA » is very common in the data set with 621/752 documents having this label. The selected pillar and subquestions appear only 3 and 8 times, respectively. Number of significant words indicates the number of words that were used in the XGBoost classifier. All models use only a small fraction of the total number of available words (4178). Human and animal RA uses the most words (87). This is in line with our expectations since this label appears frequently and hence the model has many documents to learn the relevant ones. The 32 significant words for « Assessment of methods » is high given that only 3 documents have this label. The final column shows the probability assigned by the model to an empty document. This probability is 47.2% for « Human and animal RA », which is very close to the selection threshold of 50%. As a result it is likely that some documents will wrongly be classified with this label. Empty documents are given a probability close to zero for the other two (rare) labels, which reduces the chance of accidentally attributing this label. Of course this also implies that there is a significant chance of not detecting all documents with this label.

Label	Type	Occurrences	Number of significant words	Probability empty document
Human and animal RA	Question	621	87	47.2%
Assessment of methods	Pillar	3	32	0.01%
Monitoring occ. of foodborne outbreaks	Subquestion	8	12	0.06%

Table 3: Basic properties of the selected labels that are investigated in more detail for case study 1.

The top panel in Figure 10 shows the variable importance plot for these three labels. For « Human and animal RA » there are a few very important words and then the word importance drops slowly with a long tail of words that are mildly important. For « Exposure assessment » each word is equally (un)important. This is a strong indicator that the model will have little predictive power. « Monitoring occurrence of foodborne outbreaks » has a few very important words, with more than 50% of the importance given to the presence of words stemmed to prevalently. The bottom panel shows the ROC curves for these models with the red dot indicating the selected balance between false positives and true positives when using a



threshold of 50%. These figures are created using the OOS folds in cross validation. For « Human and animal RA » almost all documents are retrieved, whereas the other two labels are rarely attributed. From the figure it might appear as if we can easily reduce the threshold for « Monitoring occurrence of foodborne outbreaks » to find most of the documents. However, to get just half of the documents this threshold has to be reduced to 5% which results in 9 false positives and 4 true positives.

Figure 11 uses Shap values to visualize how a prediction is obtained for an out-of-sample document. In blue effects are shown that reduce the probability of having the given label, while in red parameters are shown that increase the likelihood of the document having the given label. In general, we see that most words have a positive effect, note that a negative effect of `word x == 0` should be interpreted as word x has a positive effect but since it is not in the document, the probability of the document having the label is reduced. Similar to the importance plot of « Human and animal RA » we see in the Shap plot many words working together to attribute the label with a high certainty (99%). For « Exposure assessment » all Shap values are very low (both the positive and negative effects) and the final probability will remain for any document close to zero. For « Monitoring occurrence of foodborne outbreaks » the words "outbreaks" and "prevalence" have a large effect and increase the prediction from less than 15% to more than 90%. This demonstrates that this classifier strongly depends on a small vocabulary. Shap values were chosen here, over LIME, as Shap values can be displayed at the level of probabilities instead of the linear predictor which simplifies the interpretation.

Topic modelling and text classification models for applications within EFSA



Vandevoort B., Bex G. J., Crevecoeur J., Neven F.

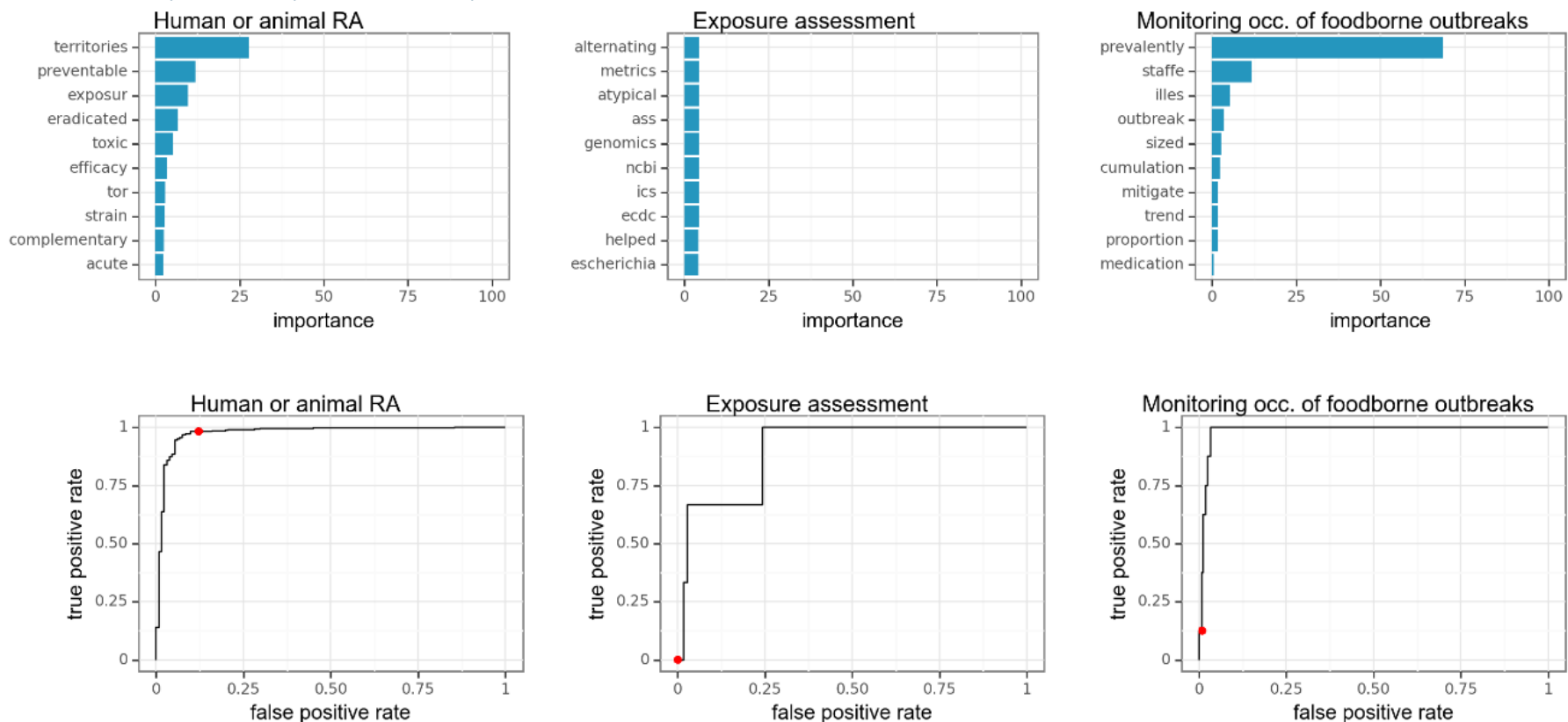


Figure 10: Variable importance and ROC plot for the three selected labels. The red dot on the ROC plot indicates the selection with threshold 50%.



Topic modelling and text classification models for applications within EFSA
 Vandevoort B., Bex G. J., Crevecoeur J., Neven F.

Human or animal RA



Exposure assessment



Monitoring occurrence of foodborne outbreaks



Figure 11: Example of shap values for a new out-of-sample document which has the given label.

For each label only a very small set of words is considered relevant for attribution. Domain experts could review the set of words to validate the model for a given label.



3.2.4.4 Identifying relevant segments

The OCR algorithm used not only returns the content of the document in plain text, but also returns the position (x, y, width, height) of each word in the text. Based on these word positions, we obtain a crude view of the document structure and are able to identify some text segments. Note that this algorithm does not identify all text segments perfectly. The performance is especially lower at tables, images and page transitions.

After training the model on the full document, we investigate whether the model can identify relevant segments by predicting the attributed labels at each segment. Only in 35% of cases where a label was assigned to the entire document, a text segment could be found to which this label would also have been assigned. This shows that labels are usually assigned based on multiple segments and short fragments do not contain sufficient keywords for assigning labels. However, it is still likely that relevant segments receive a higher probability by the model. For this reason the web application highlights for a given label in the PDF the segments that were given the highest probability. The accurateness of this assignation could not be tested in this project as the relevant segments are not marked by the expert in the training data.

3.2.5 Conclusion

NLP models can be used to reliably attribute frequently used labels with a distinct vocabulary. Less frequent labels can only reliably be attributed by domain experts. As more manually labeled documents become available, the set of labels for which NLP algorithms can be used will increase. A proof-of-concept web app that supports exploration of assigned labels has been developed as part of this project.

3.3 Assessment Case Study 2

Given the large fraction of nearly-copy-pasted comments, we propose to perform a more extensive data cleaning step before the actual topic modelling. Comments with a Levenshtein distance less than 20 are grouped. For each group, a list of submitters is maintained, as well as a single representative comment, randomly chosen from this group. Since comments within each group are almost identical, we do not expect this choice to influence the final result. A Levenshtein distance of 20 is chosen as we found that it strikes a threshold value for detecting nearly-copy-pasted comments because it gives a good balance between correctly identifying nearly-copy pasted comment pairs, without falsely identifying pairs of short comments as nearly-copy pasted. Note that Levenshtein distance is a syntactical metric, not taking into account the semantical meaning of comments. Therefore, two comments that are semantically different because of a minor difference (e.g., starting a comment with "Why ..." instead of "How ...") are incorrectly identified as identical by this approach. As part of this project, a web application was built to allow exploration of groups of nearly-copy-pasted comments. In particular, this web application highlights differences between nearly-copy-pasted comments, thereby facilitating manual validation. For the provided dataset, manual validation did not reveal any such false positives when using 20 as a threshold value. Note that grouping based on Levenshtein distance is only intended as an initial data cleaning step by grouping nearly-copy-pasted comments, instead of a replacement for semantical clustering. Choosing a moderate edit distance combined with manual inspection to validate that no false positives are present (facilitated by the provided web application) is therefore preferred over opting for higher values to increase recall at the cost of precision.



comment	submitter	representative	submitters
We find that ...	A	We find that ...	A, B
We finds that ...	B		
On page ...	C	On page ...	C

Figure 12: Schematic overview of the pre-processing step for Case Study 2, grouping nearly-copy-pasted comments and representing each such group with a randomly chosen comment from this group.

The actual topic modelling is performed over the group representatives. Comments are assumed to discuss a number of latent issues, and different comments can cover the same issue. The desired outcome of the resulting topic model is to align with these latent issues as much as possible. That is, each topic in the resulting topic model corresponds with an issue, and all comments covering this issue are assigned to this topic.

One particular challenge for this case study is detecting the different issues within a comment. For the purpose of topic modelling, we identify two types of comments: semantically separated and interwoven comments. In comments of the former type, different issues are addressed one by one (e.g., based on an enumeration), whereas comments of the latter type address multiple issues at the same time. For semantically separated comments, text segmentation can be applied to partition the comment in multiple segments where each such segment addresses a single issue. For interwoven comments, such an approach is not expected to work.

Based on these observations, we propose three different strategies:

1. The first strategy applies a multi-topic topic model where each comment can be assigned to an arbitrary number of topics. The expected advantage of this approach is that both semantically separated and interwoven comments should end up in the corresponding topics.
2. The second strategy leverages text segmentation to partition comments in segments covering a single issue. Afterwards, each segment is assigned to a single topic. The expected advantage of this approach is that topic assignment of semantically separated comments is improved, since each segment covers only one issue. The disadvantage of this approach is that interwoven comments cannot be segmented properly, and are therefore expected to end up in only one topic.
3. The last strategy combines both previous strategies into a hybrid strategy: comments are segmented and combined with the original documents in a new dataset. This dataset is clustered using a single-topic approach first. For each comment, we use the assigned topic (based on the whole comment, not the segments) unless the topic-comment probability is below a specified threshold (i.e., the model is not sufficiently confident about the assigned topic for this comment, potentially indicating that the document belongs to multiple clusters). In the latter case, the topics assigned to the different segments of this comment are assigned to the comment instead.

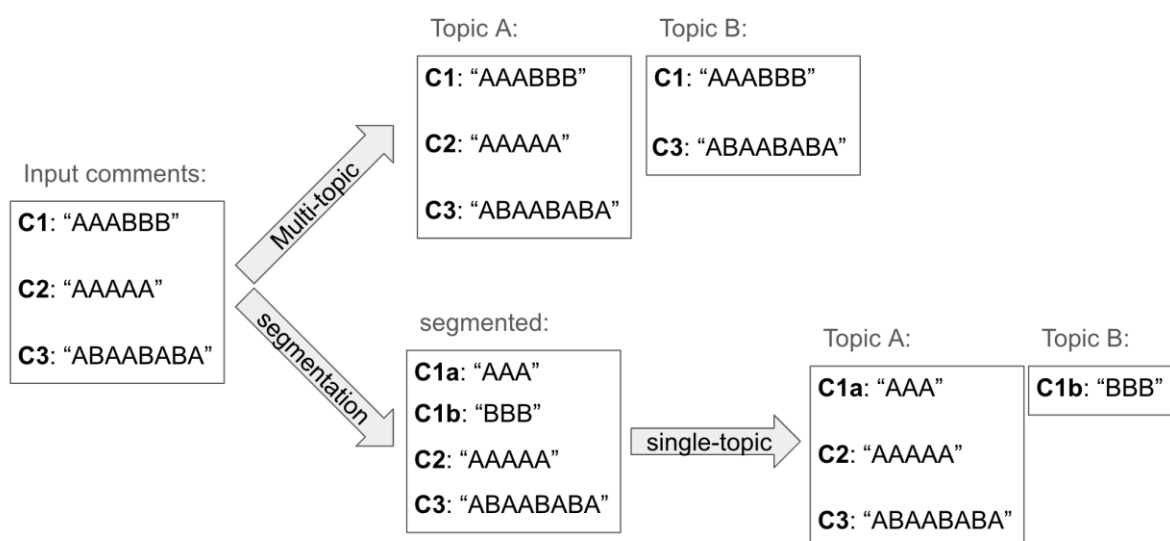


Figure 13: Visualization of the first two topic modelling strategies proposed for Case Study 2, given three comments C1, C2 and C3 over issues A and B. Note that C1 and C3 both discuss issues A and B, but comment C1 is a semantically separated comments whereas C3 is an interwoven comment. The third strategy (not visualized) combines these two strategies by applying single-topic clustering on both the original comments and segments first, followed by a segment-based multi-topic clustering for outliers (i.e., interwoven comments).

For each strategy, both different variations of LDA (LDA, neural LDA, prodLDA, CTM) and Top2Vec are applied as topic models. The best performing variation of LDA according to the ground truth will be used as a baseline for Case Studies 3 and 4, where no ground truth is available. Given the small number of comments, successfully fine-tuning a document embedding on this corpus is highly unlikely. We therefore opt for pre-trained SBERT models. For text segmentation, we will analyse both TextTiling, as well as a structural approach based on segmenting comments into sentences (in the linguistic sense). TextTiling is expected to detect semantic boundaries in the comment, whereas a sentence-based segmentation will most likely lead to over-segmentation (i.e., one issue covering multiple sentences will end up in multiple segments). However, over-segmentation is not necessarily problematic, as we can expect different sentences over the same issue to still end up in the same topic.

Since a ground truth is available and since the focus of the case study is on grouping related comments into topics rather than finding good topic representations, cluster purity will be used as the main performance metric. Other metrics will be reported, but are considered less relevant.

3.4 Results Case Study 2

3.4.1 Ground Truth Annotation

As discussed in Section 2.1.2, a separate text document is provided containing replies to the submitted comments. These replies can have multiple smaller reply points. Furthermore, replies and/or reply points can refer to earlier replies and/or reply points. Figure 14 illustrates this structure over an abstract example of 4 comments and replies with a small number of reply points each.

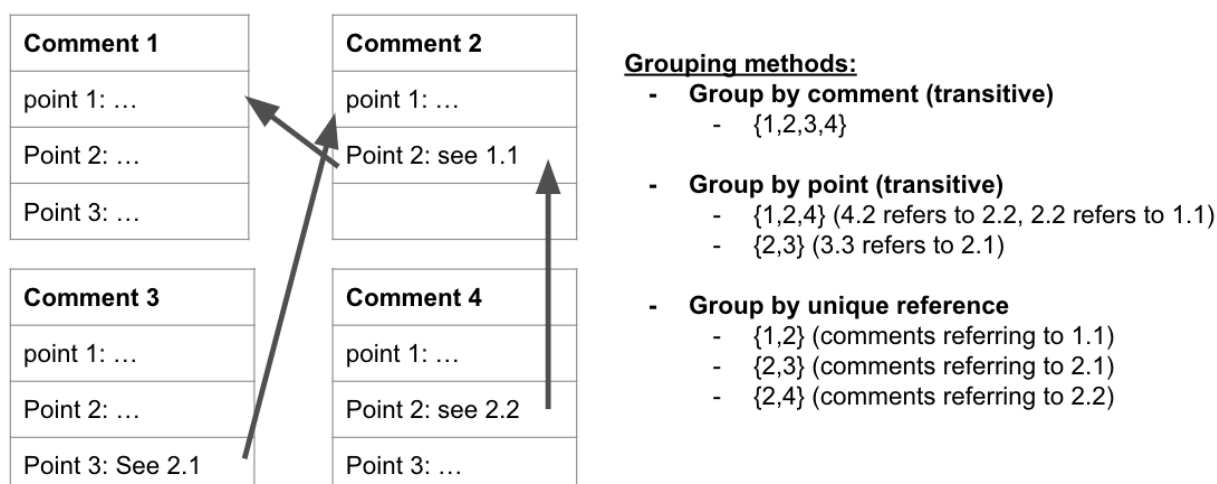


Figure 14: (left) Abstract example of comments with replies and reply points, including references to earlier reply points (indicated by arrows). (right) Resulting comment clusters for all three considered methods.

Grouping Methods

The references between replies and reply points allow for the extraction of a ground truth clustering of comments into groups. We identified three different methods to extract such a ground truth based on the considered granularity of replies (replies as a whole vs individual reply points) and whether or not transitivity of references is taken into account. Under transitivity, we say that a reply (indirectly) refers to another reply if we can find a chain of replies where each reply refers to the next one in the chain. For example, the reply for comment 4 in Figure 14 does not refer to the reply for comment 1 directly, but under transitivity we can conclude that the reply for comment 4 (indirectly) refers to the reply for comment 1, since it refers to the reply for comment 2 which in turn refers to the reply for comment 1. The three identified methods are as follows:

- **Group by comment:** Replies are considered on the granularity of replies as a whole, and transitivity is taken into account. That is, comments are grouped in the same cluster if their replies directly or indirectly reference each other. In the abstract example given in Figure 14, the replies of comments 3 and 4 both refer to comment 2, which in turn refers to comment 1. Because of this, all four comments end up in the same cluster.
- **Group by reply point:** Replies are considered on the granularity of individual reply points, and transitivity is taken into account. In the abstract example given in Figure 14, We create a cluster consisting of comments 1,2 and 4 since reply point 4.2 refers to reply point 2.2, which in turn refers to point 1.1. Reply point 3.3 on the other hand refers to point 2.1, thereby implying a group consisting of comments 2 and 3.
- **Group by reference:** Replies are considered on the granularity of individual reply points, without transitivity. Under this approach, we construct a group for each unique reply point that is referenced by other reply points. In the abstract example given in Figure 14, three unique replies are referenced to: 1.1, 2.1 and 2.2, resulting in three groups where each group contains the comment corresponding to this reference, as well as all other comments for which the reply references this reply point.

For all three methods, each remaining isolated comment (i.e., with a reply not referencing other replies, nor being referenced by other replies) is “grouped” in its own group consisting of only this isolated comment.

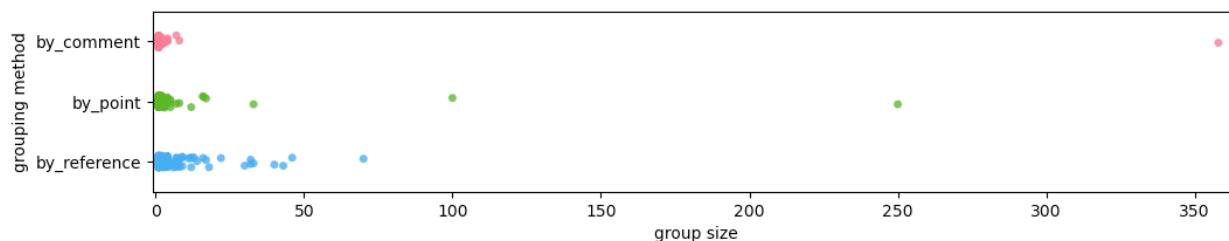


Figure 15: Distribution of group sizes (in number of comments) for each grouping method.

When investigating the resulting clusters for each method, we see that the first two methods group the vast majority of comments in one or two big clusters covering the vast majority of comments (cf. Figure 15), indicating a tight coupling between comments in the dataset. Only the third method, which does not take transitivity of references into account, distributes the comments more evenly across multiple clusters.

The majority of comments on the given opinion are highly correlated. Close relatedness between most of the comments provides an additional challenge for topic models, as the boundaries between topics will be less pronounced.

Based on the previous analysis, we will use the output of *the third method as the ground truth* during the evaluation of different topic models. An additional motivation for choosing this method is that ignoring transitivity seems to be more in line with the underlying relatedness of replies. Consider for example a reply *B* referring to an earlier reply *A*. If a reply *C* refers to *B* instead of *A*, then it is reasonable to assume that *B* not only refers to *A*, but adds to it as well, and this additional information is applicable to *C* as well. In other words, *C* is more related to *B* than to *A*, implying that a separate cluster consisting of *B* and *C* is indeed a reasonable choice.

