

Faculté	des Langues
	Université de Strasbourg

Master Technologies des Langues

2023-2025

Climate Change Representation in IPCC Reports and Wikipedia: A Comparative Analysis Through Natural Language Processing

Lucas PRÉVOT

Master's Thesis

Under the supervision of
Pablo RUIZ FABO
Senior Lecturer

La représentation du changement climatique dans les rapports du GIEC et sur Wikipédia : Une analyse comparative à l'aide du traitement du langage naturel

Cette étude présente une analyse comparative via traitement automatique du langage naturel des résumés à l'intention des décideurs (SPM) du Groupe de travail III du Groupe d'experts intergouvernemental sur l'évolution du climat (GIEC) et des versions correspondantes de l'article « Atténuation du changement climatique » en anglais sur Wikipédia. En utilisant un large éventail de techniques de TAL, notamment la lexicométrie, la stylométrie, l'évaluation de la lisibilité, l'analyse de la modalité, la similarité sémantique (Sentence BERT), le topic modeling (BERTopic), la détection des sentiments et des émotions, ainsi que la reconnaissance d'entités nommées, l'étude examine comment l'atténuation du changement climatique est représentée dans ces deux sources entre 1990 et 2022. Les résultats montrent que les SPM du GIEC conservent un niveau de technicité élevé, tandis que Wikipédia, initialement plus accessible et centré sur les événements et les personnalités, s'est progressivement aligné, tant sur le plan sémantique que stylistique, avec le GIEC. L'analyse ne révèle aucune preuve de biais délibéré dans la représentation de l'atténuation du changement climatique sur Wikipédia. Les différences de cadrage et de mise en avant s'expliquent plutôt par son rôle de ressource éditée par le public et destinée à un large public. L'étude conclut que Wikipédia joue un rôle important et en constante évolution dans la compréhension des enjeux du changement climatique chez le grand public. Le contenu de Wikipédia est de plus en plus fidèle aux rapports du GIEC.

Mots-clés : Changement climatique, GIEC, Wikipédia, traitement du langage naturel, TAL, analyse, topic modeling, reconnaissance des entités nommées, communication scientifique

Climate Change Representation in IPCC Reports and Wikipedia: A Comparative Analysis Through Natural Language Processing

This thesis presents a comparative Natural Language Processing analysis of the Intergovernmental Panel on Climate Change (IPCC) Working Group III Summaries for Policymakers and corresponding versions of Wikipedia's "Climate Change Mitigation" (CCM) article. Using a comprehensive range of NLP techniques, including lexicometry, stylistic, readability assessments, modality analysis, semantic similarity (Sentence BERT), topic modelling (BERTopic), sentiment and emotion detection, and Named Entity Recognition, the study explores how CCM is portrayed across these two sources between 1990 and 2022. The findings show that IPCC SPMs consistently maintain a high level of technicality and Wikipedia, while initially more accessible and focused on events and personalities, has gradually aligned both semantically and stylistically with the IPCC. The analysis reveals no evidence of deliberate bias in Wikipedia's representation of climate change mitigation. Instead, differences in framing and focus reflect its role as a publicly edited resource intended for a general audience. The study concludes that Wikipedia plays an important and evolving role in supporting public understanding of climate change, increasingly reflecting the IPCC's scientific assessments.

Keywords: Climate Change, IPCC, Wikipedia, Natural Language Processing, NLP, Analysis, Topic Modeling, Named Entity Recognition, Science Communication

Acknowledgements

This master's thesis has felt like a journey. While this document represents the final tangible outcome of my work, the journey truly began in 2023 when I enrolled in Technologies des Langues. From that point onward, I learned so much each day that I feel compelled to express my gratitude for every moment. Each day was a stepping stone towards the completion of this work.

First and foremost, I am deeply grateful to my supervisor, Dr Pablo Ruiz Fabo, for his teachings, dedication and passion for natural language processing. Even though I asked far too many questions in class, he always took the time to answer every single one of them and never once appeared bothered.

I also wish to thank Dr Amalia Todirascu and Dr Delphine Bernhard for the knowledge and support they shared with us. Alongside Dr Pablo Ruiz Fabo, they each played a crucial role in the programme, and the Master Technologies des Langues would not have been the same without them.

It is also important for me to thank everyone else involved, from the individual course lecturers who did an excellent job to the reactive administrative staff, as each person contributed in their own way to this thesis. Even the coffee machine in the Patio's cafeteria deserves a thank you. Thank you, coffee machine.

However, as wonderful as the Université de Strasbourg may be as a place to study, it represents only one half of a much larger picture. What would we be without the support of friends and family?

I'd like to thank my friends, especially my very good friend Antonin, and others who have come and gone from my life, but who, at one point or another, were there for me.

Most importantly, I wish to thank my mother for her unwavering, never-ending support, as well as the people I consider family in Bischwiller and Diebolsheim.

Thank you for believing in me.

Table of Contents

Introduction	1
1 Previous Works	3
1.1 Previous Works on the IPCC.....	3
1.1.1 Presentation	3
1.1.2 Previous Works	4
1.2 Previous Works on Wikipedia.....	6
1.2.1 Presentation	6
1.2.2 Previous Works	7
1.3 Section Summary: Previous Works	7
2 Technical State of the Art	8
2.1 Text Representation.....	8
2.1.1 Encoding	8
2.1.2 Format.....	9
2.1.3 Raw Text.....	9
2.2 Corpus Availability	9
2.2.1 The IPCC Reports	9
2.2.2 The Wikipedia Climate Change Portal	10
2.3 Statistical Analysis of Textual Data	11
2.3.1 Implementing Tokenization.....	12
2.3.2 The Python Programming Language	12
2.3.3 The spaCy Pipeline	13
2.3.4 Text Analysis Platforms	14
2.3.5 What is Textometry?	15
2.3.6 Lexicometry.....	15
2.4 Classification.....	18
2.4.1 Sentiment Analysis and Emotion Detection	19
2.4.2 Tools for Sentiment Analysis	19
2.5 Topic Modelling.....	20
2.5.1 Tools for Topic Modelling	20
2.6 Other Techniques	20
2.6.1 Sentence-BERT and Cosine Similarity Score	20
2.6.2 Soft-cosine Similarity	21
2.6.3 Named Entity Recognition	21
2.7 Section Summary: Technical State of the Art	23
3 Corpus and Methods	24
3.1 Corpus	24
3.1.1 Selected Article Revisions	25
3.1.2 Corpus Size.....	25
3.2 Preprocessing	26
3.2.1 Data Acquisition and Cleaning.....	26
3.2.2 Preprocessing and Linguistic Annotations	29
3.3 Methodology	30
3.3.1 Lexicometry, Stylistic, and Readability Analysis	30
3.3.2 Modality Analysis.....	31
3.3.3 Semantic Similarity Comparison	33
3.3.4 Topic Modelling	34
3.3.5 Sentiment Analysis and Emotion Detection	35
3.3.6 Named Entity Recognition Analysis	37
3.4 Section Summary: Corpus and Methods	39
4 Results and Discussion	40
4.1 Lexicometry, Stylistic, and Readability Analysis	40
4.1.1 Initial TXM Exploration	41