Data annotation

In order to annotate each comment in the original input data with a number of clusters serving as the ground truth, each comment in the input data must be matched against the comments in the text document. Here, it is important to note that these comments have been slightly modified while constructing the text document (e.g. to fix spelling mistakes or while merging multiple nearly identical comments...). Because of these changes, an exact textual match between comments in the input data and comments in the text document should not be expected.

To solve this issue, all comments in the input data are matched with the most similar comment in the text document. More formally, for each comment in the input data a corresponding comment in the textual document is chosen such that the edit distance between them is minimized. To validate this process, all pairs of matched comments together with the edit



distance between them have been written to a csv file, allowing for a fast manual validation. During this manual validation, no erroneous matches were found.

3.4.2 Models

For each of the three strategies presented in Section 3.3, five classes of NLP models are trained:

- LDA
- ProdLDA
- Neural LDA
- CTM
- Top2Vec

Before training, the comments are pre-processed based on the following parameters:

- **Punctuation and stopwords:** Basic text preprocessing is performed on each comment, involving removal of punctuation and stopwords.
- **Edit distance:** Nearly identical comments with an edit distance below 20 are grouped and included in the model as a single document, as manual inspection reveals that this value strikes a good balance between grouping analogous comments without introducing false positives.
- **Frequency filter low:** Denote this percentage L. Words which appear in less than L percent of all comments are filtered out. Due to the small number of comments which are relatively short in general, we choose a lower value of 0.5% to avoid throwing away too much information. For the given input data, this corresponds to removing all words occurring in only one or two comments.
- **Frequency filter high:** Denote this percentage H. Words which appear in more than H percent of all comments are filtered out. We pick 80% as a threshold, thereby filtering out all words that appear in the vast majority of documents. Since these words appear in almost all comments, it is not expected that they will contribute to the clustering. Further experiments for CS4 (cf. Section 3.8.3) indeed show that changing this parameter has barely any impact on the quality of the resulting models.
- **Stemming:** Indicator whether words are stemmed before applying the model. We follow the recommendations for each model and apply stemming for all models except Top2Vec.

The tuning parameters specific for each model are:

- **LDA:**
 - Num topics – tune values between 10 and 250
 - Alpha – symmetric, asymmetric or auto
 - Eta – symmetric or auto
 - Decay – tune values between 55% and 95%
- **Neural LDA:**
 - Num topics – tune values between 10 and 250
 - Activation – softplus or RELU
 - Solver – adam or sgd

- momentum – tune values between 0.98 and 0.995
- Num layers – tune values between 2 and 4
- Num neurons – tune values between 80 and 150
- Num samples – tune values between 5 and 15
- **Prod LDA:**
 - Same hyperparameters as Neural LDA
- **Top2Vec:**
 - Embedding – doc2vec, universal-sentence-encoder or all-MiniLM-L6-v2
 - Min count – tune values between 1 and 20
 - HDBSCAN min cluster size – tune values between 2 and 20

Finally, each strategy has a small number of parameters that can be tuned:

- **Strategy 1: Multitopic**
 - Outlier threshold – the probability threshold indicating whether or not a comment belongs to a cluster. We tune for values between 0.1 and 0.5.
- **Strategy 2: Segmentation**
 - Segmentation algorithm – sentence or TextTiling
- **Strategy 3: Hybrid**
 - Segmentation algorithm – sentence or TextTiling
 - Original comment threshold – if the probability for a comment is above this threshold, we consider the cluster assigned to the comment, otherwise the clusters assigned to the individual segments are used. We tune for values between 0.0 and 1.0.

3.4.3 Evaluation Metrics

We consider the following four evaluation metrics, discussed in Section 2.2.4.4:

- Topic coherence
- Topic diversity
- Topic information gain
- Multitopic cluster purity

Cluster purity is based on the ground truth discussed in Section 3.4.1.

3.4.4 Optimizing for Multitopic Cluster Purity

Since a ground truth is available, we use this metric as the optimization objective during model tuning. During tuning, a large parameter space is explored and multiple different hyperparameter configurations are tested. Each such configuration results in a trained model for which we can evaluate the different metrics. We emphasize that the other considered metrics are evaluated on each model as well during tuning, allowing us to compare different metrics.

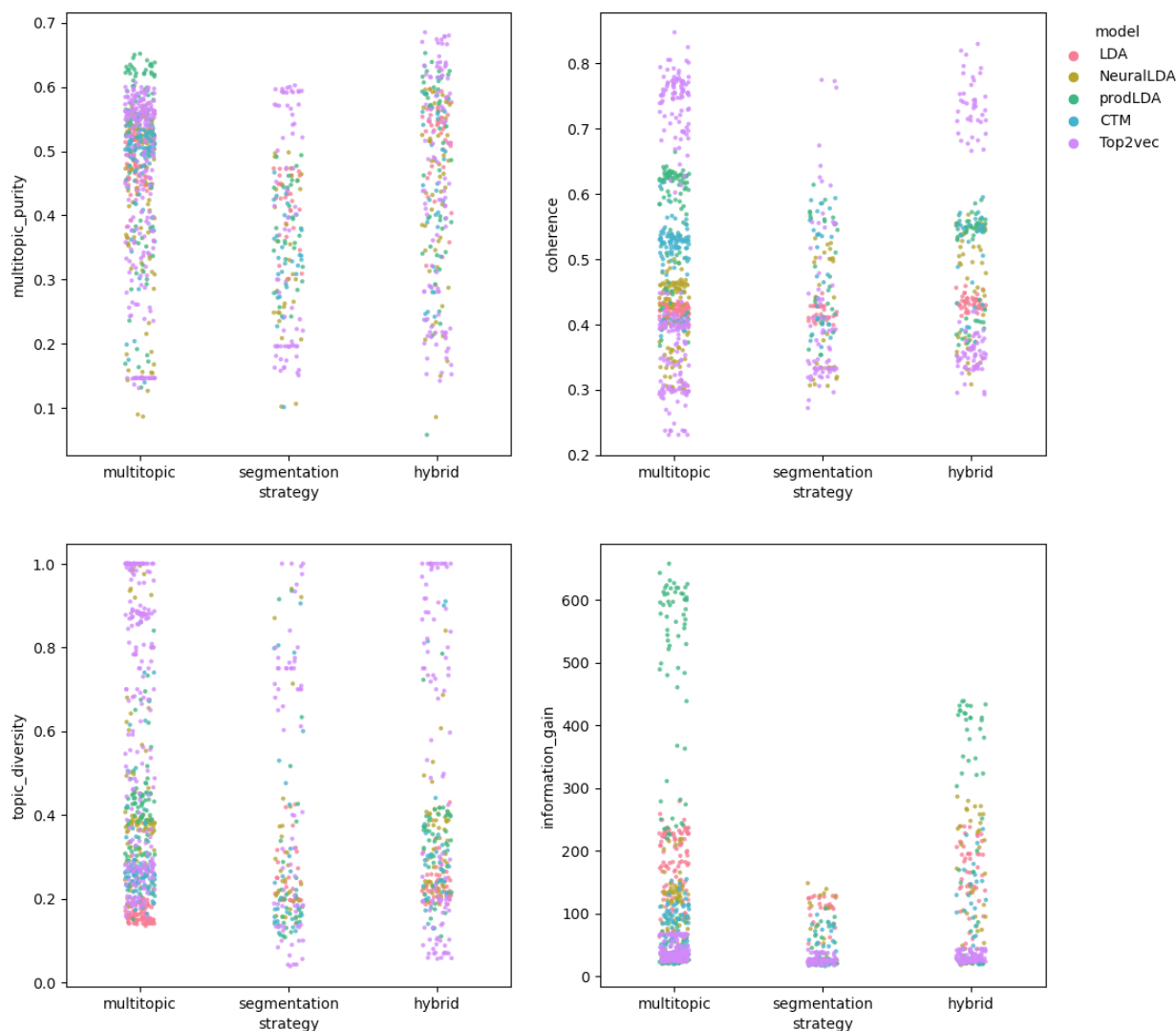


Figure 16: Overview of the different evaluation metrics for all trained models. Each dot represents a model trained based on a specific hyperparameter configuration during the tuning process.

The obtained results are summarized in Figure 16. In these scatter plots, each dot represents the performance of a specific hyperparameter configuration during the tuning process. Since the purpose of hyperparameter tuning is finding the configuration leading to an optimal model, only the dot with the highest score is relevant for each metric when comparing models and strategies. Visualizing the metrics of all tested configurations for each model instead of only the best performing one furthermore gives us additional insights in the tuning process itself. In particular, a single dense region for a specific model indicates that different hyperparameter configurations have only a minor effect on the resulting metric, indicating that extensive hyperparameter tuning is less relevant when trying to improve the model. For the multitopic strategy, this is for example the case for LDA when coherence is used as a metric. If instead the obtained scores for a specific model are more spread out, then the chosen hyperparameters have a more significant impact on the performance of the model. These models can therefore benefit more from hyperparameter tuning. When comparing the



different models, we see that the scores for Top2vec models are usually more spread out than those for the other models (except when using topic information gain as a metric).

For multitopic cluster purity, prodLDA and Top2Vec appear to be the best performing models for all three strategies, where prodLDA outperforms Top2Vec for the first strategy, but Top2Vec outperforms prodLDA on the other two strategies. For topic coherence and diversity, Top2Vec usually outperforms the other models. When looking at the topic information gain metric, prodLDA greatly outperforms the other models, except for strategy 2, where all models seem to perform significantly worse compared to the other two strategies. Recall that topic coherence and diversity only look at the topic descriptions (i.e., the list of words derived by the model for each topic to describe this topic), and not at the concrete documents assigned to each topic, whereas cluster purity on the other hand only looks at the comments assigned to each topic, without taking topic descriptions into account. The topic information gain metric is based on both topic description as well as the assignment of comments to topics. Since for this case study the grouping of comments in coherent groups is more important than the description for each such group (all comments have to be processed anyway, so a summarization is less important), cluster purity and topic information gain are more relevant. For these two metrics, prodLDA is the best performing model. In particular, the Top2Vec models are scoring significantly worse on topic information gain even though they give the best results for topic coherence and diversity. This indicates that these models create good descriptions for each topic (i.e., the group of words for each topic is coherent and has little to no overlap with descriptions of other topics), but comments assigned to these clusters are less in line with the topic description, thereby indicating less qualitative comment groups.

The observation that a baseline model (i.e., prodLDA) is on par with or outperforms Top2Vec should be contrasted with our findings for Case Study 3 (cf. Section 3.6) and Case Study 4 (cf. Section 3.8), where Top2Vec models generally provide a performance improvement over the baseline LDA models. This is most likely due to the different nature of the dataset at hand: whereas Case Study 3 and Case Study 4 apply topic modelling over a dataset consisting of a large number of documents, covering a broader range of topics, the dataset for this case study consists of a smaller number of comments, all addressing the same opinion. It is therefore expected that all these comments will be highly correlated and use a largely overlapping vocabulary. In fact, the ground truth analysis in Section 3.4.1 already indicated such a high correlation between comments.

Due to the small number of comments and high correlation between comments, baseline models (ProdLDA in particular) are on par with or outperform the more complex Top2Vec models when considering evaluation metrics relevant for this case study.

Focussing on the different strategies, we conclude that for most evaluation metrics, the second strategy based on segmentation never outperforms the other two strategies, and is even greatly outperformed by the other two strategies when considering topic information gain as evaluation metric. The other two strategies usually result in similar scores, except for topic information gain, where the best performing prodLDA models for the first strategy clearly outperform those for the third strategy. For the strategies involving text segmentation, only sentence-based segmentation provided meaningful results, as TextTiling could not identify segments due to the comments being too short.



Topic modelling and text classification models for applications within EFSA

Vandevoort B., Bex G. J., Crevecœur J., Neven F.

We next look at the relationships between different evaluation metrics. Since for new datasets no ground truth will be available, identifying a good alternative metric to optimize for is crucial for practical applicability. Figure 17 to Figure 21 visualize for each of the five considered models all pairwise relationships between the four considered evaluation metrics. For topic coherence and diversity, there does not always seem to be a relation with cluster purity. In particular, the Top2vec models having good coherence and diversity scores can still perform bad on multitopic cluster purity, thereby validating our earlier observation that Top2Vec models with good topic descriptions are not necessarily creating qualitative groups of comments. Topic information gain seems to be a good choice for an alternative metric, since models with a high topic information gain always have a high cluster purity score as well. The opposite is not always true: models with a low topic information gain can still have a high cluster purity.

For new datasets without a ground truth, topic information gain is a good choice for model evaluation during optimization.

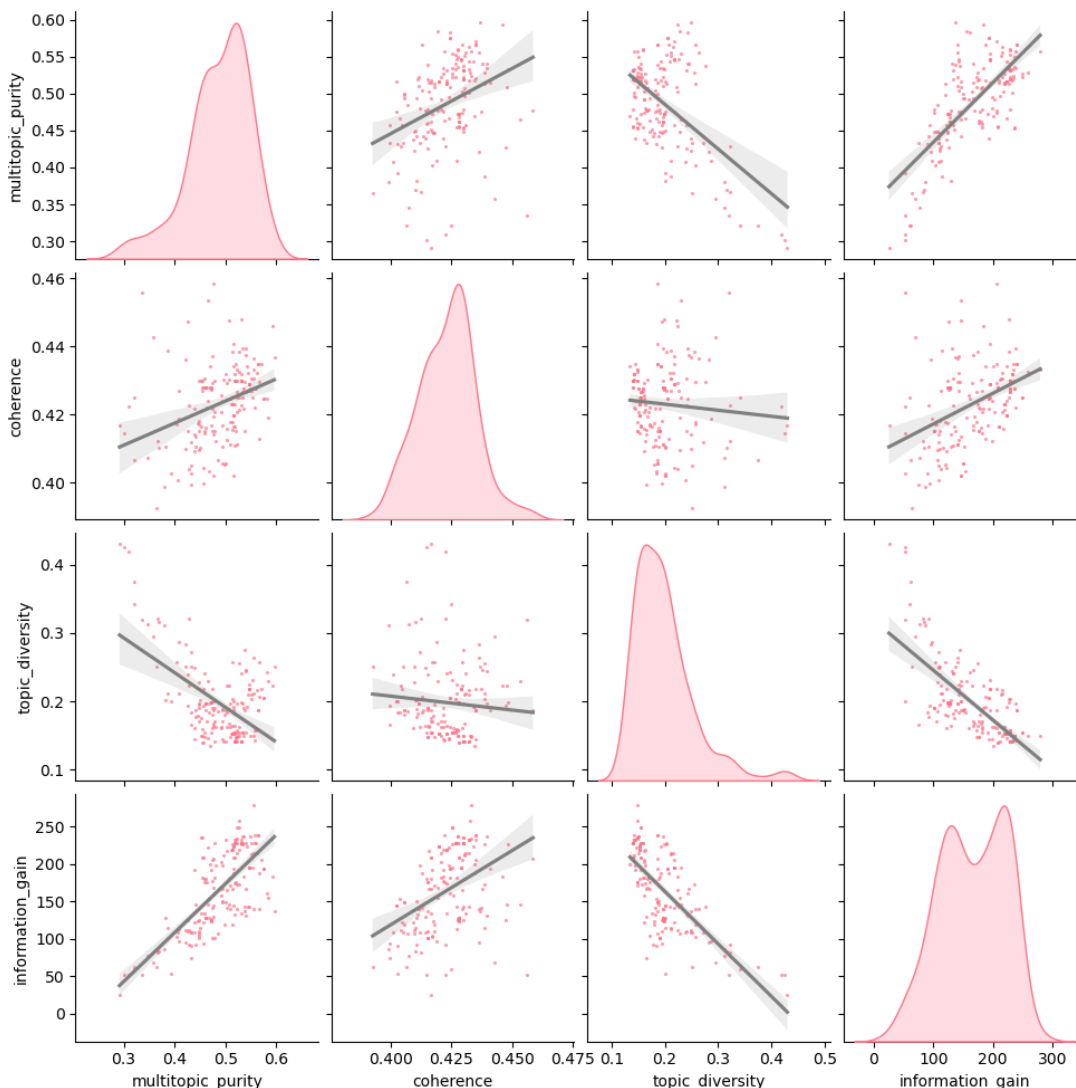


Figure 17: Overview of the relationships between different evaluation metrics for LDA models. For each pair, a linear regression model is fitted with the gray area visualizing the 95% confidence interval.

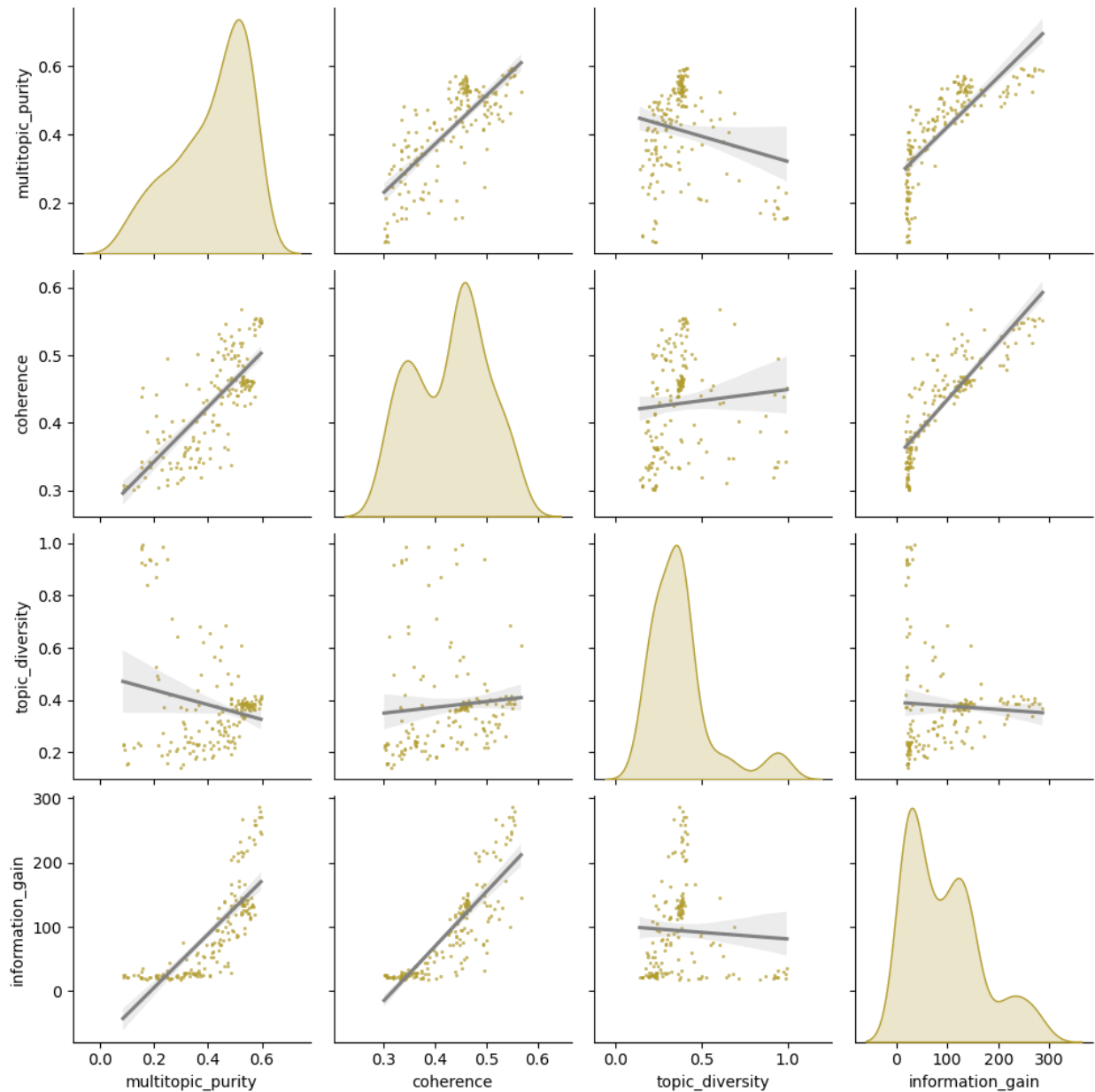


Figure 18: Overview of the relationships between different evaluation metrics for NeuralLDA models. For each pair, a linear regression model is fitted with the gray area visualizing the 95% confidence interval.