4.1.2	Word Count	42
4.1.3	Lexical Diversity	42
4.1.4	Lexical Density	43
4.1.5	Relative Frequency of Function Words	44
4.1.6	Sentence Count, Length, and Word Length	45
4.1.7	Readability	46
4.1.8	POS Tags Distribution	47
4.1.9	TF-IDF scores	48
4.2	Modality Analysis	50
4.2.1	Verb and Adverbs	50
4.2.2	Likelihood and Confidence	52
4.3	Semantic Similarity Comparison	53
4.4	Topic Modelling	54
4.5	Sentiment and Emotions Analysis	55
4.5.1	Sentiment Analysis with VADER	55
4.5.2	Transformer-based Sentiment Analysis	56
4.5.3	Emotion Detection	56
4.6	Named Entity Recognition Analysis	57
4.6.1	SpaCy's NER Results	57
4.6.2	Climate Change NER Results	58
4.6.3	Results Synthesis	59
4.7	Section Summary: Results and Discussion	59
	Conclusion	60
	References	62
	Appendices	70
	Appendix A: Libraries and Software Versions	71
	Appendix B: Documents, Links and GitHub	72
	Appendix C: [Python code] Extracting Text from a PDF File Using pypdf	74
	Appendix D: [Python code] Text Extraction from Wikipedia	75
	Appendix E: [Python code] Preprocessing with spaCy	76
	Appendix F: [Python code] Lexicometry, Stylistic and Readability Processing	77
	Appendix G: [Table] Top 10 TF-IDF Scores per Document	80
	Appendix H: [Python code] Expression of Modality Processing	81
	Appendix I: [Python code] Semantic Similarity Processing	83
	Appendix J: [Python code] Topic Modelling Processing	85
	Appendix K: [Python code] Sentiment and Emotion Processing	87
	Appendix L: [Python code] Named Entity Recognition Processing	90
	Appendix M: [Table] Complete spaCy NER Results	93
	Appendix N: [Table] Complete Climate Change NER Results	94
	Appendix O: [Python code] Data Visualisation	95

List of Figures

Figure 1: IPCC Structure	3
Figure 2: SpaCy NLP Pipeline	13
Figure 3: Cosine similarity in a 2-dimensional space	21
Figure 4: Earliest Version Available of CC Mitigation (26 th of June 2005)	24
Figure 5: Text size by Number of Words as Computed with TXM's "Dimensions" Tool	41
Figure 6: Evolution of Herdan's C Over Time.....	43
Figure 7: Evolution of Ure's Lexical Density Over Time	44
Figure 8: Relative Frequency of Function Words Over time (reversed Y Axis).....	45
Figure 9: Flesch Reading Ease Over Time	46
Figure 10: Flesch-Kincaid Grade Level Over Time	47
Figure 11: Total Modal Verb Frequency (per 1000) Over Time	50
Figure 12: Total Modal Adverb Frequency (per 1000) Over Time	51

List of Tables

Table 1: spaCy Named Entities Categories Available in Their Models.....	22
Table 2: Selection of Climate Change Mitigation Article Revisions from Wikipedia	25
Table 3: Token Count of the Raw Text WG3 SPMs from AR1 to AR6 According to Gemini..	27
Table 4: IPCC Confidence Terminology	32
Table 5: IPCC Likelihood Terminology	33
Table 6: Document Pairs for Semantic Similarity Comparison	33
Table 7: Climate-Change-NER Categories	39
Table 8: Token Count per Document Obtained with spaCy	42
Table 9: TTR, Herdan's C, Guiraud's R	42
Table 10: Ure's Lexical Density per Document	43
Table 11: Relative Frequency of Function Words per Document	44
Table 12: Sentence Count, Length and Word Length per Document	45
Table 13: Readability Scores Based on the Flesch Reading Ease (FRE, 1948) and its Later Development, the Flesch-Kincaid Grade Level (FKGL, 1975).....	46
Table 14: Distribution of Relevant POS Tags	47
Table 15: Total Modal Verb/Adverb Freq. per Text	50
Table 16: "Likely" Adverb Normalized Frequency per Text	52
Table 17: Total Likelihood and Confidence Frequencies per Text	52
Table 18: Average and Median Document Similarity Scores	53
Table 19: Topic Distribution for Each Text According to BERTopic.....	54
Table 20: VADER Sentiment Analysis Results	55
Table 21: RoBERTa Sentiment Analysis Results.....	56
Table 22: RoBERTa Emotion Detection Results	56
Table 23: spaCy Relevant NER Categories	57
Table 24: Relevant CC NER Categories.....	58

List of Abbreviations

- API: Application Programming Interface
- AR: Assessment Report
- CC: Climate Change
- CCM: Climate Change Mitigation
- CLT: Construal Level Theory
- EPPM: Extended Parallel Process Model
- FKGL: Flesch-Kincaid Grade Level
- FLE: Flesch Reading Ease
- GHG: Greenhouse Gas
- HDP: Hierarchical Dirichlet Process
- IPCC: Intergovernmental Panel on Climate Change
- KWIC: Key Word in Context
- LD: Lexical Density
- LDA: Latent Dirichlet Allocation
- LLM: Large Language Model
- NER: Named Entity Recognition
- NLP: Natural Language Processing
- OCR: Optical Character Recognition
- PDF: Portable Document Format
- POS: Part-of-Speech
- RAG: Retrieval Augmented Generation
- RFFW: Relative Frequency of Function Words
- SBERT: Sentence-BERT
- SPM: Summary for Policymakers
- TF-IDF: Term Frequency-Inverse Document Frequency
- TTR: Type-Token Ratio
- WG: Working Group

Introduction

The Earth, our beautiful planet, is in deep trouble. Centuries of industrial activity have pushed it to the brink of collapse. The climate is changing, causing irreversible damage to ecosystems, including, but not limited to, species extinction. We, as humans, are the primary cause of climate change (CC), and we now risk the extinction of our own species: humankind. Of course, we are not responsible for this extinction as individuals, but as a society. A single person cannot put an end to CC, but a collective effort could influence the policymakers to regulate industrial activity as part of climate change mitigation (CCM).

To document and provide humankind with a state-of-the-art overview of knowledge related to CC, the IPCC (Intergovernmental Panel on Climate Change) was created under the impulse of the United Nations in 1988. In over 30 years, the IPCC has produced six assessment reports (AR), synthesizing the risks, impacts, and mitigation strategies for CC (IPCC, 2024a). Each AR consists of contributions from three working groups, each focusing on different aspects of CC. The ARs are highly technical and complex documents, making them a challenging read for non-specialists. However, they are “... *widely used by policymakers, scientists and other experts* ...” (IPCC & WMO, 1992, p. vii) to take action on CC.

Consequently, despite being published in multiple languages, these works are rarely read by the public, who instead tend to rely on other sources for information on CC. Given that Wikipedia is the 8th most visited website in the world (Similarweb.com, 2025), we can assume that it plays a significant role in informing the general public: Wikipedia hosts an entire portal, or central page, dedicated to CC in multiple languages.

Considering that these two sources are read by distinct audiences yet cover the same topic, studying the differences could yield valuable insights. This thesis aims to compare the IPCC reports and the Wikipedia portal on CC using Natural Language Processing (NLP) tools, with the goal of detecting potential differences or biases in their portrayal of CC. The analysis will focus exclusively on the English-language versions of both sources, as Wikipedia is significantly more developed in English (Wikimedia Foundation, 2025), offering a more comprehensive CC portal for comparison.

The IPCC reports and the Wikipedia CC portal are both extensive resources, a direct comparison between them is not effective due to significant structural differences. Their vastly different formats and content structures mean that information is organized, detailed, and communicated in distinct ways. A direct comparison could lead to content misalignment that could distort or falsify results. Instead, we found it to be more effective when aligning specific Wikipedia articles with corresponding sections of each report, particularly regarding controversial topics.

The thesis is structured as follows: Section 1, “Previous Works”, provides a review of the existing literature. Section 2, “Technical State of the Art”, covers fundamental NLP concepts and tools. Section 3, “Corpus and Methods”, details the corpus and research methodology. Finally, Section 4 “Results and Discussions”, presents and interpret the findings. As an overview of the thesis, the rest of this introduction contains a detailed outline of the content of each of the sections.

Section 1, “Previous Works” reviews the established research relevant to the study. It begins with an overview of the IPCC including analyses of its Assessment Reports (ARs), communication strategies, and the criticisms it has encountered concerning factual accuracy, procedural integrity, epistemic representation, and ontological limitations according to De Pryck & Hulme (2022). It also considers how the IPCC is portrayed in various forms of media.