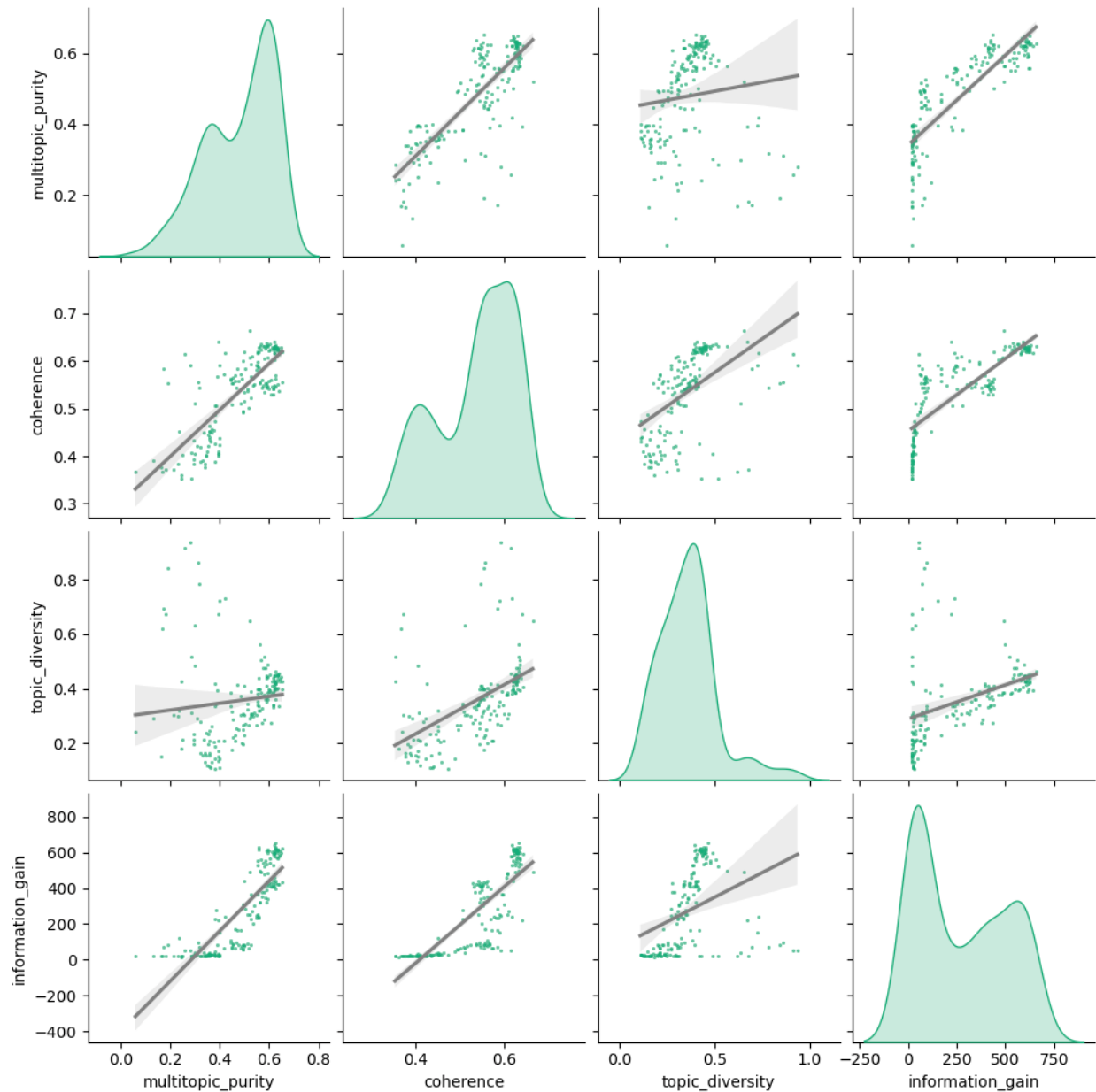


Figure 19: Overview of the relationships between different evaluation metrics for prodLDA models. For each pair, a linear regression model is fitted with the gray area visualizing the 95% confidence interval.

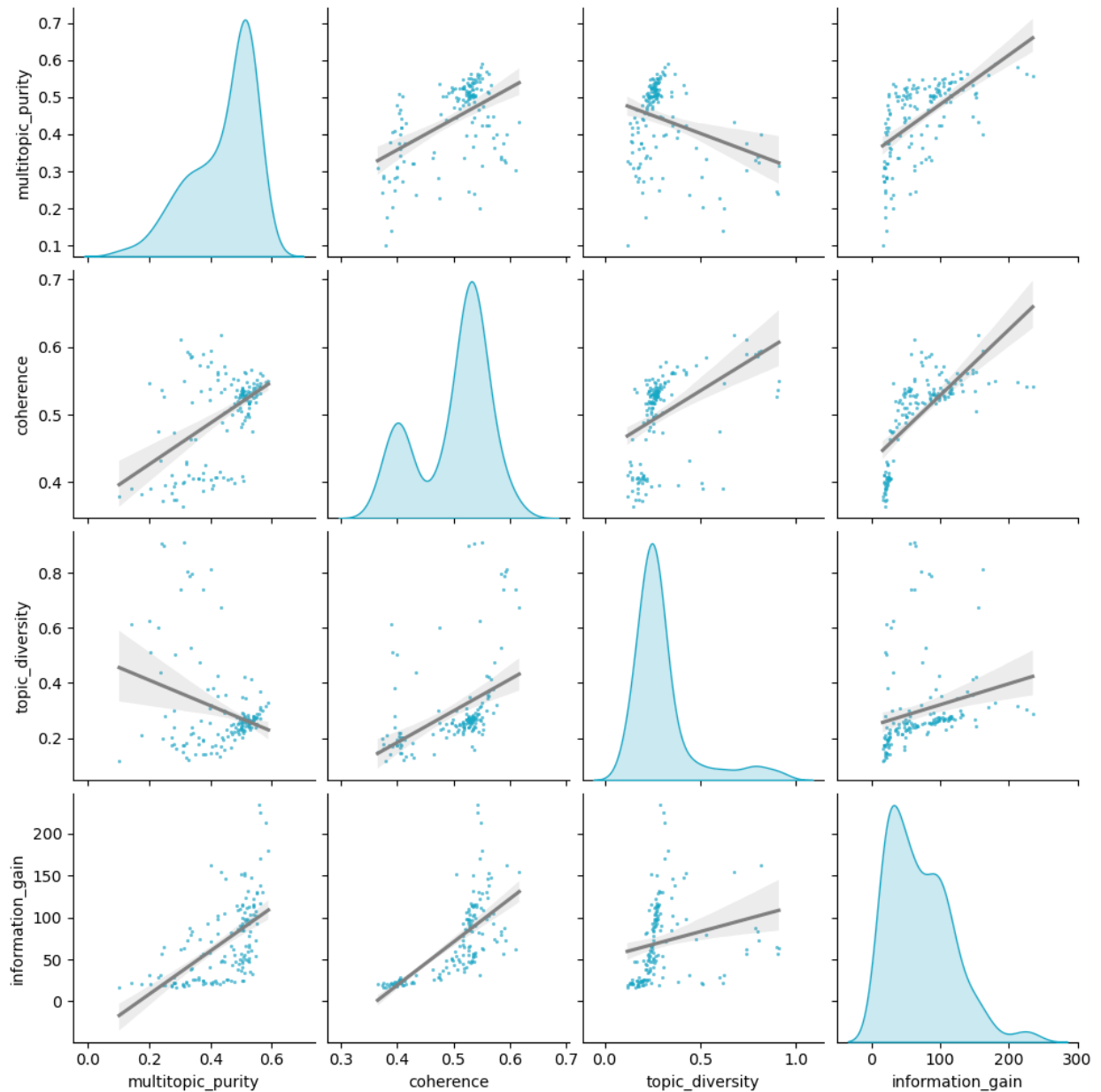


Figure 20: Overview of the relationships between different evaluation metrics for CTM models. For each pair, a linear regression model is fitted with the gray area visualizing the 95% confidence interval.

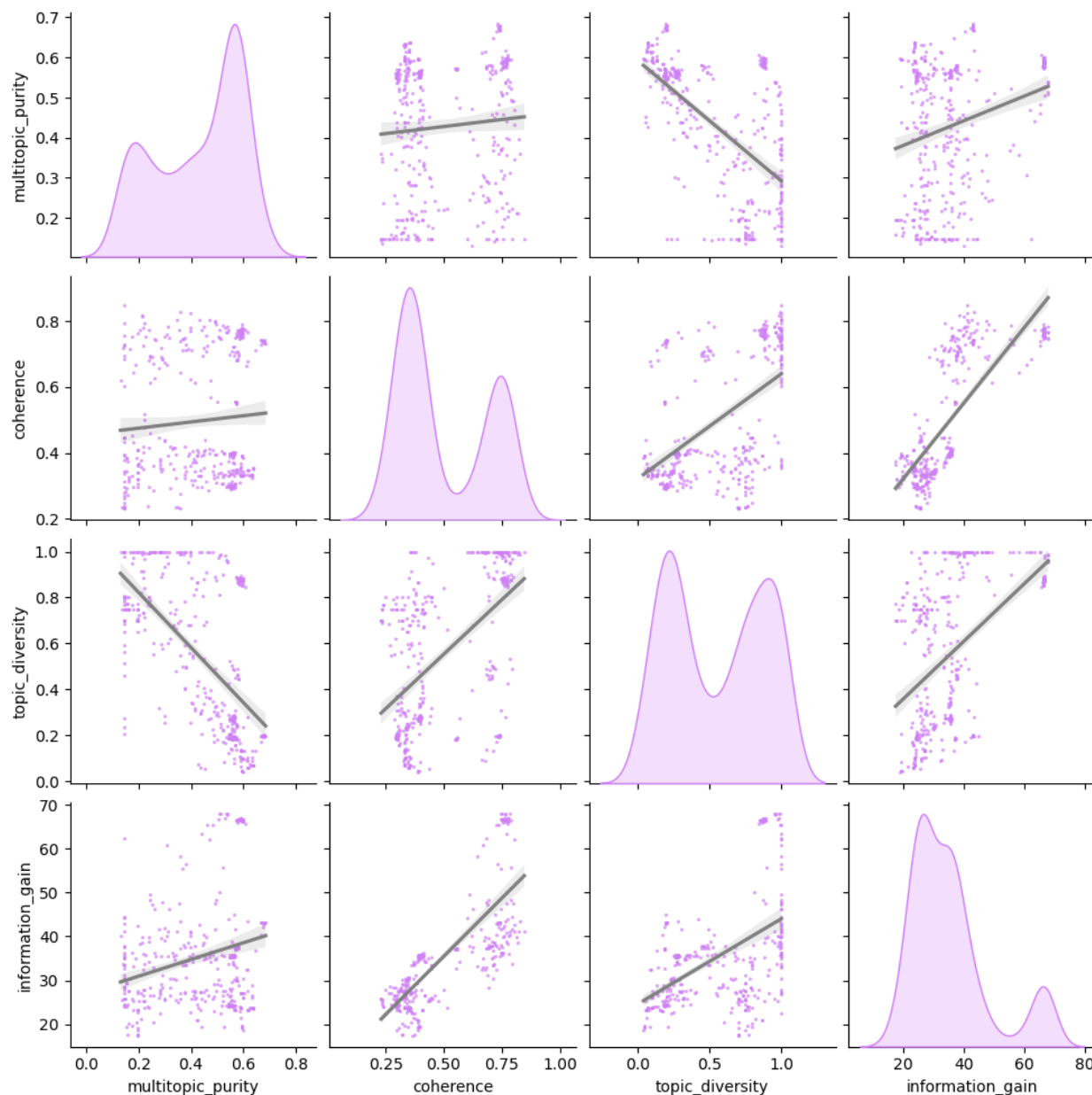


Figure 21: Overview of the relationships between different evaluation metrics for Top2vec models. For each pair, a linear regression model is fitted with the gray area visualizing the 95% confidence interval.

Since prodLDA is the best performing model, we conclude this section by zooming in on this model and identify the hyperparameters with most impact on the resulting model performance. The results of this analysis for each strategy are visualized in Figure 22. As the performance of each combination depends on all parameters, we fit a multivariate regression model with outcome multitopic_purity and covariates the hyperparameters to estimate the individual contribution of each parameter. The results of this model are summarized in Table 3.

We conclude that the chosen solver, number of topics and number of layers have most impact. In particular, using “adam” instead of “sgd” as the solver improves multitopic purity by 11% on average. It is interesting to note that making the model more flexible by adding additional layers reduced performance, whereas increasing the flexibility by increasing the number of neurons has a positive effect. This shows that more complex models do not always perform better. In particular, too complex models are more likely to overfit the model, which reduces the performance on the validation/test data set (bias-variance trade-off). Finding this balance between bias and variance is one of the main reasons for hyperparameter tuning.

Note that these hyperparameters are only optimized for the provided dataset, and therefore are not necessarily optimal for other datasets. Transferability of these hyperparameter configurations to other datasets is unclear. In particular, the hyperparameter indicating the number of topics should reflect the actual number of topics in the datasets, and should therefore be changed for datasets where a different number of topics is expected. Because of this, hyperparameter tuning is always recommended for new datasets.

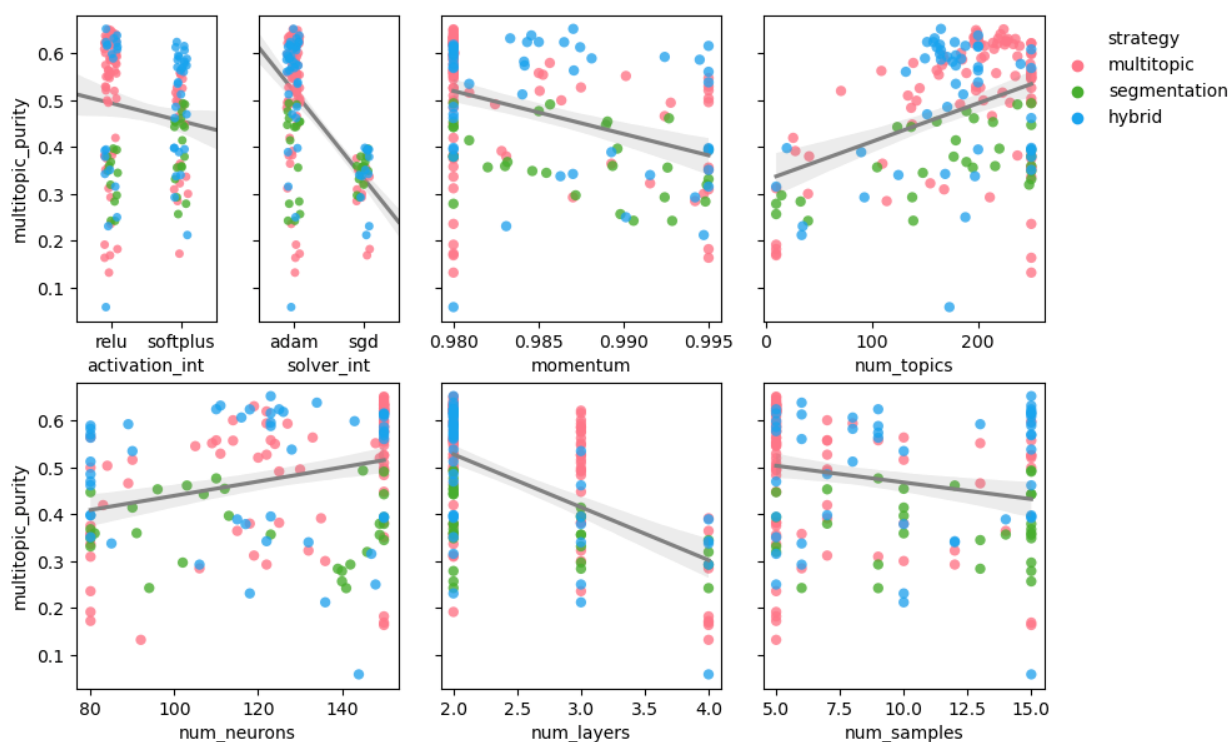


Figure 22: Impact of the tuned prodLDA hyperparameters on model performance for each considered strategy. For each hyperparameter, a linear regression model is fitted to better quantify the impact on the model performance, With the gray area visualizing the 95% confidence interval.

Coefficient	Estimate	P-value
Activation		
Relu (reference)	0 (ref)	
Softplus	0.0035	0.8034

Solver

Adam (reference)	0 (ref)	
SGD	-0.1102	3.22e-10 ***
momentum	-2.1004	0.0747 *
Num_topics		
Less than 100	0 (ref)	
More than 100	0.1369	8.09e-11***
Num_layers	-0.07667	9.49e-13 ***
Num_neurons	0.001056	4.14e-5***
Num_samples	-0.002271	0.1537

Table 3: multivariate regression model of multitopic purity as a function of the tuning parameters.

Since these prodLDA models have a stochastic component, repeated runs with the same hyperparameter configuration are expected to have slightly different topics and scores. To assess the impact of this randomness on the obtained result, we repeatedly trained a prodLDA model with the same hyperparameter configuration 25 times, using the hyperparameters identified as optimal during tuning for strategy 3. The obtained metrics are summarized in **Figure 23**. Due to these differences in metrics between different runs of the same hyperparameter configuration, it is recommended to run the same hyperparameter configuration multiple times to improve the obtained model.

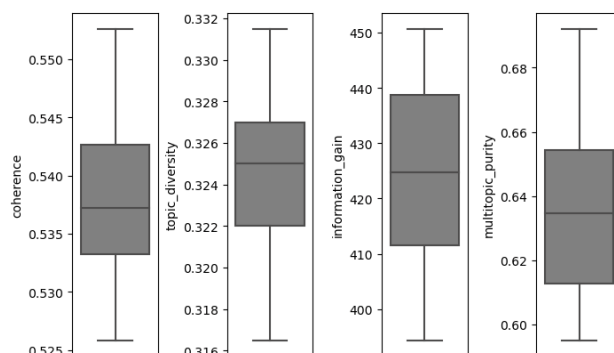


Figure 23: Box plots visualizing the obtained metrics for 25 runs of the prodLDA model, using the same hyperparameter configuration.

3.4.5 Expert evaluation

The best performing models of strategy 1 and strategy 3 were presented to domain experts for further manual evaluation. Of particular interest for this case study is the quality of the resulting clusters. To this end, two tasks were performed:



- **Task 1:** Given a cluster of comments, judge whether this cluster is too broad or not. If too broad, *split* the cluster in two or more smaller clusters.
- **Task 2:** Given all clusters, *merge* clusters consisting of comments covering the same topic.

To facilitate Task 2, the domain experts first listed the topic(s) covered by each comment appearing in one of the evaluated clusters. Due to time constraints, only a random subset of the clusters for each model were effectively evaluated. The table below summarizes the results. To facilitate comparison, the fraction of clusters split and merged relative to the total number of evaluated clusters is given as well.

	Model	#clusters	#evaluated clusters	#clusters split	#clusters after splitting	Merged clusters
Strategy 1 (multitopic)	ProdLDA	202	102	51 (50%)	195	30 (29%) clusters merged into 11 new clusters
Strategy 3 (hybrid)	ProdLDA	177	63	25 (40%)	100	15 (24%) clusters merged into 6 new clusters

The results in this table show that the resulting models are still far from perfect: for both models, (nearly) half the evaluated clusters were too broad and required further splitting, whereas around one fourth of the evaluated clusters were too narrow, requiring merges with other clusters covering related comments.

Since the considered models assign a score to each comment in the cluster indicating the probability that this comment belongs to the cluster, an interesting question is whether these probabilities can be used to identify clusters with a high potential of requiring further splitting, thereby potentially further improving the models. To this end, for each cluster the probabilities of all comments in this cluster are averaged resulting in a single score for each cluster. If a cluster is too broad, we expect this score to be lower as in general the comments assigned to this cluster will be less related to this cluster. Analogously, we look at the variance of comment probabilities for each cluster, expecting a higher variance for clusters split by the expert. The results of this analysis are visualized in Figure 24. In general, we see that the clusters being split by the expert typically consist of more comments, which can be expected, as larger clusters are more likely to be too broad. Unfortunately, the large overlap in size means that no meaningful threshold can be defined in terms of cluster size. Looking at the average comment probability for each cluster, there is no visible relationship between this value and the cluster being too broad. When considering the comment probability variance instead, we see that for the best performing model for strategy 3, clusters split by the expert indeed tend to have a higher variance, but for the best performing model for strategy 1, no such relationship is observed.

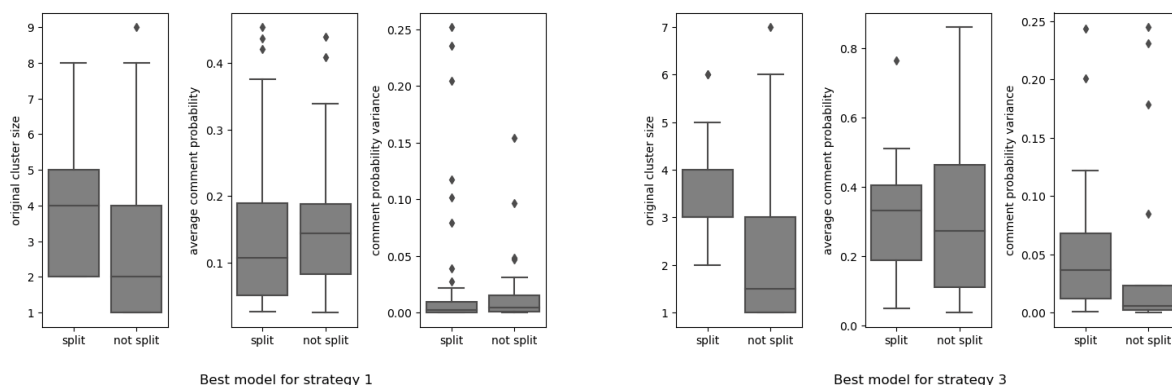


Figure 24: Box plots comparing the evaluated clusters that are split with those not split by the expert. (left) Best performing model for strategy 1. (right) Best performing model for strategy 3.

For clusters split by the expert, the distribution of comment probabilities is expected to be multimodal, as the groups of comments belonging together within this broad cluster should be highly related, and therefore have similar probabilities of belonging to this cluster. **Figure 25** and **Figure 26** visualize these distributions for the largest clusters evaluated by the experts for the best model for strategy 1 and strategy 3, respectively. From these figures, no significant difference in multimodal distribution is observed between clusters split by the expert and those not split by the expert.