Subsequently, this section reviews literature on Wikipedia, focusing on its development, quality control mechanisms, and its function as a public source of scientific information, especially in relation to CC. The section also explores Wikipedia's role in NLP research and existing studies examining how it frames complex topics such as CC.

Section 2, “Technical State of the Art”, outlines the theoretical and practical foundations of the NLP methods employed in the research. It introduces essential concepts such as text representation, including encoding and formatting, and describes methods of data acquisition and conversion from sources such as PDF reports from the IPCC and web pages from Wikipedia. Then, the discussion turns to techniques in statistical text analysis, tokenisation, and preprocessing. Alongside text analysis platforms, core programming tools are introduced with Python recognised as the most suitable language for NLP, including key libraries such as spaCy, NLTK, scikit-learn, BERTopic and sentence-transformers. The section concludes with a detailed account of the specific NLP techniques used in the study.

Section 3, “Corpus and Methods”, outlines the study's methodology. It begins by defining the corpus, which includes all IPCC Working Group III Summaries for Policymakers (From ARs 1-6) and different historical revisions of Wikipedia's “Climate Change Mitigation” article. The section then explains the processes of data acquisition and cleaning, involving tools such as pypdf, regular expressions, and Large Language Models (LLMs). It then describes the preprocessing pipeline implemented using the spaCy library, followed by an explanation of how each NLP technique is applied to the corpus. These methods include lexicometric analysis, readability scoring, modality extraction, semantic similarity computation, topic modelling, sentiment and emotion classification, and Named Entity Recognition (NER).

Section 4, “Results and Discussion”, presents the findings of the comparative analyses and provides a comprehensive interpretation of the results. For each NLP technique applied, it systematically compares the IPCC documents with the corresponding Wikipedia articles. The analysis highlights both similarities and differences, examining these patterns in relation to the nature of each source, its intended audience, and the way in which they influence the framing and communication of climate change mitigation. It also identifies areas of convergence and divergence between the two platforms, to assess the extent to which Wikipedia reflects the original source material.

Through this approach, the thesis examines how the critical issue of climate change mitigation is represented on two influential yet contrasting platforms. By applying a range of NLP techniques, the research offers new quantitative insights into differences in framing, emphasis, and potential bias. In doing so, this thesis seeks to deepen our understanding of how scientific knowledge about climate change is communicated and reshaped in the public sphere.

1 Previous Works

1.1 Previous Works on the IPCC

1.1.1 Presentation

The IPCC is a United Nations body responsible for assessing CC science, hence the term “Assessment Report” (AR). Established by the United Nations and the World Meteorological Organization, the IPCC reports draw on contributions from thousands of experts worldwide. These reports are considered the most significant scientific work on CC and serve as the foundation for policymakers in CC negotiations (Masson-Delmotte et al., 2021, p. vii).

The IPCC comprises 3 working groups, each specializing in a specific area of CC (Figure 1).

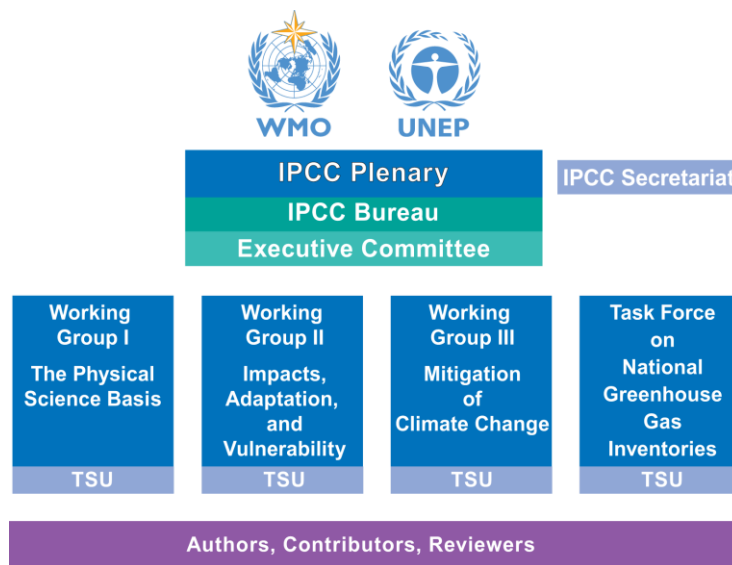


Figure 1: IPCC Structure

Source: archive.ipcc.ch/organization/organization_structure

Working Group 1 (WG1) focuses on the physical science basis of CC (CC), Working Group 2 (WG2) assesses impacts, adaptation, and vulnerability, and Working Group 3 (WG3) addresses CC mitigation. Each WG report includes a technical report and a Summary for Policymakers (SPM). After the working groups complete their individual reports, a final Synthesis Report and a corresponding SPM are created.

The Task Force on National Greenhouse Gas Inventories, added in 1998, focuses on methodology for greenhouse gas (GHG) estimation rather than producing assessment reports (IPCC, 2024b). As the focus of this article is textual content, and the Task Force on National Greenhouse Gas Inventories differs from the regular working groups by not producing reports, it will not be a focus.

The IPCC reports are authoritative, but they are not free from errors or criticism: De Pryck & Hulme (2022, p. 149–151) explain that the IPCC is criticized on four fronts: factual, procedural, epistemic and ontological:

- Factual: Occasionally, IPCC reports contain mistakes, such as the exaggerated claim about the rate of Himalayan glacier melt in 2010. This error attracted media attention and sparked controversy, ultimately prompting the IPCC to revise its procedures.
- Procedural: The IPCC has faced accusations of not strictly adhering to its own procedures, for example, there have been unauthorized changes to the report content

of WG1 in AR2 after final approval. This controversy has led the organization to refine its rules and to add a “Review Editor” role.

- Epistemic: Disputes over the interpretation and representation of data, such as the representation of long-term temperature change with a “hockey-stick graph” in AR3.
- Ontological: Criticism over the lack of integration of indigenous knowledge and the over reliance on quantitative models and natural sciences, which limits epistemic diversity.

The IPCC has historically focused on hypothetical policy evaluations due to the previous lack of actual climate policies. However, after examining the AR6, Tol (2023) showed that it largely ignored empirical evaluations of these now-enacted policies. He also criticized the reports for presenting overly optimistic assumptions that lack political realism, such as the assumption of universal peace. He further criticized the IPCC for not incorporating recent findings in game theory, which suggest that collective international efforts to reduce emissions are unlikely to succeed without global enforcement mechanisms.

The politically charged nature of the IPCC (as an organization that blends politics and science) appears to be the most significant issue, as the reports are influenced by political representatives that may resist acknowledging policy failures or politically inconvenient truths. Tol (2023) advocates for an IPCC that operates with greater academic independence, distancing itself from politically charged institutions that may compromise its task.

1.1.2 Previous Works

There have been numerous works on the IPCC and representation of CC. Barkemeyer et al. (2016) did a linguistic analysis of the SPMs and compared their readability and tone with media coverage. They found, using the Flesch Reading Ease score, that the IPCC SPMs are less readable than the articles in scientific journals and quality newspapers. The SPMs aim for a neutral tone, while media coverage tends to be more pessimistic and sensational (measured with DICTION optimism score, a rule-based analysis that matches a document’s words to specialized dictionaries to compute scores (Hart, 1984)). Tabloid newspapers are generally the most readable, but climate information is presented in a much more emotive and sensational tone.

The article also points out that IPCC report readability sometimes declines after IPCC plenary sessions, as political interests shape the presentation of scientific content. This politicization leads to more complex language. As a result, the article suggests that the IPCC could improve the readability of the SPM by involving professional science communicators in the review process or by providing communication training for the scientists involved.