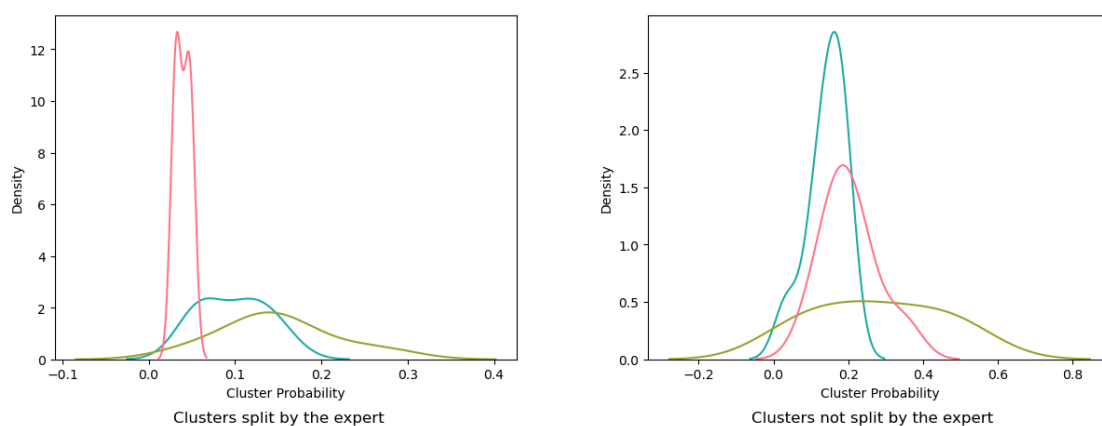


Figure 25: KDE plots visualizing cluster probabilities for comments belonging to the largest clusters (at least 7 comments in the cluster) evaluated by the expert for the best performing model for strategy 1. Each line represents the distribution for one such cluster. (left) Distributions for clusters split by the expert. (right) Distributions for clusters not split by the expert.

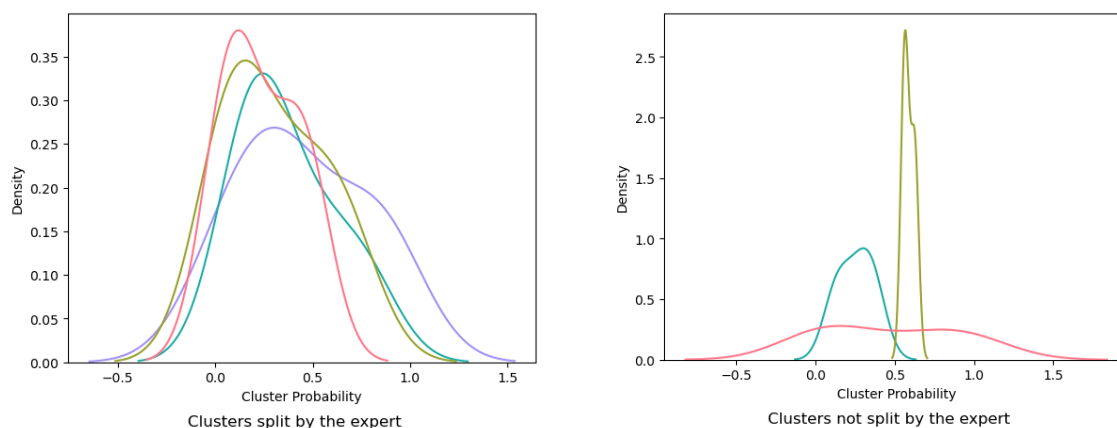


Figure 26: KDE plots visualizing cluster probabilities for comments belonging to the largest clusters (at least 5 comments in the cluster) evaluated by the expert for the best performing model for strategy 3. Each line represents the distribution for one such cluster. (left) Distributions for clusters split by the expert. (right) Distributions for clusters not split by the expert.

Cluster sizes and average comment probabilities are insufficient to predict whether clusters are too broad.

To estimate the time saved by the expert while answering the comments when starting from a topic model, we compare the number of lexical and semantical clusters with the original number of comments:

	#comments	#lexical clusters	#semantical clusters
Strategy 1 (multitopic)	723	493 (-32%)	202 (-59%)
Strategy 3 (hybrid)	723	493 (-32%)	177 (-64%)

Note that these percentages do not reflect a net time gain, as manual validation is still required. Manual validation of the lexical clusters revealed that there were no false positives (i.e., two comments with a different meaning ending up in the same lexical cluster). Furthermore, the proof-of-concept web app developed as part of this project further facilitates this manual validation of lexical clusters by highlighting the parts of a comment that are different from other comments in each lexical cluster. Manual validation of the semantical clusters requires more time to complete, as each comment in a semantical cluster must still be read entirely (note that these semantical clusters are over 493 comments due to the lexical clustering performed first, which should be contrasted with the reading all of the original 723 comments). Overall, a speedup of roughly 30% or more is realistic when starting from the output of a topic model instead of the original comments.

3.4.6 Conclusion

Currently, the state of the art topic modelling techniques do not allow (near) perfect clustering for small datasets with highly correlated text, meaning that expert involvement is still required for such a task. However, manual validation of the best



performing models by the domain experts showed that around half the considered clusters were identified as being not too broad, with the best performing model for strategy 3 slightly outperforming the best performing model for strategy 1. This indicates that the output of these models can still serve as a good initial clustering, thereby facilitating the task of the domain expert in finding related comments. A speedup of roughly 30% or more is realistic when starting from the output of a topic model instead of the original comments. A proof-of-concept web app that supports exploration of groupings has been developed as part of this project.

3.5 Assessment Case Study 3

We propose the following three topic modelling strategies:

1. Topic models based on variations of LDA (LDA, neural LDA, prodLDA, CTM) serve as a baseline.
2. Top2Vec, based on both pre-trained embeddings as well as an embedding fine-tuned over the given data by using TSDAE. Since the number of documents exceeds the recommended threshold of 10,000 documents (cf. Section 2.2.2.9), we expect this model to potentially outperform pre-trained embeddings. An important consideration is that the computational requirements for training such a model based on TSDAE cannot be derived from the literature. Without actual experiments, it is therefore still unclear whether training is practically achievable.
3. Since the corpus consists of titles and abstracts of scientific papers, SPECTER is expected to give better results than a general-purpose pre-trained embedding. We intend to use SPECTER within Top2Vec. However, the documentation of Top2Vec is vague on the possibility of including a custom embedding that is not an SBERT model or Universal Sentence Encoder. If including SPECTER into Top2Vec is not technically possible, we plan to apply UMAP and HDBSCAN directly on top of the computed SPECTER embeddings instead.

For each strategy, we report and compare topic information gain, coherence and diversity. Since a ground truth is not available (cf. Section 2.1.3), we will not report on cluster purity. Expert input can be used to provide additional validation.

3.6 Results Case Study 3

3.6.1 Models

Five classes of NLP models are trained on the provided dataset:

- LDA
- ProdLDA
- Neural LDA
- CTM
- Top2Vec

Before training, the documents are pre-processed based on the following parameters:

- **Punctuation and stopwords:** Basic text preprocessing is performed on each comment, involving removal of punctuation and stopwords.



- **Frequency filter low:** Denote this percentage L. Words which appear in less than L percent of all comments are filtered out. Due to the large number of documents and since an earlier analysis on this dataset (cf. Section 2.1.3) detected 400 topics with some topics only containing 15 documents, we choose a lower value of 0.005% to avoid throwing away too much information. For the given input data, this corresponds to removing all words occurring in only one or two documents.
- **Frequency filter high:** Denote this percentage H. Words which appear in more than H percent of all comments are filtered out. We pick 80% as a threshold, thereby filtering out all words that appear in the vast majority of documents. Since these words appear in almost all comments, it is not expected that they will contribute to the clustering. Further experiments for CS4 (cf. Section 3.8.3) indeed show that changing this parameter has barely any impact on the quality of the resulting models.
- **Stemming:** Indicator whether words are stemmed before applying the model. We follow the recommendations for each model and apply stemming for all models except Top2Vec.

The tuning parameters specific for each model are:

- **LDA:**
 - Num topics – tune values between 50 and 600
 - Alpha – symmetric, asymmetric or auto
 - Eta – symmetric or auto
 - Decay – tune values between 55% and 95%
- **Neural LDA:**
 - Num topics – tune values between 50 and 600
 - Activation – softplus or RELU
 - Solver – adam or sgd
 - momentum– tune values between 0.98 and 0.995
 - Num layers – tune values between 2 and 4
 - Num neurons – tune values between 80 and 150
 - Num samples – tune values between 5 and 15
- **Prod LDA:**
 - Same hyperparameters as Neural LDA
- **Top2Vec:**
 - Embedding – we consider a wide range of document embeddings:
 - *Pre-trained general-purpose embeddings:* universal-sentence-encoder and all-MiniLM-L6-v2
 - *Pre-trained specialized embeddings:* SPECTER
 - *Custom embeddings:* Doc2Vec and TSDAE
 - Min count – tune values between 1 and 100

The custom TSDAE embedding is trained on the complete dataset. We adhere to the configuration recommended by the TSDAE documentation²⁵, and use "bert-base-uncased" as the word embedding model to start from with a pooling layer based on the [CLS] token.

3.6.2 Evaluation Metrics

We consider the following three evaluation metrics, discussed in Section 2.2.4.4:

- Topic coherence
- Topic diversity
- Topic information gain

3.6.3 Optimizing for Topic Information Gain

For this case study, we assess whether topic modelling can be used as a tool to explore a large body of evidence more efficiently by summarizing the corpus into a reduced number of topics and to possibly reveal unknown topics. Because of this, both the grouping of related documents into topics as well as the description of each topic are important. Since topic information gain is the only metric taking both the description and the assigned documents into account for each topic, we use this metric as the optimization objective during model tuning. During tuning, a large parameter space is explored and multiple different hyperparameter configurations are tested. Each such configuration results in a trained model for which we can evaluate the different metrics. We emphasize that the other considered metrics are evaluated on each model as well during tuning, allowing us to detect dependencies between different metrics.

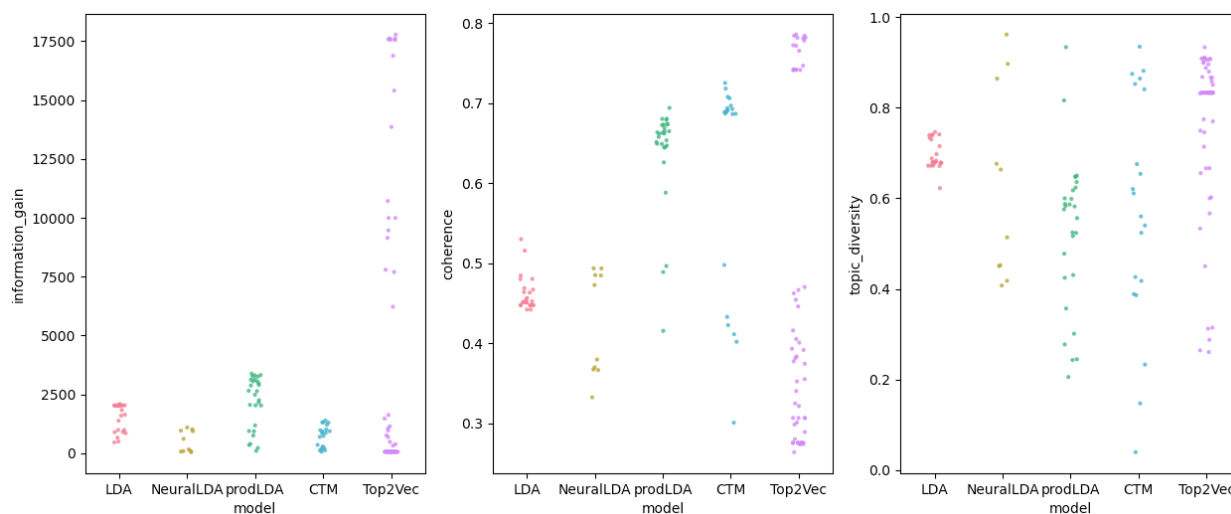


Figure 27: Overview of the different evaluation metrics for all trained models. Each dot represents a model trained based on a specific hyperparameter configuration during the tuning process.

Comparing the different models in Figure 27, we see that the best performing Top2Vec models clearly outperform all other models when looking at topic information gain and coherence. For topic information gain, the metric we are optimizing for, the difference is especially

²⁵ https://www.sbert.net/examples/unsupervised_learning/TSDAE/README.html

pronounced, with the best performing Top2vec models obtaining scores of 17500, whereas all other models never obtain a score above 5000. For topic diversity the difference between models is less pronounced, with most models (except LDA) having at least one model with a score close to 1. ProDLDA is the best performing LDA variant for topic information gain, and is on par with CTM when using topic coherence as evaluation metric. Similar to our findings for Case Study 2 (cf. Figure 16), we see that for some models, such as LDA, different hyperparameters have little effect on the performance of the model. For other models (Top2vec in particular), the obtained scores are more distributed, indicating that the performance of these models can be influenced significantly by performing hyperparameter tuning to find an optimal hyperparameter configuration.

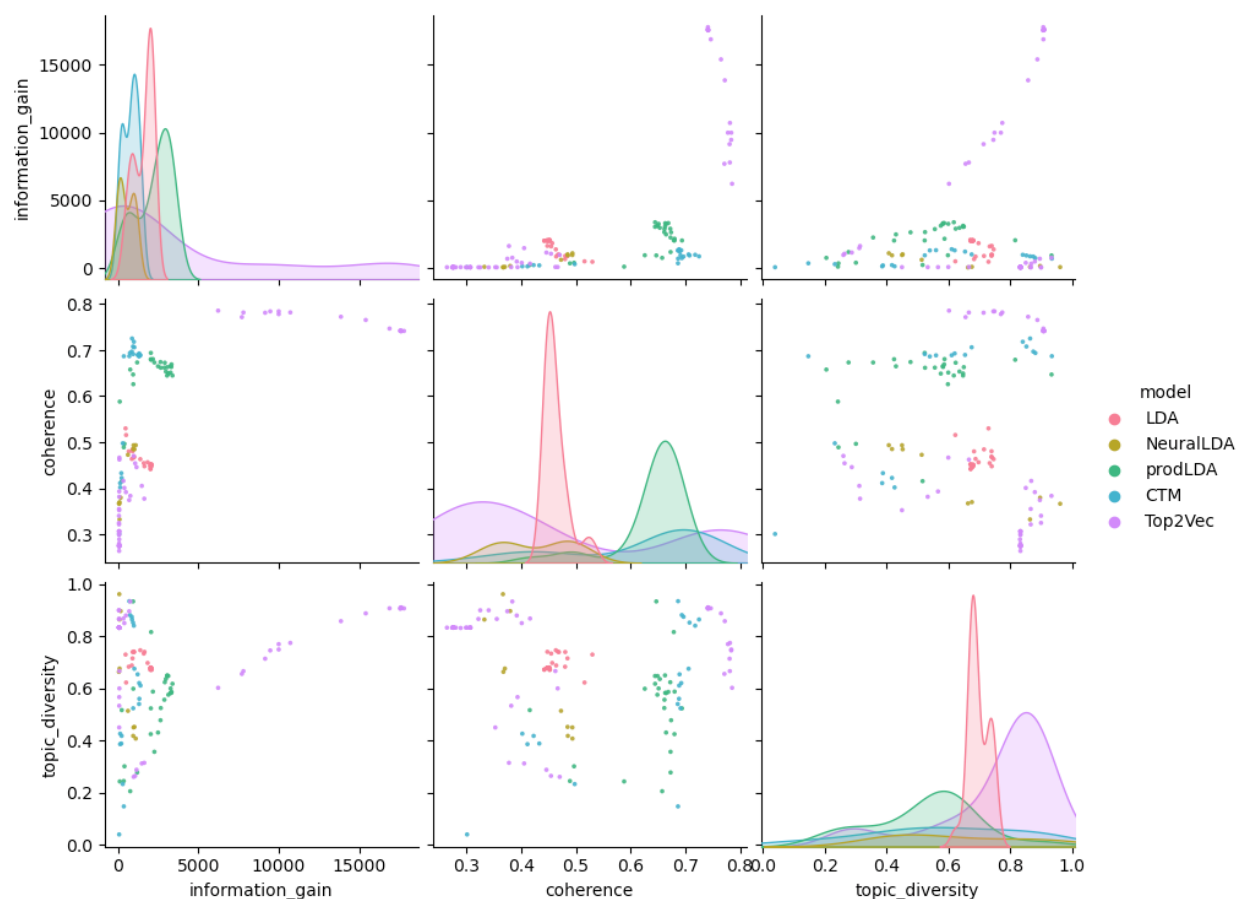


Figure 28: Overview of the relationships between different evaluation metrics.

Pairwise relationships between the three considered evaluation metrics are summarized in Figure 28. In most cases, there does not seem to be an immediate relationship between these metrics, except for the best performing Top2Vec models based on topic information gain, where a better topic information gain clearly implies a higher topic diversity. The opposite direction doesn't hold, as there are some Top2Vec models with a low score on topic information gain still have a high topic diversity.

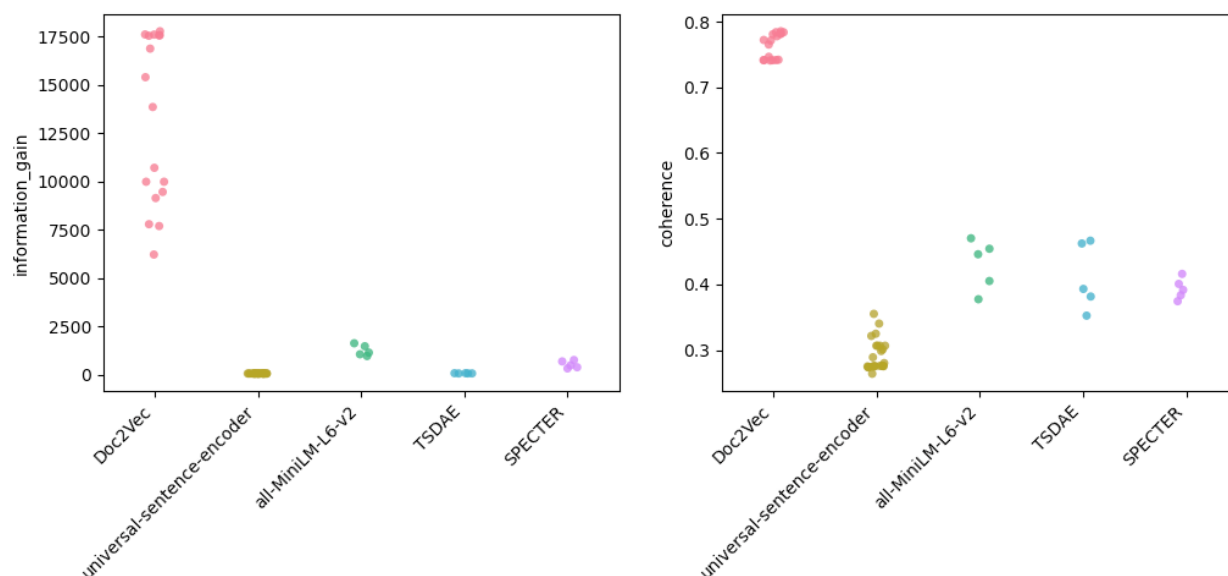


Figure 29: Topic information gain and topic coherence for different Top2Vec embeddings.

For this case study, we considered both pre-trained document embeddings (Universal-sentence-encoder, all-miniLM-L5-v2 and SPECTER) as well as custom document embeddings trained on the dataset at hand (Doc2Vec and TSDAE) as hyperparameter for the Top2Vec model. For each such embedding, a Top2vec model was finetuned using Bayesian optimization as discussed in Section 2.2.5, keeping track of the obtained metrics for each trained model during this optimization. These metrics are visualized in Figure 29. This figure shows that the best performing models obtained during tuning are all based on Doc2Vec embeddings, significantly outperforming the other embeddings. Surprisingly, both SPECTER (pre-trained on scientific text) and TSDAE (trained on the dataset at hand, using unsupervised learning) did not outperform the general-purpose pre-trained all-MiniLM-L6-v2 embedding. In fact, all embeddings based on an underlying BERT (cf. Section 2.2.1.5) model (i.e., all-MiniLM-L6-v2, TSDAE and SPECTER) obtained similar topic coherence scores, outperforming the embedding based on Universal Sentence Encoders (cf. Section 2.2.2.5).

Top2Vec generally outperforms the other models, with the chosen document embedding significantly influencing the results. Document embeddings based on Doc2Vec significantly outperform embeddings based on BERT or Universal Sentence Encoders. When choosing a BERT-based model for an unlabelled dataset, pre-trained embeddings are recommended over custom embeddings using unsupervised learning, as the latter require additional training without providing improved results.

In Figure 29, the topic information gain differs greatly between the models using Doc2Vec as embedding. As illustrated in Figure 30, this difference between models can be related to the “min count” hyperparameter for Top2Vec, with smaller values resulting in higher topic information gain and vice versa. Top2Vec ignores all words with a total frequency below this value, thereby indicating that removing infrequent words from the input data actually degrades performance in this case. This is not surprising, as for this case study we expect a

large number of topics with some of them containing only ten to twenty documents²⁶, rather than a small number of topics with each topic covering a large portion of the corpus. Removing too many infrequent words might therefore result in a loss of words relevant to smaller topics in the dataset.