Biros & Peynaud (2019) also conducted a comparative study on the representation of the IPCC reports in various sources, specifically examining its portrayal in Earth Negotiations Bulletins¹, United Nations reports, and the general press. The ENB and UN reports maintain a highly technical tone, closely mirroring the IPCC’s original language, while the general press simplifies IPCC findings by rephrasing or summarizing to improve readability and appeal to a wider non-specialist audience, omitting technical details found in the ARs. The general press also often recontextualizes, introducing quotations with explanations and interpretations to engage public interest. The general press selectively emphasizes certain aspects of IPCC reports, such as extreme weather events and threats to human life, while

¹ Earth Negotiations Bulletins are objective and comprehensive summaries of ongoing international environmental negotiations.

omitting technical details necessary to accurately assess the probability statements. However, the tone of the general press shifted from scepticism during the 2009-2010 Climategate controversy to a more supportive stance in the 2014-2017 period.

In summary, text aimed at the public often transforms information to enhance accessibility by stripping probabilistic adverbials such as “likely” or “very likely” to provide clearer and more definitive statements, which the general audience tends to prefer. However, this simplification isn’t necessarily negative, as probabilistic adverbials can sometimes diminish the perceived urgency or importance of the message for the public.

The usage of modal adverbials by the IPCC was already criticized by Roeder (2011), who argued that expressing uncertainty through these terms is opaque, and reflects the scientific uncertainties and the political landscape that influences their communication. He suggests that the IPCC could enhance transparency in how uncertainties are quantified and explained, as it could prevent policymakers from underestimating the risks of CC. Herrando-Pérez et al. (2019) also studied modal adverbials and found that the IPCC frequently uses moderate confidence qualifiers, mostly in the 66%-100% (likely) and 0%-33% (“unlikely”) ranges. This cautious wording can lessen the sense of urgency and potentially reducing public motivation to address CC. Given their importance, we compared the usage of modal adverbials in our analysis.

In a content analysis of the IPCC SMP, Poortvliet et al. (2020) used the Extended Parallel Process Model (EPPM)² and the Construal Level Theory (CLT)³ to assess how threat and efficacy (for EPPM) and psychological distance (for CLT) are communicated. They found that the SPM emphasizes information on threats, particularly regarding extreme weather, sea-level rise, and food security risks. However, details on effective actions to mitigate these risks are less common and appear mostly in later sections. They argue that the separation of threat and efficacy in the text could reduce message effectiveness. Applying the CLT to the reports, they found that abstract representations dominate the SPM, potentially diminishing the urgency and personal relevance of climate issues. Concrete messages, which are considered more effective for engagement, are less frequent. They suggest that the future IPCC reports should integrate more concrete examples and closer threat/efficacy association. Comparing the use of concrete and abstract examples in this thesis could provide insight into how effectively Wikipedia and the IPCC communicate climate risk. A quantitative approach to this analysis would be interesting, though its effectiveness would depend on the length of the corpus. For a smaller corpus, a qualitative approach would be more appropriate.

How the IPCC is represented on YouTube was analysed by Bounegru et al. (2020). More specifically, they examined 40 high-ranking YouTube videos about the representation of the *Special Report (SR15): Global Warming of 1.5 °C* (2018). Four themes emerged:

- Disaster and Impacts (The consequences of global warming)
- Policy Options and Solutions
- Political and Ideological Struggles
- Contested Science (Scepticism, criticism of the IPCC projections)

² Witte's (1992) Extended Parallel Process Model explains that individuals respond to fear-arousing messages by first evaluating the threat's severity and their susceptibility to it and then assessing their ability to effectively counter the threat, leading them to either take action or avoid it.

³ Trope & Liberman's (2010) Construal Level Theory of Psychological Distance is a psychological model that explains how the psychological distance of an event, object, or person influences our thinking, specifically whether we perceive it in abstract or concrete terms.

Professional channels primarily dominated the audience, while less-viewed amateur channels often presented more critical and conspiratorial theories that challenged the IPCC's conclusions, despite YouTube's efforts to reduce misinformation.