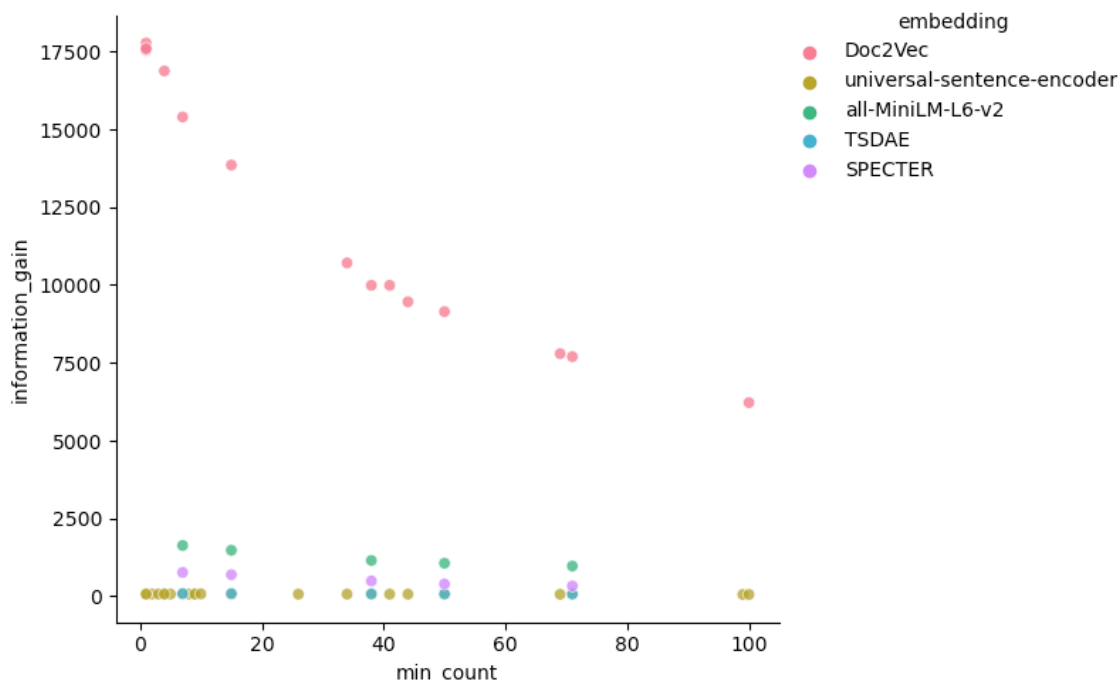


Figure 30: Influence of the “min count” hyperparameter for Top2Vec models.

For datasets where numerous topics are expected with each topic covering a small fraction of the corpus, the removal of infrequent words is not recommended. Such a removal is expected to rule out words relevant for these smaller topics, thereby reducing topic quality.

We end this section by providing a summary of the best performing model after tuning for each of the considered models. For Top2Vec, we provide the best performing model for each embedding. Notice in particular how some of the poor performing Top2Vec models are only able to identify a handful of topics, indicating that the quality of these document embeddings is insufficient for the underlying clustering algorithm to detect clusters of documents.

	Num topics	Information gain	Coherence	Diversity
LDA	598	2086	0.45	0.67
NeuralLDA	590	1117	0.50	0.41

²⁶ Earlier analysis identified 400 topics with the smallest topics containing only 15 documents (cf. Section 2.1.3)



<i>ProdLDA</i>	394	3610	0.66	0.62
<i>CTM</i>	226	1477	0.68	0.60
<i>Top2Vec – Doc2Vec</i>	453	18449	0.74	0.91
<i>Top2Vec – universal sentence encoder</i>	4	83	0.35	0.83
<i>Top2Vec – all-MiniLM-L6-v2</i>	286	1626	0.38	0.31
<i>Top2Vec – TSDAE</i>	3	77	0.46	0.67
<i>Top2Vec – SPECTER</i>	33	768	0.39	0.94

Although predicting precise scores for a new dataset is difficult, we expect that our findings will carry over to new datasets with similar characteristics. That is, Top2Vec outperforming the other considered topic models, with Doc2Vec significantly outperforming pre-trained embeddings.

3.6.4 Expert Evaluation

Further manual evaluation of the best performing model was performed by a domain expert. Due to the large dataset and high number of resulting topics, thorough evaluation would only be achievable for a fraction of the topics. To this end, a two-phased approach was followed with high-level evaluation over all topics during the first phase and a more in-depth evaluation for a small subset of potentially relevant topics during the second phase.

Phase 1

The eventual purpose of topic modelling for this case study is to explore a large body of evidence by summarizing the corpus into a reduced number of topics and to possibly reveal unknown topics. Due to the broad scope of the corpus, only a small fraction of the corpus is actually related to the case study at hand. Therefore, the domain expert ideally focusses the manual evaluation on relevant topics to get a better insight on the quality of such topic models. However, a topic modelling algorithm cannot differentiate between relevant and irrelevant topics (such a distinction would be case specific). Because of this, the goal of Phase 1 is to identify a set of keywords that can be used during Phase 2 to select topics for manual evaluation.

Task Phase 1: Given a set of topics where each topic is visualized by a word cloud consisting of the 50 words best describing the topic:

- Decide whether the topic is relevant to the use case; and
- List all words indicating inclusion and/or exclusion.

Notice that during Phase 1, only the topic description is evaluated. In particular, the documents assigned to each topic were not provided to the domain expert. After evaluating a Top2Vec model consisting of 453 topics,

- 95 (20.97%) topics were labelled as relevant,
- 337 (74.39%) topics were labelled as irrelevant, and



- 21 (4.64%) topics were labelled as unclear²⁷.

With less than 5% of the topics labelled as unclear, this labelling already indicates that the topic descriptions adequately describe the underlying topics.

Towards Phase 2, the expert compiled a list of words from the topic descriptions that are important for the case study (the inclusion list) and a list of words that are irrelevant for the case study (the exclusion list). The inclusion list consists of 288 unique words, whereas the exclusion list consists of 1078 unique words.

Phase 2

Using the exclusion words identified in Phase 1, 50 topics were selected that did not contain any of the exclusion words in their topic description. We emphasize that this selection based on exclusion words is merely intended to find a small but interesting set of topics for manual evaluation, and not as a technique to automatically derive all topics relevant to the use case at hand. Indeed, since new models can have different topic descriptions, manual evaluation performed by a domain expert is still required to avoid overlooking topics containing relevant documents.

As input to Phase 2, the topic descriptions of the 50 select topics were provided as word clouds, as well as the documents assigned to each topic. Due to time constraints, evaluating each document assigned to one of these topics was not achievable. Instead, for each topic the documents were sorted by decreasing probability of belonging to the topic, and only the top-5 and bottom-5 documents were evaluated.

Task Phase 2: For each of the selected 50 topics, the following three tasks are to be completed:

1. Indicate whether the topic is indeed relevant to the use case (Yes/No/Unclear)
2. Rate the quality of the word cloud (Good/Mediocre/Bad). As a rule of thumb, a word cloud is good if it clearly describes a single, coherent topic.
 - "Good": The word cloud describes a single coherent topic
 - "Mediocre": The words in the word cloud are less coherent, two or three topics can be identified
 - "Bad": The topic is not clear from the word cloud
3. For the top-5 and bottom-5 documents belonging to this topic, decide whether this document is indeed related to the topic at hand (Yes/No/Unclear).

To facilitate exploring the topics and performing the task, a web app was provided to the domain expert.

Out of the 50 topics, 31 (62%) were identified as relevant, and the remaining 19 (38%) were identified as not relevant. Thereby indicating that a keyword-based filtering with keywords from a compiled exclusion list is indeed not recommended to select relevant topics.

With 49 (98%) wordclouds labelled as good, 1 (2%) topic description labelled as mediocre and no topic descriptions labelled as bad, this in-depth evaluation confirms our earlier conclusion that the topic descriptions adequately describe the underlying topics.

²⁷ Since the domain expert could only evaluate the topic descriptions, "unclear" means that the topic description contains insufficient information to decide whether or not the topic is indeed relevant.



For each topic, the top-5 documents were always identified as related to the topic. For the bottom-5 documents of each topic, only 2 out of these 250 evaluated documents (0.8%) were identified as not related to the assigned topic.

Time gained

Providing a rough estimate of the time saved by using a topic model when exploring a large corpus is difficult, and largely depends on the number of documents that can be interpreted more quickly by leveraging the output of the topic model. In particular, the word clouds provided by the model must be accurate and easy to interpret to allow the domain expert to explore groups of documents more efficiently, without having to read each document separately.

3.6.5 Conclusion

The state-of-the-art topic modelling algorithm results in qualitative topics with good topic descriptions. However, it is important to note that topic models cannot classify documents based on a classification specific to the use case at hand. For example, they cannot differentiate between topics relevant to a specific use case and topics irrelevant to this use case. Topic modelling should therefore be situated as a tool to aid the domain expert in exploring and classifying documents in a large corpus more efficiently, rather than a fully automated replacement. Providing a rough estimate of potential speedup by using topic models is hard, but the time saved is expected to be significant, especially when a larger number of documents can be discarded quickly by only looking at the word cloud for each topic. A proof-of-concept web app that supports exploration of and interaction with topics has been developed as part of this project.

3.7 Assessment Case Study 4

Since the number of documents for this case study is too small to expect custom models to outperform pre-trained document models, we focus instead on a wider range of pre-trained models and compare Top2Vec to BERTopic. We propose the following three strategies:

1. Topic models based on a variation of LDA (LDA, neural LDA, prodLDA) serves as a baseline. Since Case Study 2 has a ground truth, we will use the best performing variation of LDA according to Case Study 2.
2. Top2Vec on a wide range of document embeddings (Doc2Vec, pre-trained SBERT models, pre-trained Universal Sentence Encoder models)
3. BERTopic on a wide range of document embeddings (Doc2Vec, pre-trained SBERT models, pre-trained Universal Sentence Encoder models)

Although Doc2Vec is not a pre-trained model and therefore not assumed to outperform the pre-trained models, we still include it in the latter two strategies as it is readily available in both frameworks and therefore a straightforward technique to validate this assumption.

For each strategy, we report and compare topic information gain, coherence and diversity. Since a ground truth topic model is not available (cf. Section 2.1.4), we will not report on cluster purity relative to such a topic model.



Since each document is manually labelled as relevant or not relative to beeswax adulteration, we propose a different evaluation metric based on cluster purity over these two labels to compare topic models. More formally, we construct two classes: one with all documents considered relevant, and another class with all other documents. As cluster purity relative to these two classes measures how well each cluster contains either relevant or non-relevant documents, we argue that this metric is a good indication of usefulness for the task at hand. Indeed, topics containing a mixture of relevant and non-relevant documents still require manual screening of each document within this topic, and therefore do not reduce the time needed to screen for useful documents.

3.8 Results Case Study 4

3.8.1 Models

Five classes of NLP models are trained on case study 4:

- LDA (standard version)
- Neural LDA
- Prod LDA
- Top2Vec
- BERTopic

When calibrating these models to the documents a number of hyperparameters have to be chosen or tuned. These parameters can be split into preprocessing parameters and model-specific parameters. The preprocessing parameters are:

- Edit distance, this is included as a percentage. Articles which differ less than the set percentage are grouped and included in the model as a single document. We tune for values between 2% and 20%.
- Include metadata: Each article has a single metadata attribute, namely the site from which the article was fetched. We investigate whether adding this information to the model improves classification.
- Frequency filter low: Denote this percentage L. Words which appear in less than L percent of all documents are filtered out. We tune for values between 0% and 20%
- Frequency filter high: Denote this percentage H. Words which appear in more than H percent of all documents are filtered out. We tune for values between 80% and 100%
- Stemming: Indicator whether words are stemmed before applying the model.

The model specific parameters are:

- LDA (standard):
 - Alpha – symmetric, assymmetric or auto
 - Eta – symmetric or auto
 - Decay – tune values between 55% and 95%
 - Num topics – tune values between 5 and 50



- Neural LDA:
 - Activation – softplus or RELU
 - Num topics – tune values between 5 and 50
 - Dropout – tune values between 0 and 50%
 - Learn priors – True or False
 - Learning rate – Tune values between 1/16 and ¼
- Prod LDA:
 - Same hyperparameters as Neural LDA
- Top2Vec:
 - Embedding – doc2vec, universal-sentence-encoder, universal-sentence-encoder-larger, universal-sentence-encoder-multilingual
- BERTopic:
 - Top n words – tune values between 5 and 15
 - Min topic size – tune values between 5 and 20
 - Num topics – tune values between 5 and 50
 - Diversity – tune values between 0% and 100%

3.8.2 Evaluation Metrics

Evaluation metrics are split into internal and external metrics. Internal metrics are computed based on the available documents only, whereas external metrics also use the classification by the case expert. External metrics will be unavailable in future scenarios, hence the goal is to choose the internal metric that best optimizes our external objective.

Internal metrics:

- Coherence
- Diversity – fraction of unique keywords for each topic
- Stability – All models are stochastic, which means that fitting the same model twice will give different results. Stability measures the similarity between these models.

External metrics:

- Cluster purity
- Cluster entropy
- Penalized cluster entropy
- Penalized cluster purity

Since cluster purity and entropy are maximal when each document has its own topic, we add a penalty term which penalizes the model for having many topics. If two models explain the data equally well, we prefer the model with less topics as it is easier to interpret and evaluate.

3.8.3 Optimizing for Penalized Entropy

We start by tuning the hyperparameters based on the external metric penalized entropy. This metric is not available for new data sets. We optimize penalized entropy to determine a good preprocessing strategy and to choose reasonable starting values for new data sets that are similar. We let the Bayesian optimization algorithm test between 50 and 100 parameters for each of the models.

Figure 31 shows the evaluation metrics obtained for the different models during hyperparameter tuning. The models were trying to optimize penalized entropy. BERTopic and Doc2Vec outperformed LDA and its variations in all three external metrics. BERTopic selects a smaller number of topics than Doc2Vec, which gives it a small advantage on the penalized metrics.

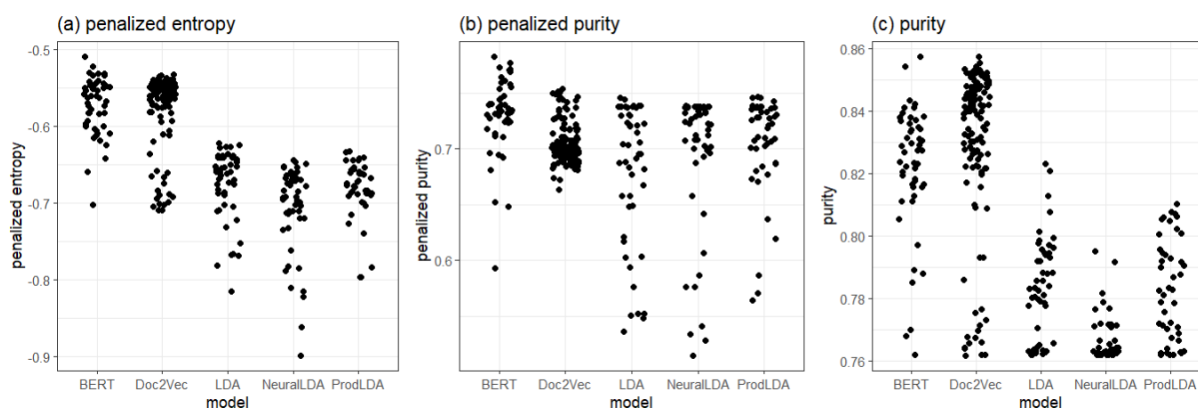


Figure 31: External evaluation metrics for the trained NLP model while optimizing penalized entropy.

In Figure 32 below we evaluate the models on the internal metrics. Coherence is the most likely candidate for a tuning metric when external data is not available. We see however that there are large variations in coherence between model fits. BERT and Doc2Vec can obtain higher coherence levels than the LDA variations. Diversity indicates how often keywords are repeated across topics. Although having different keywords for each topic is a nice to have, this is not a criterium based on which we can select our NLP model.

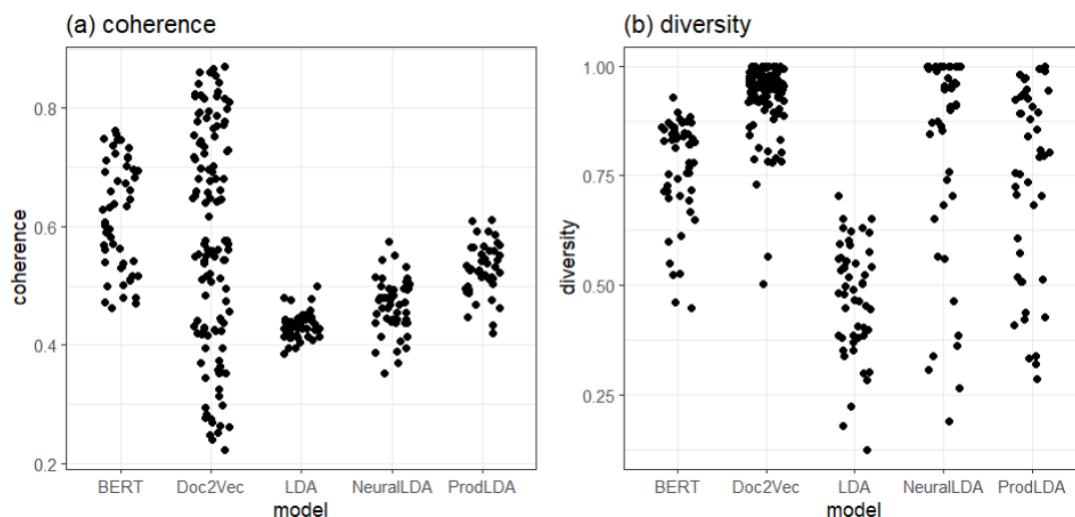


Figure 32: Internal (unsupervised) evaluation metrics for the trained NLP models while optimizing penalized entropy.

Figure 33 shows the relation between coherence and penalized entropy in the fitted models. High penalized entropy does not imply that the model also has a high coherence. However, the reverse might still hold as most models with high coherence values on this figure also appear to have a high penalized entropy. Section 3.7.5 investigates this reverse relation further.

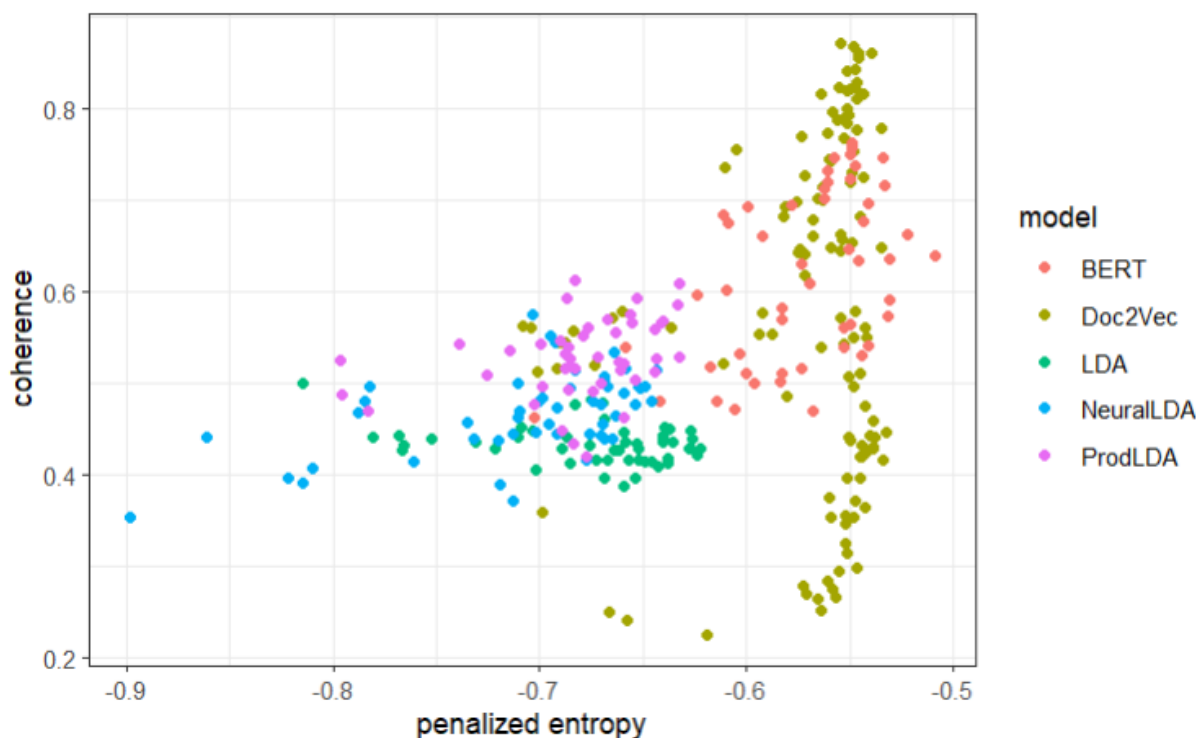


Figure 33: Relation between penalized entropy (external tuning metric) and coherence (internal evaluation metric)



We want to use this tuning based on the external metric penalized entropy to choose a fixed data preprocessing strategy. Later when we are optimizing for an internal metric, we will then only focus on the model specific parameters. Figure 34 shows penalized entropy as a function of the preprocessing parameters in the various models.

- Edit distance: Grouping documents based on edit distance does not appear to have a large effect on the quality of the fit. This is not unexpected, as these are news articles which mostly have large edit distances. Only a small number of papers can be grouped in this way. Since the impact is minimal we suggest to not group the documents based on edit distance in this case study.
- Meta data: We see a small benefit of including metadata (source URL of the article) in the model. In general we suggest to include this data in the model and let the NLP model decide how to use it.
- Frequency filter: This is the parameter for which we see the largest effect. When we filter too many infrequent words the performance of BERT and Doc2Vec decreases. Words that appear in many documents are less important. We choose:
 - Frequency filter low: 3%
 - Frequency filter high: 95%
- Stemming: Stemming is not needed/suggested for models using an embedding (Doc2Vec, BERTopic). There is a small benefit when using LDA. We choose:
 - LDA → Stemming
 - Doc2Vec, BERTopic → No stemming

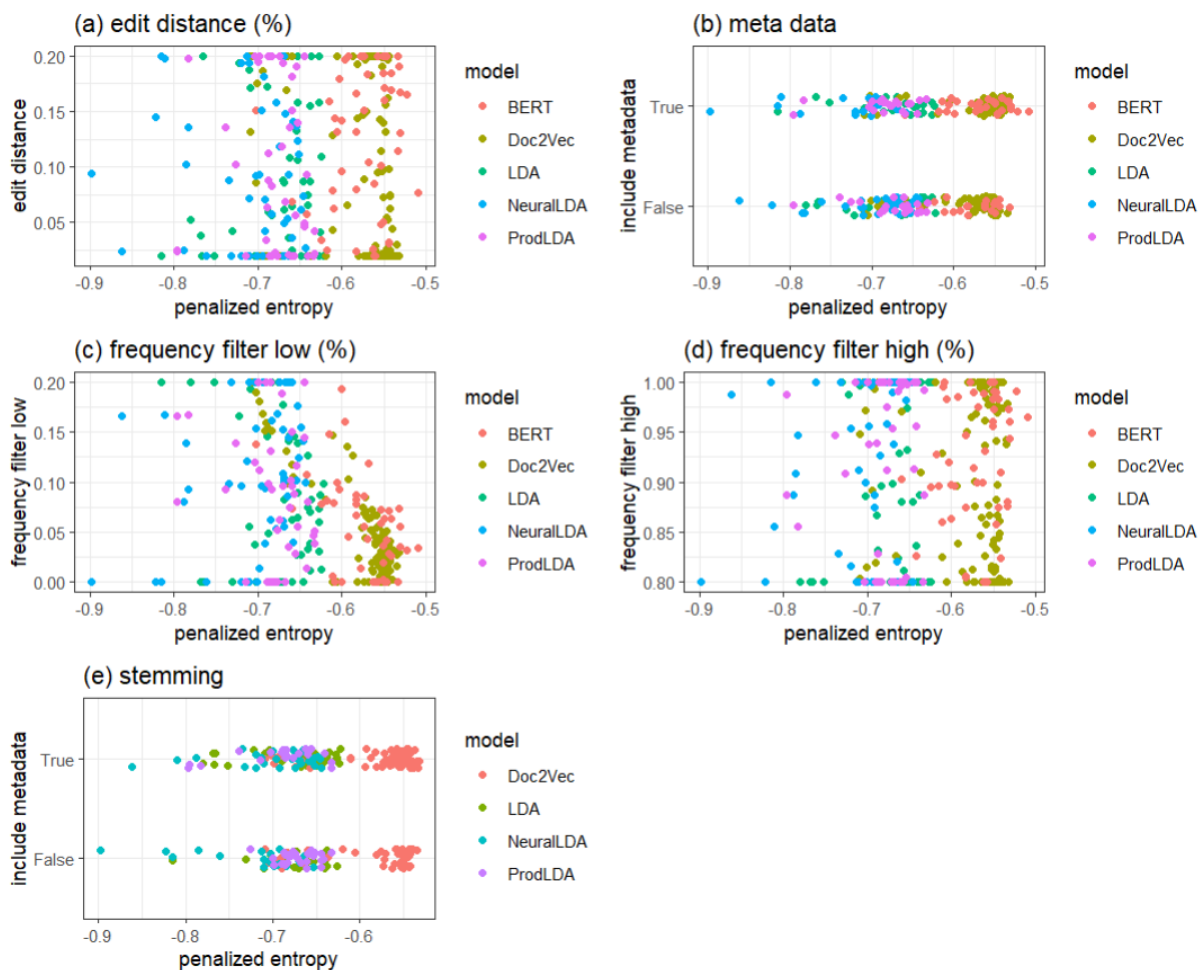


Figure 34: Effect of hyperparameters on penalized entropy in fitted NLP models.

The improvement of preprocessing NLP models before training is limited. The most important preprocessing step is the removal of words that either appear in almost all documents or appear in only a few documents. The optimal preprocessing parameters are used as the default values for training new models.

3.8.4 Optimizing for coherence

We continue investigating the calibration method when optimizing for coherence. This is a realistic scenario that can be applied to new data sets. We still compute penalized entropy to evaluate the performance of the optimization strategy.

We use the preprocessing parameters determined in section 4. As all models follow the same preprocessing steps, the stability across fits can be evaluated. Hyperparameter tuning now considers only the model specific parameters. The number of evaluated points is lower for Top2Vec as we only tune the choice of the embedding.

Figure 35 plots the internal metrics coherence and stability against penalized entropy. Unfortunately optimizing for high coherence does not automatically result in a good penalized

entropy score. Surprisingly, stability would be a better tuning metric as it is more strongly associated to high penalized entropy.

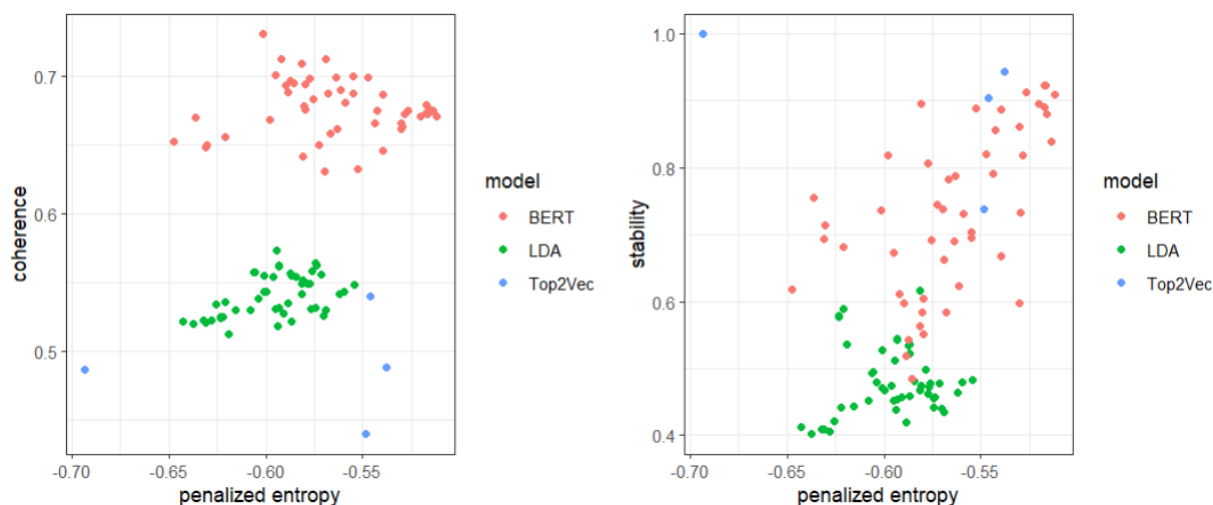


Figure 35: Relation between penalized entropy (external evaluation metric) and coherence (internal evaluation metric) when tuning for coherence.

We look in more detail to some of the best selected models

- BERTopic (coherence), the selected BERTopic model when optimizing for coherence
- BERTopic (stability), the selected BERTopic model when optimizing for stability
- Top2Vec, the selected Top2Vec embedding when optimizing for coherence
- LDA, the selected LDA model when optimizing for coherence

Some properties of these models:

	<i>BERTopic</i> (coherence)	<i>BERTopic</i> (stability)	<i>Top2Vec</i>	<i>LDA</i>
<i># Topics</i>	8	6	32	7
<i>Penalized entropy</i>	-0.62	-0.57	-0.55	-0.57
<i>Diversity</i>	0.85	0.92	0.67	0.86

After optimization all models have a similar penalized entropy. When optimizing for coherence or stability, LDA and BERTopic tend to prefer a small number of topics, whereas Top2Vec retains a larger number of topics. The higher number of topics retained in Top2Vec also implies that these topics are less distinct and hence have more keywords in common which results in a lower diversity.

So far, we have only looked at the models quantitatively. A final step will be to assess which model produces the best qualitative results based on human judgement. The case expert does this by answering questions related to the clustered documents by these models. An example task is given in appendix B.

Automatic evaluation of NLP clustering using evaluation metrics remains a hard task. The unsupervised model might find different groupings than the one of interest to the expert. In this study, none of the unsupervised evaluation metrics aligned perfectly with the supervised clustering by the expert.

3.8.5 Optimizing for expert judgement

The next two tables, summarize the results of the expert evaluation of both the descriptive keywords of the topics and the relevance of the newly suggested articles. In both cases, Top2Vec clearly outperforms BERTopic and LDA. In two-third of all tasks, Top2Vec provided an accurate or mediocre description of the article and in half of the cases the newly proposed article was relevant for the cluster. Performance is significantly better than that of the other two models, but the metrics remain low from a human viewpoint. This indicates that the task at hand is difficult to learn by automated algorithms and using expert judgement in model selection adds a lot of value to the modelling process.

Quality of the keywords identified by the model:

Model	Good	Mediocre	Bad
BERTopic (coherence)	5.7%	2.9%	91.4%
BERTopic (stability)	0%	4.3%	95.7%
Top2Vec	45.7%	18.6%	35.7%
LDA	13%	15.9%	71.0%

Relevance of the suggested article, based on the ones seen before:

Model	Good	Bad
BERTopic (coherence)	26.1%	73.9%
BERTopic (stability)	11.4%	88.6%
Top2Vec	47.8%	52.2%
LDA	20.6%	79.4%

Expert evaluation of trained NLP models remains important. From the expert evaluation, we learned that the results obtained by the `Top2Vec` align best with human interpretation.

3.8.6 Conclusion



Automated clustering of newspapers through NLP models is a complex undertaking due to the wide variety of topics covered and the articles being written for diverse audiences by numerous authors. In this regard, we have identified three key insights that could inform future NLP models. Firstly, hyperparameter tuning and preprocessing may only yield limited improvements and should be implemented when the base model's performance is already near deployment standards. Secondly, unsupervised clustering has a high chance of finding different clusters from those intended by domain experts. Lastly, expert evaluation is crucial in selecting a model that aligns abstract numerical metrics with human interpretation.

4 Conclusion

In this report, we covered a wide range of existing word and document embeddings. For word embeddings we discussed Word2Vec, GloVe, FastText, ELMo, and BERT. For document embeddings we discussed TF-IDF vectors, averaging word embeddings, fine-tuning BERT for document similarity, universal sentence encoder, SimCSE, Sentence-T5, TSDAE, GPL and SPECTER. The applicability of these models depends on a number of task-specific factors and requirements, such as the need for context-sensitive word embeddings, the availability of labelled data for supervised training or fine-tuning and the size of the dataset that can be used for (self-)supervised training or fine-tuning.

Fine-tuning or creating document embeddings from scratch is only feasible in the presence of enough data and has an associated computational cost. When there are more than 10000 documents available TSDAE could be considered for training document embeddings. If no or only little data is available, pre-trained embeddings are often the better choice. It was found that a multitude of increasingly more complex pre-trained embeddings are readily available for off-the-shelf use. But as they are trained on large but mostly general text corpora, their utility for domain specific text varies. For some domains (like scientific articles), the pre-trained sentence embedding SPECTER can be used.

For topic modelling, we discussed standard techniques like non-negative matrix factorization (NMF) and latent Dirichlet allocation (LDA), including variations of LDA (Neural LDA, ProdLDA and CTM), as well as more recent methods based on clustering of document embeddings like Top2Vec and BERTopic. It was found that LDA provides a good baseline and is helpful for situations that require mixed topics. Top2Vec is a very promising method and allows the possibility to use various document embedding methods.

For text classification, we consider hierarchical text classification approaches combined with established techniques for text classification via document embeddings as XGBoost, support vector machines and FastText. Some case studies require segmentation of the input for which we consider a structural as well as a semantic approach called TextTiling.

Training and fine-tuning document embeddings is still an active research area, with a lot of recent advances that push the state-of-the-art forward. Given these ongoing research advances, it is expected that topic modelling and text classification methods leveraging these document embeddings can benefit from their increasing accuracy, thereby resulting in improved topic and text classification models. We summarize our findings on model selection and hyperparameter finetuning in Section 4.1.

To assess the effectiveness of topic modelling we considered the following metrics: topic information gain, topic coherence, topic diversity and cluster purity. Of these, topic coherence has been shown to reflect human judgment. Cluster purity is a metric that requires a ground truth to be available.

The main conclusions are as follows:

Case Study 1:

- All tested NLP models performed equally well in predicting the APRIO labels. The models are capable of retrieving 80% of the labels given to each document, and 20% of the labels attributed by the models were wrong. As a result, this task cannot be fully automated and domain experts still have to carefully review the predicted labels. Models based on TF-IDF have the additional advantage that most important features

are interpretable. For this case study, we hence propose the TF-IDF with the non-hierarchical XGBoost strategy for classification. In applications where preserving the hierarchical structure is essential, this hierarchical structure can be restored by applying a filtering step on the results from the non-hierarchical model.

- Performance of the NLP model depends on the question, pillar and subquestion at hand. The models perform very well on labels which are frequently attributed and have a specific vocabulary. The models should not be used to predict infrequent labels. As more manually labeled documents become available, the set of labels for which NLP algorithms can be used will increase.
- A proof-of-concept web app that supports exploration of assigned labels has been developed as part of this project.

Case Study 2:

- The majority of comments on the given opinion are highly correlated. Close relatedness between most of the comments provides an additional challenge for topic models, as the boundaries between topics will be less pronounced.
- Due to the small number of comments and high correlation between comments, baseline models (ProdLDA in particular) are on par with or outperform the more complex Top2Vec models when considering evaluation metrics relevant for this case study.
- For new datasets without a ground truth, topic information gain is a good choice for model evaluation during optimization. We refer to Section 4.1 for more details on hyperparameter optimization.
- Currently, the state-of-the-art topic modelling techniques do not allow (near) perfect clustering for small datasets with highly correlated text, meaning that expert involvement is still required for such a task. However, manual validation of the best performing models by the domain experts showed that around half the considered clusters were identified as being not too broad. This indicates that the output of these models can still serve as a good initial clustering, thereby facilitating the task of the domain expert in finding related comments. A speedup of roughly 30% or more is realistic when starting from the output of a topic model instead of the original comments. A proof-of-concept web app that supports exploration of groupings has been developed as part of this project.

Case Study 3:

- Top2Vec generally outperforms the other models, with the chosen embedding significantly influencing the results. Document embeddings based on Doc2Vec significantly outperform embeddings based on BERT or Universal Sentence Encoders. When choosing a BERT-based model for an unlabelled dataset, pre-trained embeddings are recommended over custom embedding using unsupervised learning, as the latter require additional training without providing improved results.
- For datasets where numerous topics are expected with each topic covering a small fraction of the corpus, the removal of infrequent words is not recommended. Such a removal is expected to rule out words relevant for these smaller topics, thereby reducing topic quality. We refer to Section 4.1 for more details on hyperparameter optimization.
- The state-of-the-art topic modelling algorithm results in qualitative topics with good topic descriptions, thereby facilitating the task of a domain expert to screen this corpus and to allow scoping of the literature to reveal possible topics that were unknown. However, it is important to note that topic models cannot classify documents based on a classification specific to the use case at hand. For example, they cannot differentiate

between topics relevant to a specific use case and topics irrelevant to this use case. Topic modelling should therefore be situated as a tool to aid the domain expert in exploring and classifying documents in a large corpus more efficiently, rather than a fully automated replacement. Providing a rough estimate of potential speedup by using topic models is hard, but the time saved is expected to be significant, especially when a larger number of documents can be explored quickly by only looking at the word cloud for each topic. A proof-of-concept web app that supports exploration of and interaction with topics has been developed as part of this project.

Case Study 4:

- The improvement of preprocessing NLP models before training is limited. The most important preprocessing step is the removal of words that either appear in almost all documents or appear in only a few documents. The optimal preprocessing parameters are used as the default values for training new models. We refer to Section 4.1 for more details on hyperparameter optimization.
- Automatic evaluation of NLP clustering using evaluation metrics remains a hard task. The unsupervised model might find different groupings than the one of interest to the expert. In this study, none of the unsupervised evaluation metrics aligned perfectly with the supervised clustering by the expert.
- Expert evaluation of trained NLP models remains important. From the expert evaluation, we learned that the results obtained by the Top2Vec model align best with human interpretation.
- Automated clustering of newspapers through NLP models is a complex undertaking due to the wide variety of topics covered and the articles being written for diverse audiences by numerous authors. In this regard, we have identified three key insights that could inform future NLP models. Firstly, hyperparameter tuning and preprocessing may only yield limited improvements and should be implemented when the base model's performance is already near deployment standards (see Section 4.1 for more details on hyperparameter tuning). Secondly, unsupervised clustering has a high chance of finding different clusters from those intended by domain experts. Lastly, expert evaluation is crucial in selecting a model that aligns abstract numerical metrics with human interpretation.

4.1 Model Selection and Hyperparameter Finetuning

For each case study, we implemented and evaluated multiple models, applying Bayesian optimization to explore optimal hyperparameter configurations for each model (cf. Section 2.2.5). Our main conclusions on model selection and hyperparameter finetuning are as follows:


- For text classification, no significant difference in performance between models is observed (cf. Case Study 1). Models facilitating interpretability of the results are therefore recommended over those that do not allow this. In general, the performance of these models mostly depends on the availability of training data, rather than model selection and hyperparameter tuning.
- For topic modelling, the recommended model largely depends on the corpus at hand. For smaller datasets with highly related documents (cf. Case Study 2), baseline models such as prodLDA are recommended since more complex models are more prone to overfitting, thereby reducing performance. For larger datasets (cf. Case Study 3 and Case Study 4), Top2Vec is recommended.
- When using a model based on document embeddings (e.g. Top2Vec or BERTopic), the chosen document embedding has a significant impact on the final performance of the

topic model. In particular, Doc2Vec significantly outperforms the other considered document embedding models.


- Hyperparameter finetuning does allow for further performance improvements, but the potential performance gain depends on the chosen model. Overall, the impact of hyperparameter tuning was more pronounced on Top2Vec models when compared to the different baseline models based on variations of LDA. Furthermore, our analysis reveals that some hyperparameters have little to no effect on the resulting performance, indicating that optimizing for these hyperparameters is less relevant.
- Hyperparameter tuning should be considered as a tool to further improve the performance of models with a reasonable base performance, and therefore is not recommended for models with very poor initial performance. If the output of the base model is still far from desired, hyperparameter tuning is not expected to improve this model to deployment standards.

References

- [1] D. Angelov, *Top2Vec: Distributed Representations of Topics*, CoRR, abs/2008.09470 (2020).
- [2] F. Archetti and A. Candelieri, *Bayesian Optimization and Data Science*, Springer International Publishing, 2019.
- [3] I. Beltagy, K. Lo and A. Cohan, *SciBERT: A Pretrained Language Model for Scientific Text*, 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*, Association for Computational Linguistics, 2019, pp. 3613-3618.
- [4] F. Bianchi, S. Terragni, D. Hovy, D. Nozza and E. 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, EACL 2021*, Association for Computational Linguistics, 2021, pp. 1676-1683.
- [5] D. M. Blei, A. Y. Ng and M. I. Jordan, *Latent Dirichlet Allocation*, *Journal of Machine Learning Research*, 3 (2003), pp. 993-1022.
- [6] H. Blockeel, L. Schietgat, J. Struyf, S. Dzeroski and A. Clare, *Decision Trees for Hierarchical Multilabel Classification: A Case Study in Functional Genomics*, in *Proceedings of the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases, PKDD 2006*, Springer, 2006, pp. 18-29.
- [7] P. Bojanowski, E. Grave, A. Joulin and T. Mikolov, *Enriching Word Vectors with Subword Information*, *Transactions of the Association for Computational Linguistics*, 5 (2017), pp. 135-146.
- [8] G. Bouma, *Normalized (pointwise) mutual information in collocation extraction*, *Proceedings of the Biennial GSCL Conference*, 30 (2009), pp. 31-40.
- [9] R. J. G. B. Campello, D. Moulavi and J. Sander, *Density-Based Clustering Based on Hierarchical Density Estimates*, in *Advances in Knowledge Discovery and Data Mining, 17th Pacific-Asia Conference, PAKDD 2013*, Springer, 2013, pp. 160-172.
- [10] R. J. G. B. Campello, D. Moulavi, A. Zimek and J. Sander, *Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection*, *ACM Transactions on Knowledge Discovery from Data*, 10 (2015), pp. 5:1-5:51.
- [11] D. Cer, Y. Yang, S. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. GuajardoCespedes, S. Yuan, C. Tar, Y. Sung, B. Strope and R. Kurzweil, *Universal Sentence Encoder*, CoRR, abs/1803.11175 (2018).
- [12] D. M. Cer, M. T. Diab, E. Agirre, I. LopezGazpio and L. Specia, *SemEval-2017 Task 1: Semantic Textual Similarity - Multilingual and Cross-lingual Focused Evaluation*, CoRR, abs/1708.00055 (2017).
- [13] C. Chelba, T. Mikolov, M. Schuster, Q. Ge, T. Brants, P. Koehn and T. Robinson, *One billion word benchmark for measuring progress in statistical language modeling*, in *INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association*, ISCA, 2014, pp. 2635-2639.
- [14] T. Chen and C. Guestrin, *XGBoost: A Scalable Tree Boosting System*, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016*, ACM, 2016, pp. 785-794.

- 
- [15] A. Cohan, S. Feldman, I. Beltagy, D. Downey and D. S. Weld, *SPECTER: Document-level Representation Learning using Citation-informed Transformers*, in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*, Association for Computational Linguistics, 2020, pp. 2270-2282.
- [16] J. Devlin, M. Chang, K. Lee and K. Toutanova, *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, NAACL-HLT 2019*, Association for Computational Linguistics, 2019, pp. 4171-4186.
- [17] A. B. Dieng, F. J. R. Ruiz and D. M. Blei, *Topic Modeling in Embedding Spaces*, *Transactions of the Association for Computational Linguistics*, 8 (2020), pp. 439-453.
- [18] M. Ester, H. Kriegel, J. Sander and X. Xu, *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*, in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD 1996*, AAAI Press, 1996, pp. 226-231.
- [19] T. Gao, X. Yao and D. Chen, *SimCSE: Simple Contrastive Learning of Sentence Embeddings*, in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*, Association for Computational Linguistics, 2021, pp. 6894--6910.
- [20] J. Gareth, W. Daniela, H. Trevor and T. Robert, *An introduction to statistical learning: with applications in R*, Springer, 2013.
- [21] M. Grootendorst, *BERTopic: Neural topic modeling with a class-based TF-IDF procedure*, *CoRR*, abs/2203.05794 (2022).
- [22] M. A. Hearst, *TextTiling: Segmenting Text into Multi-paragraph Subtopic Passages*, *Computational Linguistics*, 23 (1997), pp. 33-64.
- [23] M. Iyyer, V. Manjunatha, J. L. BoydGraber and H. Daum III, *Deep Unordered Composition Rivals Syntactic Methods for Text Classification*, in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015*, Association for Computational Linguistics, 2015, pp. 1681-1691.
- [24] J. H. Lau, D. Newman and T. Baldwin, *Machine Reading Tea Leaves: Automatically Evaluating Topic Coherence and Topic Model Quality*, in *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014*, Association for Computational Linguistics, 2014, pp. 530-539.
- [25] Q. V. Le and T. Mikolov, *Distributed Representations of Sentences and Documents*, in *Proceedings of the 31th International Conference on Machine Learning, ICML 2014*, JMLR.org, 2014, pp. 1188-1196.
- [26] D. Lee and H. S. Seung, *Algorithms for Non-negative Matrix Factorization*, in *Advances in Neural Information Processing Systems*, MIT Press, 2000.
- [27] V. I. Levenshtein, *Binary codes capable of correcting deletions, insertions, and reversals*, in *Soviet physics doklady*, 1966, pp. 707-710.
- [28] S. M. Lundberg and S.-I. Lee, *A Unified Approach to Interpreting Model Predictions*, in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, 2017, pp. 4765-4774.

- [29] L. McInnes, J. Healy, N. Saul and L. Grossberger, *UMAP: Uniform Manifold Approximation and Projection*, *Journal of Open Source Software*, 3 (2018), pp. 861.
- [30] T. Mikolov, K. Chen, G. Corrado and J. Dean, *Efficient Estimation of Word Representations in Vector Space*, in *CoRR*, 2013.
- [31] A. Natekin and A. Knoll, *Gradient boosting machines, a tutorial*, *Frontiers in neurorobotics*, 7 (2013), pp. 21.
- [32] J. Ni, G. H. brego, N. Constant, J. Ma, K. B. Hall, D. Cer and Y. Yang, *Sentence-T5: Scalable Sentence Encoders from Pre-trained Text-to-Text Models*, in *Findings of the Association for Computational Linguistics: ACL 2022*, Association for Computational Linguistics, 2022, pp. 1864-1874.
- [33] J. Pennington, R. Socher and C. D. Manning, *Glove: Global Vectors for Word Representation*, in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014*, ACL, 2014, pp. 1532-1543.
- [34] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee and L. Zettlemoyer, *Deep Contextualized Word Representations*, in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018*, Association for Computational Linguistics, 2018, pp. 2227-2237.
- [35] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li and P. J. Liu, *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*, *Journal of Machine Learning Research*, 21 (2020), pp. 140:1–140:67.
- [36] N. Reimers and I. Gurevych, *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*, 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*, Association for Computational Linguistics, 2019, pp. 3980-3990.
- [37] M. T. Ribeiro, S. Singh and C. Guestrin, "Why Should I Trust You?": *Explaining the Predictions of Any Classifier*, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016*, ACM, 2016, pp. 1135-1144.
- [38] A. Rortais, F. Barrucci, V. Ercolano, J. Linge, A. Christodoulidou, J.-P. Cravedi, R. Garcia-Matas, C. Saegerman and L. Svenjak, *A topic model approach to identify and track emerging risks from beeswax adulteration in the media*, *Food control*, 119 (2021), pp. 107435.
- [39] R. E. Schapire, *The boosting approach to machine learning: An overview*, *Nonlinear estimation and classification* (2003), pp. 149-171.
- [40] A. D. Secker, M. N. Davies, A. A. Freitas, J. Timmis, M. Mendao and D. R. Flower, *An experimental comparison of classification algorithms for hierarchical prediction of protein function*, *Expert Update (Magazine of the British Computer Society's Specialist Group on AI)*, 9 (2007), pp. 17-22.
- [41] C. N. Silla Jr. and A. A. Freitas, *A survey of hierarchical classification across different application domains*, *Data Mining and Knowledge Discovery*, 22 (2011), pp. 31-72.
- [42] J. Snoek, H. Larochelle and R. P. Adams, *Practical Bayesian Optimization of Machine Learning Algorithms*, in *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012.*, 2012, pp. 2960-2968.

- 
- [43] A. Srivastava and C. Sutton, *Autoencoding Variational Inference For Topic Models*, in *5th International Conference on Learning Representations, ICLR 2017*, 2017.
 - [44] R. A. Stein, P. A. Jaques and J. F. Valiati, *An analysis of hierarchical text classification using word embeddings*, *Information Sciences*, 471 (2019), pp. 216-232.
 - [45] S. Terragni and E. Fersini, *OCTIS 2.0: Optimizing and Comparing Topic Models in Italian Is Even Simpler!*, in *Proceedings of the Eighth Italian Conference on Computational Linguistics, CLiC-it 2021*, CEUR-WS.org, 2021.
 - [46] S. Terragni, E. Fersini, B. G. Galuzzi, P. Tropeano and A. Candelieri, *OCTIS : Comparing and Optimizing Topic Models is Simple!*, in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, EACL 2021*, Association for Computational Linguistics, 2021, pp. 263-270.
 - [47] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, *Attention is All you Need*, in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, 2017, pp. 5998-6008.
 - [48] K. Wang, N. Reimers and I. Gurevych, *TSDAE: Using Transformer-based Sequential Denoising Auto-Encoder for Unsupervised Sentence Embedding Learning*, in *Findings of the Association for Computational Linguistics: EMNLP 2021*, Association for Computational Linguistics, 2021, pp. 671-688.
 - [49] K. Wang, N. Thakur, N. Reimers and I. Gurevych, *GPL: Generative Pseudo Labeling for Unsupervised Domain Adaptation of Dense Retrieval*, *CoRR*, abs/2112.07577 (2021).
 - [50] L. Wang, M. Feng, B. Zhou, B. Xiang and S. Mahadevan, *Efficient Hyper-parameter Optimization for NLP Applications*, in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, 2015, pp. 2112-2117.
 - [51] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes and J. Dean, *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*, *CoRR*, abs/1609.08144 (2016).
 - [52] Y. Zhao and G. Karypis, *Criterion functions for document clustering: Experiments and analysis*, 2001.

Abbreviations

BERT	Bidirectional Encoder Representations from Transformers
biLM	Bidirectional Language Model
CBOW	Continuous Bag-of-Words
CTM	Contextualized Topic Model
DAN	Deep Averaging Network
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
ELMo	Embeddings from Language Models
GBM	Gradient Boosting Machine
GPL	Generative Pseudo Labeling
HDBSCAN	Hierarchical Density-Based Spatial Clustering of Applications with Noise
LDA	Latent Dirichlet Allocation
LIME	Local Interpretable Model-agnostic Explanations
LSTM	Long Short-Term Memory
MLM	Masked Language Model
NLI	Natural Language Inference
NLP	Natural Language Processing
NMF	Non-negative Matrix Factorization
NPMI	Normalized Pointwise Mutual Information
OCTIS	Optimizing and Comparing Topic models Is Simple
ProdLDA	Product-of-experts LDA
PV-DBOW	Paragraph Vector-Distributed Bag of Words

Topic modelling and text classification models for applications within EFSA

PV-DM	Paragraph Vector-Distributed Memory
PWI	Probability Weighted amount of Information
SBERT	Sentence-BERT
SHAP	SHapley Additive exPlanations
SimCSE	Simple Contrastive Sentence Embedding
SPECTER	Scientific Paper Embedding using Citation-informed TransformERs
ST5	Sentence-T5
STS	Semantic Textual Similarity
SVM	Support Vector Machine
T5	Text-to-Text Transfer Transformer
TF-IDF	Term Frequency-Inverse Document Frequency
TSDAE	Transformer-based Sequential Denoising Auto-Encoder
UMAP	Uniform Manifold Approximation and Projection
USE	Universal Sentence Encoder
XGBoost	eXtreme Gradient Boosting

Appendix A – List of questions, pillars and subquestions for APRIO classification in case study 1

In total there are 10 questions, 33 unique question-pillar combinations and 155 sub-questions.

Question	Pillar	SubQuestion		
Human or animal RA	EFFECT IDENTIFICATION	Identification of the agent characteristics (e.g. molecule or mixture, production of secondary metabolites, modification introduced in the genes)		
		History of safe use and use patterns		
		Hazard identification - i. Inherent properties of the agent (e.g. toxic effects, toxigenic or allergenic potential, genes of concern, AMR, virulence factors, antibiotics production)		
		Hazard identification - ii. Assessments of relationship between the agent and the adverse effect(s). If needed, definition of the evidence streams (e.g. human, in vivo, in vitro, in silico studies)		
		ADME (it relates to both effect identification and characterisation)		
		MoA (it relates to both effect identification and characterisation)		
		Methods development		
		Uncertainty assessment		
		Identification of characteristics of other agents (e.g., food packaging technology)		
			EFFECT CHARACTERISATION	Dose-response to establish a reference point/point of departure (e.g. NOAEL, LOAEL, BMDL)
Estimate of reference values for humans and animals applying uncertainty factors to the established reference point (e.g. ADI, TDI, acute RFD) or deriving them from human studies (e.g. UL), or adoption of a TTC value				
ADME (it relates to both effect identification and characterisation)				
MoA (it relates to both effect identification and characterisation)				
Methods development				
Uncertainty assessment				
Derivation or estimation of MRL for contaminants in commodities				
EXPOSURE ASSESSMENT				Dietary exposure - Food Consumption
				Dietary exposure - Feed consumption:
				Dietary exposure - Occurrence (prevalence) and/or level of the agent (it includes also viable cells, DNA, toxic metabolites, antibiotics, leachates)
		Non dietary exposure (e.g. inhalation, dermal absorption)		
		Methods development		
		Uncertainty assessment		
		Biomonitoring		

		Estimate of residue level
		Comparison to conventional counterpart (e.g. level of toxicant/constituent) for human or animal impact
		Dietary exposure - Assessment of intake
	CHARACTERISATION OF RISK OR BENEFIT	Risk characterisation - Comparison of exposure with: reference points (e.g. NOAEL, BMDL) using the Margin of exposure or Margin of safety;
		Risk characterisation - Comparison of exposure with: reference values (e.g. ADI, TDI, UL) using the hazard quotient
		Methods development
		Uncertainty assessment
		Qualified presumption of safety (QPS)
		Qualitative risk characterization (not ratio based)
		Determination of equivalent safety and/or benefit with other agents
		Identification of risk control options
		Comparison of the baseline and mitigated risk given risk controls
		Comparison of contaminant level to an MRL
Nutritional assessments	EFFECT IDENTIFICATION	Properties of the nutrient/food constituent/food (all)
		Assessment of relationship between the nutrient/food constituent/food and adverse/beneficial effects (in human studies) (all)
		Uncertainty assessment
		Nutritional composition/analysis
	EFFECT CHARACTERISATION	Dose-response to establish - reference values (DRVs)
		Dose-response to establish - minimal eliciting dose
	EXPOSURE ASSESSMENT	Dietary exposure (DRVs, exemptions from labelling) - Food or Supplement Consumption
		Dietary exposure (DRVs, exemptions from labelling) - Occurrence of the nutrient/food constituent
		Human dietary exposure (intake) assessment - whole food for nutrition
		Comparison to conventional counterpart (e.g. composition, concentration of nutrients, endpoints) for nutrition
	CHARACTERISATION OF RISK OR BENEFIT	Comparison of exposure with - minimal eliciting doses
		Comparison of exposure with - threshold levels (exemptions from labelling)
		Uncertainty assessment

		Determination of equivalent nutritional value
Surveillance	EFFECT IDENTIFICATION	Methods development
	EXPOSURE ASSESSMENT	Monitoring - Prevalence of exceedance of MRL/ML/RPA
		Monitoring - Prevalence of diseases
		Monitoring - Occurrence and/or level of potential hazard in humans, animals and food and feed
		Monitoring - Occurrence of foodborne outbreaks
		For surveillance, the above data are collected for assessing managerial/mitigation measures (see 'assessments of methods')
		Methods development
		Uncertainty assessment
	CHARACTERISATION OF RISK OR BENEFIT	Uncertainty assessment
		Estimation of risk based on surveillance data
Animal welfare	EFFECT IDENTIFICATION	Definition of animal population and system (e.g. on farm, during transport, at slaughter)
		Identification of the hazards (e.g. lack of drinkers, lack of ventilation, rough handling from operators)
		Identification of the welfare consequences (e.g. thirst, thermal stress, pain) including severity
		Assessment of impact on animal welfare - assessment of the relationship between the exposure to a system (and related hazards) and the welfare consequences
		Assessment of impact on animal welfare - assessment of occurrence of welfare consequences in the system and related hazards
		Uncertainty assessment
	EFFECT CHARACTERISATION	Outcome: Identification of the system-related hazards mainly contributing to the welfare consequences in the population
		Application of a scoring system for the severity of individual types of impaired welfare consequences
		Estimation of the duration of the impaired welfare consequences in a given production system
		Estimation of uncertainty in severity and/or duration of impaired welfare
		Uncertainty assessment
	EXPOSURE ASSESSMENT	Assessment of frequency of occurrence of welfare consequences in the system and related hazards
		Estimation of uncertainty in occurrence frequency.

	CHARACTERISATION OF RISK OR BENEFIT	Estimation of an overall score for welfare consequences allowing comparison and risk ranking of production systems.
		Probabilistic uncertainty analysis on overall risk level
		Identification of risk control options
Efficacy	EFFECT IDENTIFICATION	Inherent properties of the agent/measure of which efficacy is assessed
		Assessment of relationship agent/measure-beneficial effects
		Methods development
		Intended Use
		Mode of Action
		History of beneficial use
	EFFECT CHARACTERISATION	Dose-response assessment to evaluate the efficacy of the agent (e.g. to estimate the log-reduction of the pathogen on a food item)
		Methods development
		Evaluation of efficacy using endpoint measurements (e.g. plant height, days to maturity, days to 50% flowering, etc)
	EXPOSURE ASSESSMENT	Estimation of occurrence based on process efficacy
	CHARACTERISATION OF RISK OR BENEFIT	Determination of benefit/efficacy (e.g. resistance to pathogen, reduction to survival/reproductive capacity of pathogen, yield, pest susceptibility) compared to non GM
		Determination of safety based on comparison to reference value
Emerging risks Identification	EFFECT IDENTIFICATION	Inherent properties of the microorganism, chemical substance, drug, additive
		Hazard identification - Identification of a new hazard
		Hazard identification - Identification of new adverse effect of known hazard
	EXPOSURE ASSESSMENT	Assessment of increased exposure of a known hazard in terms of: new susceptibility; new target groups
Identify food vehicle of infection	EFFECT IDENTIFICATION	Step2: Identification of food - pathogen relation and tracing across countries
		Step3: microbiological characterization in the human cases and in the food to identify similarities
	EXPOSURE ASSESSMENT	Step1: Consumption of specific food items in human cases
		Step4: Traceability of the food distribution across food production and consumption chain

	CHARACTERISATION OF RISK OR BENEFIT	Identify vehicle of infection and source of contamination of the food item in the food production process
		Uncertainty assessment
Plant pest / microbial / animal health RA	EFFECT IDENTIFICATION	Methods development
		Uncertainty assessment
		Definition of the characteristics of the biological agent
		Identify potential for spread or MOA for spread
		Factors influencing establishment and spread
		Economic and environmental impact of introduction
		Factors affecting risk control measures
		Identification of the food/vehicle-pathogen relation
		Relationship pathogen-adverse effect(s) (e.g., <i>Listeria monocytogenes</i> , Schmallenberg)
		Dose-response assessment
	EFFECT CHARACTERISATION	Methods development
		History of impact
	EXPOSURE ASSESSMENT	Occurrence (prevalence and concentration) of the pathogen at one or more stages
		Methods development
		Uncertainty assessment
		Potential or known entryways into region
		Presence in the host range
		Consumption of the food (frequency and serving size)
		Probability of exposure
	CHARACTERISATION OF RISK OR BENEFIT	Pest categorisation
		Public, animal and plant pest health impact at baseline OR under alternative risk reduction options/managerial options
		Methods development
		Uncertainty assessment
		Identification of risk control options
		Comparison of the baseline and mitigated risk given risk controls
		Economic and environmental risk characterisation
		Identification of characteristics of the potential stressor
Environment al RA	EFFECT IDENTIFICATION	



		Translation, according to the SC GD (EFSA Scientific Committee, 2016), of the General Protection Goal into Specific Protection Goals (SPGs) - biological entity
		Translation, according to the SC GD (EFSA Scientific Committee, 2016), of the General Protection Goal into Specific Protection Goals (SPGs) - attribute
		Translation, according to the SC GD (EFSA Scientific Committee, 2016), of the General Protection Goal into Specific Protection Goals (SPGs) - magnitude of effect
		Translation, according to the SC GD (EFSA Scientific Committee, 2016), of the General Protection Goal into Specific Protection Goals (SPGs) - temporal and geographical scale of the effect
		Translation, according to the SC GD (EFSA Scientific Committee, 2016), of the General Protection Goal into Specific Protection Goals (SPGs) - tolerable harm
		Hazard identification - Assessment of relationship stressor-adverse effects and factors influencing the relationship (e.g. soil characteristics, wind, rain, temperature)
		Hazard identification - Identification of Pathway to Harm
		Hazard identification - Sequence homology with toxicants (in silico studies)
		Hazard identification - Read across for metabolites
		Identification of pathway to harm/ MoA (e.g. how the protein operates in the target organisms – target gene in case of RNAi that and help understand off-target effect)
		Identification of specific MoA of concern or potential for accumulation
		Methods development
		Environmental risk from modified genetic characteristics
		Environmental risk from newly expressed proteins
	EFFECT CHARACTERIZATION	Concentration –response to establish a reference concentration value (e.g. selecting the NOEC, EC50, LC50, ErC50 to be used in the risk assessment for each group of organisms, Predicted No Effect Concentration)
		Assess activity spectrum of the molecule to identify species that it can affect
		Methods development
		Identification of spatio-temporal controllability (e.g. persistence of a genetically engineered gene drive)
		Heritability and genetic stability of inserted/modified sequence
	EXPOSURE ASSESSMENT	Environmental fate for exposure assessment - Presence, concentration and biological activity of the stressor in the Environment
		Environmental fate for exposure assessment - Predicted environmental concentration (PEC)
		Probability of exposure

		intake/consumption/ingestion/absorption
		Methods development
		Comparison to conventional counterpart (e.g. cross-pollination rate) for environmental impact
	CHARACTERIZATION OF RISK OR BENEFIT	Comparative safety - Risk Quotient (lethal or sublethal dose/environmental exposure)
		Comparative safety - Exposure/reference concentration
		Methods development
		Qualitative risk characterization (not comparison based)
Assessment of methods	EXPOSURE ASSESSMENT	Assessment of sensitivity of detection of disease or conditions

Appendix B – Case Study 4: Example task given to the case expert to evaluate qualitative model performance

The assignment given to the expert consisted of 70 tasks each consisting of two elements. The results of this expert classification are discussed in the main text.

First, the case expert reads three articles, which are clustered together by all NLP algorithms

- **Article 1:**

Preserving ethnic minority musical instruments The musical instruments of Vietnam's ethnic minority groups are on the verge of disappearing. Most of the artisans who make them are old. But some of them are working hard to teach younger generations to make and play the instruments. Artisan Ama H?Loan of Buon Ma Thuot city, Dak Lak province, can make and play the E De people's musical instruments. When H?Loan realized that gongs, trumpets, and flutes had disappeared from local festivals, he began to recreate them, which he learned to do when he was young. He also traveled to many places to study other E De instruments from even older artisans. His musical instruments, made from bamboo, buffalo horns, dry gourds, and beeswax, are used at local festivals and national art performances. H?Loan said ?I had been to many places but didn?t see flutes or the percussion instruments anywhere. In 1999, when I retired, I thought of reviving those instruments because I love them. My neighbors, especially those in their 80s and 90s, were delighted when I played these instruments, which, they said, reminded them of their youth?. Sharing the same passion for ethnic musical instruments, artisan Luong Xuan Nghiep of Con Cuong district, Nghe An province, has set up a Thai minority folk singing club, which now has 40 members. Nghiep spends a great deal of time collecting and preserving gourd lutes (d...n tinh), 2-string fiddles (d...n nh<U+1ECB>), flutes, and wind instruments called ?khŠn?. He wants to transfer his love to young people. Nghiep said ?Young people love modern trendy music, no traditional music. So I set up a club, a venue where they can learn and practice folk songs and folk instruments. The result is quite encouraging. 20 of the club members are children, including my own grandchildren?. Nghiep's club in Cang village and a similar club in Mon Son commune often send their members to performances in Nghe An province and other localities. Both H?Loan and Nghiep are concerned about who, in the future, will preserve and promote ethnic musical

instruments when they are no longer able to do it. H?Loan said ?Ethnic instruments have almost disappeared. I?m now 79. Only a few people know about these instruments. It?s very hard to find younger people who can make and play the instruments proficiently. I?ve proposed that administrators assess the situation carefully?. VOV5.\r\n\r\nPreserving ethnic minority musical instruments, entertainment events, entertainment news, entertainment activities, what?s on, Vietnam culture, Vietnam tradition, vn news, Vietnam beauty, news Vietnam, Vietnam news, Vietnam net news, vietnamnet news, viet.

- **Article 2:**

Sick of traditional fall activities? Here are some offbeat seasonal fests Ah, fall. The time of year for scary walks in the park among actors re-enacting scenes from Shakespeare?s "The Bard." The season of finally learning how to become a home bee-keeper. The time when we all celebrate Oct. 3, the unofficial national holiday celebrating the cult classic movie "Mean Girls." Well, maybe these aren?t the most traditional fall activities, but normal is boring. Event organizers in Western New York are deviating from some regularly scheduled seasonal programming to bring you unique festivals, like a "Mean Girls" viewing party, a kombucha-meets-corn festival or a build-your-own-scarecrow festival. We?ve searched for the more off-the-beaten-path and lesser-known festivals. Here are some we found:\r\n\r\nWhere: Delaware Park. When: Friday, Oct. 12. Tours depart every 15 minutes, from 6:15 to 10:15 p.m. Beware of wandering actors. On a Halloween-themed walking tour in Delaware Park, Shakespeare in the Park actors will portray spooky scenes from "The Bard." Reservations are required and can be made by calling (716) 856-4533. Tickets are \$15 "per victim" and the proceeds benefit Shakespeare in Delaware Park. P.S. The sun sets at about 6:38 p.m. that night, so reserve after 7 p.m. for a darker, creepier setting. Can?t-miss detail: The buy-one-get-one-free drink that comes with your ticket purchase. Make time to redeem your ticket at The Terrace at Delaware Park either before or after your tour.\r\n\r\nWhere: 1 Main St., Wyoming. When: Saturday and Sunday. Begins at 11 a.m. Ends at 5 p.m. Since 1986, residents in this small village about 50 miles from Buffalo host a village-wide fall arts and craft festival. Complete with strolling musicians, apple and pumpkin vendors and a bake-off fundraiser, it?s the classic fall festival that could be a scene in nearly every Hallmark movie shown between August and November. The community bands together, with parking at the fire hall, with juggling and even clogging on Main Street and bluegrass under the village tent. Plus, if pumpkin and apple season is getting overwhelming, Sage Family Maple will have a tent serving bourbon barrel aged maple syrup, cold "mapleccinos," maple coffee and maple-frosted doughnuts. Can?t-miss detail:\r\n\r\nWhere: Masterson?s Garden Center, 725 Olean Road, East Aurora. When: Saturday, Oct. 6 and Sunday, Oct. 7. Events run from 11 a.m. to 3 p.m. If you?ve ever wanted to learn how to do anything bee-related, from beeswax candle-making to backyard beekeeping, local beekeepers will be at Masterson?s Honey Harvest demonstrating how to do exactly that and more. At the sixth annual festival, learn how to render beeswax or how beekeepers harvest honey. Learn about honey?s health benefits. Aside from education, local meaderies and wineries will be offering tastings. East Aurora?s 42 North Brewing even made their own brew for the occasion, "Honey Harvest Brew." Can?t-miss detail: In their annual honey tasting contest, area beekeepers submit their best honey for festival-goers to try. Sample the honey, decide the sweetest batch and vote for the tastiest honey, which will make the winning farm?s queen bee happy.\r\n\r\nWhere: Iron Island Museum, 998 Lovejoy St. When: Friday, Oct. 12. Tours run from 5:30 to 8:30 p.m. Ghosts are scary. Fat Bob?s Smokehouse is not. While you?re eating a barbecue pulled pork sandwich and strolling around

haunted grounds, the two activities are sure to balance each other out. It might seem like a weird combination -- food trucks and haunted house tours -- but it actually makes a lot of sense. What's more comforting after being scared to pieces than diving into a bowl of Sweet Melody's salted caramel flavored gelato? Can't-miss detail: The supposedly haunted museum has been featured on ghost-chasing shows like "GhostHunters" and "My Ghost Story" and used to be a funeral home, which is spooky enough, without the Discovery Channel-confirmed hauntings.

Where: EXPO Market, 617 Main St. #200. When: Monday, from 7 p.m. to 1 a.m. Blimey! When the weather cools and jack-o-lanterns appear, all "Potterheads" (fans of Harry Potter, for all you muggles out there) start celebrating. Normally, that means movie marathons while forcing down weirdly flavored, overpriced Bertie Bott's jelly beans or breaking out your old books. Here is your chance to leave your living room, unearth your black robes from the back of your closet and drink authentic Butterbeer, at the EXPO Market's Wizard Fest. There will be a costume contest, wands for sale, a live DJ and dance party. But sorry, they have a strict no-tolerance policy on bringing your owl, cat or toad. Tickets are \$15 for wizards who are over 18. Can't-miss detail: Quidditch pong, the much more magical version of beer pong.

- **Article 3:**

A mind-changing week with the T?boli The handing down of tradition became evident to us as we listened to the chanting of an eight-year-old T?boli named Ronnie. The chant was about a boy who was advised not to roam on a day when it drizzles and the sun is out. The T?bolis believe that evil spirits are out at this time, and the boy might run into them. Everyone in the audience was clearly mesmerized listening to Ronnie. The T?boli continue to engage in their prime crafts?they are loom weavers, t?nalak weavers and brass makers. By continuing these activities, they preserve the beautiful tradition of the T?boli tribe, and the young maintain their link to their ancestors. In my days there, I visited weavers and brass casters, witnessed the Helobung festival and watched many beautiful performances.

Dream weavers? I learned how weavers encourage their children to start learning as early as age six. T?nalak weaving is a fascinating craft which originated in Lake Sebu. Some t?nalak weavers have very special skills. They are called ?dream weavers.? There are only a few dream weavers or ?Tau Mewel Kena.? Not all weavers are Tau Mewel Kena. The T?boli believe that the ?spirit of abaca,? or ?Fudalu,? chooses a special weaver. All the patterns woven by a Tau Mewel Kena are believed to be given to her in her dreams. The other types of weavers are loom weavers. The Klowil Kem Libun Organization Inc., which was founded by Bernadeth ?Nadeth? Ofong, supports this craft and livelihood. Ofong is a hardworking, forward-thinking T?boli who wanted to preserve their culture, to empower the T?boli women and improve their lives. She is the daughter of a dream weaver. She started a weaving center in 2006, which now has 15 usable looms and 25 malong weavers. The women in her community use the looms to weave cloth to sell in their community and in the region. On weekends and when school is out, the looms are used to teach the T?boli children the skill at an early age. The major issue with weaving is that a woven product is hard to sell in Lake Sebu, as there are not enough tourists coming in, and the more expensive and intricately made weaves have very few buyers among the local folk who can't afford them. There are some foundations, such as Gifts and Graces Fair Trade Foundation, that help create a sustainable market for them. Gifts and Graces works closely with Ofong and her organization to ensure that their beautiful products reach a wider market, and they are paid a fair price.

Brass casting or kem tau temwel is another honored tradition among the T?boli. Brass casters allow their children to handle the equipment only at age 15, a later age than in other jobs, since the equipment is heavy; to handle

liquid brass and fire is very dangerous. Each brass bell is unique as each one comes with a different mold. There are very few brass makers in Lake Sebu since the craft is hard to master. It is a niche market. Brass making is very interesting. First, the brass maker needs to find his raw materials, so he digs the earth for clay, then goes to a metal shop to get his brass and to the forest to collect beeswax. He needs the clay to make sure there are no bubbles. He then forms a bell, using a pattern, with the beeswax. He covers the beeswax with the clay and puts the bell in a pit where it is heated or torched with the use of a blower.

The case expert, then rates the quality of the description given to this topic by the different algorithms.

Rate the following descriptions of the above three articles:

Description 1: plastic, waste, use, food, bags

good mediocre bad

Description 2: plastic, waste, use, food, bags

good mediocre bad

Description 3: festival, beekeeping, beekeepers, Saturday, harvest

good mediocre bad

Description 4: work, said, year, likeness, days

good mediocre bad

The case expert then assesses for a number of new articles, whether they should be added to the original cluster.

Would you add the following articles to the cluster consisting of the above three articles?

- **Article 4:** yes no

Interviews and photos: Lyric Theatre takes 'Junie B. Jones, The Musical' from the page to the stage Kristin Kuns stars as Junie B. Jones in Lyric Theatre's production of 'Junie B. Jones, The Musical.' Photo provided by KO Rinearson. An abbreviated version of this story appears in the Sunday Life section of The Oklahoman. From page to stage: Lyric Theatre working to make the grade with 'Junie B. Jones' musical. Getting into character for her latest role requires Kristin Kuns to don mismatched socks, purple glasses and a reddish-brown wig, plus occasionally snacking on candy for a handy boost. 'It definitely takes a lot of energy, because 6-year-olds just naturally have more energy than the rest of us,' said the 2017 Oklahoma City University graduate with a laugh. 'I'm really excited. If I could have told my 6-year-old self this would happen, I don't think I would have believed myself.' In her Lyric Theatre debut, the local actor is embodying one of her childhood heroes as the star of 'Junie B. Jones, The Musical,' which is already a smash hit ahead of its opening Wednesday morning at the Plaza Theatre. 'Junie B.' is off the charts and it's going to be fabulous,' said

Lyric Theatre Associate Artistic Director Ashley Wells, who is directing and choreographing the show. "What's on the page is so amazing. It really is. It's a very well-crafted show for kids. I get to say, "Hey actors, let's play and have fun." And it's so great. We're having such a wonderful time." Micah Martine plays Herb in Lyric Theatre's production of "Junie B. Jones, The Musical." Photo provided by KO Rinearson. Precocious program. Following 2017's first foray into theater for young audiences with "James and the Giant Peach," Lyric will again focus on children and families with "Junie B. Jones, The Musical," with performances through March 25. A sensory-friendly show designed for children with autism, other sensory-processing disorders or special needs is planned for 11 a.m. March 10, and an American Sign Language interpreted performance is set for 2 p.m. March 17. Due to the demand for tickets including orders from Kansas, Arkansas and Texas as well as from across Oklahoma -- Michael Bratcher, Lyric's audience services and public relations manager, said a 20th performance has been added for 1 p.m. March 22. An adaptation of Barbara Park's best-selling children's books, the musical is aimed at children ages 4 to 10 and their parents, an age group Lyric had not often catered to until launching its theater for young audiences program last year. "I purposefully picked "Junie B. Jones" because last year we did a show that had a little boy as the lead, and I wanted to make sure the next one had a little girl as the lead," said Michael Baron, Lyric's producing artistic director. "It's amazing. I didn't think that initiative of doing theater for young audiences would take off as quickly and as strongly as it has, but it's like gangbusters. There's something special about doing an hour-long (show) for that younger age group. We will, I'm sure, do a theater for young audiences show every year." Kristin Kuns, left, stars as Junie B. Jones and Mahalia Gronigan plays Mother in Lyric Theatre's production of "Junie B. Jones, The Musical." Photo provided by KO Rinearson. Popular title. Created by Marcy Heisler and Zina Goldrich, "Junie B. Jones, The Musical" draws from several books in Parks' long-running series that launched in 1992 with "Junie B. Jones and the Stupid Smelly Bus" and has for more than two decades chronicled the kindergarten and first-grade experiences of the precocious and funny title character. "Everybody that I've talked to about the show it's like, "Oh, I read those books! I read all those books!" Wells said. "To actually see this character that you grew up with and now your kids are growing up with on stage, it's just a treat. "It is a perfect way to introduce theater to a young person, with a character they already know." When she was a girl, Kuns said she and her mom would laugh over Junie B.'s adventures together. "I had almost all the books, and one of my biggest regrets regarding this is when I was 16 I was having a garage sale and I sold them all. I sold them to a teacher, so they went to a good home. But looking back, I'm like, "why did I do that?" because they were a precious childhood memento," Kuns said. "My mom and I would read them together, and for the longest time, she was the voice of all the characters in my head because she would read them to me. So, she has definitely been an inspiration through this in creating Junie B.'s voice." Featuring an array of eclectic and catchy musical numbers, the show is set on Junie B.'s first day of first grade and delves into her search for a new best friend, her experiences in a kickball tournament and the discovery that she needs glasses. "What I love about this show is to see these obstacles that our kids find and have in their lives, and Junie goes about tackling these obstacles in a very real first-grader way and that's good," Wells said. "When she gets her glasses in this, she says it's the worst thing to ever happen to her in her whole life. As adults, we're like, "OK, what?" But no, to her, it is. It's a huge difference in who she is and how people see her. Now that I'm working on the show, when my kid comes home from school and something's crazy and I just want to go, "Really?" I'm trying to realize, "No, wait, in his life right now, this is huge, so maybe I should pay attention.?" Kristin Kuns stars as Junie B. Jones in Lyric Theatre's

production of "Junie B. Jones, The Musical." Photo provided by KO Rinearson. Serious theater. Although the show centers on the exploits of a child who keeps a "Top-Secret Personal Beeswax Journal," that doesn't mean the cast and crew aren't taking Junie B. seriously. "I was fortunate enough to be able to take a children's theater course in college, and I learned a lot about what makes authentic children's theater. It's all about meeting the kids where they're at; kids are human beings, too," Kuns said. "I think we've all seen a kids' show where we're just like, 'Wow, I'm being talked down to.' For kids, it's hard when you lose your best friend or you find out you have to wear glasses. Those are real emotions and real things that kids are going through, and we by no means are making fun of that. We're just trying to tell it honestly." Not only do the actors have to turn in performances that are authentic and true rather than campy and cartoonish, but with only a six-person cast, every performer except Kuns, whose Junie B. never leaves the stage, also has to play multiple roles. "I've got boys playing girls, girls playing boys, and of course, everybody's playing a kid at one point and then they have to turn around and be an adult. And it's going to be a workout for the dressers backstage because some of this is so fast," Wells said. "This show is actually very hard. One of the songs is 'When Life Gives You Lemons,' and it's a ragtime feel. You have a hard-rock feel with the song 'Show and Tell,' and you've got music from the 1950s and '60s with the song 'Lucille, Camille, Chenille' with shoo-be-doo-wops and four-part harmony in different places. But the performers also get to experience the pure joy of revisiting their childhood days, which Kuns said came rushing back to her when she put on the multi-colored and patterned costume Lyric's Jeffrey Meek designed for her turn as Junie B. "I love all types of theater: I love Noel Coward, Oscar Wilde, George Bernard Shaw, Shakespeare. But Junie B. is just fun," Kuns said. "Sometimes it's just nice to be able to get into the body of a character whose job is to play and juggle biscuits and wear glasses, whose job is literally to be silly." ON STAGE. Lyric Theatre's "Junie B. Jones, The Musical" When: Wednesday through March 25. When: Lyric at the Plaza, 1725 NW 16. Tickets and information: or 524-9312. -BAM Related to this story. www.lyrictheatreokc.org Kristin Kuns stars as Junie B. Jones in Lyric Theatre's production of "Junie B. Jones, The Musical." Photo provided by KO Rinearson

- **Article 5:** yes no

Prominent Brexiter James Dyson turns first farming profit Dyson has invested €75 million into his farming and energy business (Photo: Eva Rinaldi/CC BY-SA 2.0) Prominent Brexiter and landowner James Dyson has turned his first farming profit, generating a pre-tax profit of €747,000 in 2017. Dyson's Beeswax Dyson Farming generated an 11 per cent increase in turnover to €15.7m last year, according to Financial Times. The business comprises 35,000 acres of land throughout Lincolnshire, Oxfordshire and Gloucestershire, with the main activity farming and energy. Financial support from EU schemes totalled €2.8m, up from €2.4m the previous year, because of land purchases. The company said: "The subsidy receipt is constant on a per hectare basis from year to year. Our move into profit is a result of the investments that we have been making in our soil health, technology and infrastructure." "Beeswax Dyson is a commercial farming business, and receives the subsidies that any similar business would. These subsidies, along with other voluntary action, have ensured very high levels of environmental stewardship and investment." Dyson has invested €75 million into his business over the past five years, mainly in technology, training, soil improvement and environmental stewardship. Since the EU referendum, Dyson has stated that Britain should leave the EU Single Market and that this would "liberate" the economy and allow Britain to strike its own trade deals around the world. However,


the prominent Brexiter has been critical of the Conservative Government's direction. Last year, Dyson criticised Defra Secretary Michael Gove's 'Green Brexit' approach to the industry after the UK leaves the EU, saying small farmers may be at a disadvantage to their European counterparts under the plans. Copyright , 2018 FARMINGUK. Owned by Agrios Ltd. Managed under license by Red Hen Promotions Ltd - 01484 400666

- **Article 6:** yes no

Can Ty brocade weaving village Can Ty village in Quan Ba district, Ha Giang province, is famous for flax growing and weaving. Villagers don't know exactly when the craft emerged but at present local women with deft hands make brocade dresses and many other products to serve the needs of locals and tourists who want purchase a souvenir. Can Ty village is a small village with only about 270 households. All villagers are Mong ethnic people who mainly live on farming on mountain fields and growing corn, potato, and cassava. At leisure time after harvest, most of them weave brocade. The principle raw material is locally grown flax fibers. Flaxes are planted on the field by strewing them. After being harvested, they are dried and split to yarn for weaving. Linen cloth is manually woven. Traditionally, Mong girls must know sewing before getting married to make their own wedding dresses. When they are still small, Mong girls have been taught to do embroidery, needlework, and weaving. Thus a majority of them excels in weaving and can pass down the craft to the next generation. can ty brocade weaving village hinh 1. Cu Thi My demonstrates brocade weaving at the Vietnam Craft Village Trade Fair 2018 in Hanoi. Cu Thi My, a local craft-woman, said "Years ago, the craft was handed down to the descendants in a hope that the cultural characteristics of the people will not fade away. When the children get married, two sets of clothes must be prepared to give to them. When a Mong person dies, they must have three sets of clothes to wear so that they can meet their ancestors. It's necessary to plough the soil thoroughly to plant flaxes. After harvesting, the trees should be dried carefully and are not exposed to the rain. To have the product soft and beautiful, it's necessary to make fibers small and even." The village authorities have focused on vocational training, creating jobs and developing the local economy. In 2011, the Can Ty Linen Weaving Cooperative was established. Sung Thi M<U+1EF5>, a villager, said "Twelve households have joined the cooperative and engaged in the craft for a long time. The cooperative has provided training for women and girls. The craft requires a skilled hand. Our products include skirts, dresses, bags, blankets, and towels with diverse patterns." It requires a Can Ty weaver to coordinate rhythmically the hands to drive the shuttle, the foot to pedal the treadles to let out heddles, and the back to stretch the fabric. Each patterned brocade cloth is not only a souvenir of a traditional craft village in Ha Giang, but also demonstrates the cultural characteristics of the Mong ethnic people. Cu Thi My, a Can Ty craft-woman, said "The most typical feature of the product is the pattern painted with beeswax. From 2004 on, the craft village has prospered. At the beginning we only weaved for ourselves and later on for tourists who want to buy cloth as souvenirs. I usually deliver products to shops around the Temple of Literature in Hanoi, which sponsors our craft by purchasing all our products." With Quan Ba district's tourism potential, the Mong ethnic people in Can Ty village have more opportunities to develop the brocade weaving and expand the market towards building a brand and increasing the locals' income. This contributes to the preservation and promotion of local ethnic culture. VOV5. Can Ty brocade weaving village, entertainment events, entertainment news, entertainment activities, what's on, Vietnam culture, Vietnam tradition, vn news, Vietnam beauty, news Vietnam, Vietnam news, Vietnam net news, vietnamnet news, vietnamnet bridge.

- **Article 7:** yes no

www.efsa.europa.eu/publications



Exorcists on call to raise French spirits amid surge in private treatment for demonic possession
O ut, out,? chanted Jean Cl,ment, as he performed a ceremony to rid a Parisian businessman of ?negative spirits? supposedly conjured up by a jealous rival. As the chanting went on, Robert ? the ?patient?, as Mr Cl,ment calls him ? appeared to have entered a trance-like state. In his 50s, with purple circles below his puffy eyes so deeply indented that they look like bruises, he said he had been suffering from chronic insomnia and anxiety as his business goes through a bad patch. He is one of a growing number of people in France who are turning to exorcists to save their careers or restore love to troubled relationships. As he prepared for the exorcism, Mr Cl,ment lit candles and incense and placed them beside a crucifix and semi-precious stones on a coffee table in the man?s elegant apartment. The ritual involves candles, incense, and semi precious stones Credit: Magali Delporte/ ,Magali Delporte. N earby he laid out beeswax, nails and a twig from a wild cherry tree, which he says will ward off...