Finally, Ceylan (2022) explored the application of NLP techniques using the same report as Bounegru et al., (2020), SR15, as a test dataset. He demonstrated the viability of NLP tools to create a question-answering system to improve accessibility over unstructured data, such as the IPCC reports. He used Google's Universal Sentence Encoder and BERT to facilitate semantic search, and an associative MongoDB database was then created to link textual data with figures and tables. A similar question-answering system was also experimented by Vaghefi et al. (2023) with the creation of ChatClimate, a conversational AI powered by OpenAI's GPT-4 large language model (LLM). ChatClimate was developed to address the common issues of hallucinations and outdated data in LLMs regarding CC. By incorporating up-to-date information from AR6 (converted from PDF reports to a JSON format) and instructing the model to prioritize AR6 content, ChatClimate provided more reliable responses than standalone GPT-4, but needed a considerable amount of experiment with hyperparameters (such as K-nearest vectors) to find the optimal balance between accuracy and relevance.

1.2 Previous Works on Wikipedia

1.2.1 Presentation

Wikipedia is a web-based non-profit collaborative encyclopaedia founded in 2001 that allows users to freely create, edit, and share knowledge on a wide range of topics. The articles on Wikipedia are collaborative and voluntary, so they are not signed by individual authors, which poses a fundamental challenge to traditional academic standards.

At first, this openness attracted many volunteers but varying levels of quality in Wikipedia articles (Wöhner & Peters, 2009). Incidents such as the Seigenthaler biography incident—where a defamatory Wikipedia entry was created about journalist John Seigenthaler, falsely claiming he could have been implicated in the Kennedy assassination—led Wikipedia to take drastic measures to ensure the quality of its content (Cohen, 2009), even if it meant losing volunteers (Halfaker et al., 2013). Even though Halfaker et al. (2013) predicted a decline, Wikipedia continued to attract more and more visitors and remained an independent website through annual donation campaigns. As such, it is the most popular non-commercial website, the most popular encyclopaedia, and the 8th most popular website worldwide, according to similarweb.com (May 2025)

This infinite repository of free knowledge makes it the most accessible and prevalent source of information for the public and even sometimes researchers, as each statement in every article must be referenced or will be flagged by the community if unverified and eventually deleted. This community control, combined with automated tools for detecting vandalism ('Wikipedia:Cleaning up vandalism', 2024), mimics peer review and ensures high-quality articles.

However, even modern Wikipedia is not free from conflicts of interest, as companies try to edit Wikipedia for their own benefit (Beutler, 2019). Some controversial articles can be protected from edition, to ensure their integrity ('Wikipedia:Protection policy', 2024). But we could argue that the most important issue with Wikipedia is its lack of language diversity: the majority of the encyclopaedia is in English, and content gaps exist between English, the most represented language, and other languages (Ribé et al., 2021)

In the case of CC, Wikipedia offers a dedicated portal on the subject⁴. Wikipedia portals are thematic pages focused on specific topics, grouping and recommending related articles. For major subjects like CC, Wikipedia articles are generally well-developed and largely based on findings synthesized in the IPCC reports. Since Wikipedia is one of the most visited websites worldwide, the general public likely uses it to stay informed about CC, instead of the IPCC reports. Given that Wikipedia articles provide an overview of a subject by explaining technical terms and aiming for clarity, they serve as a more accessible summary for the general public. This contrasts with the comprehensive and technical nature of the IPCC Assessment Reports, which are intended for a different audience. For these reasons, we find it interesting to compare Wikipedia's CC portal with the IPCC Assessment Reports, to assess if, in addition to the increased accessibility of the content, other differences or potential biases in Wikipedia can be revealed through a content comparison.

1.2.2 Previous Works

As the most popular and accessible knowledge platform, Wikipedia has also become a frequent target for researchers for its interesting dynamics, for example an article from Keegan et al. (2012) analysed how Wikipedia editors collaborate on breaking news articles.

But in the field of NLP, Wikipedia has been used as a major source of training and test data: BERT (Devlin et al., 2019) was trained using large amount of English Wikipedia texts to capture language representations.

Wikipedia2Vec was introduced by Yamada et al. (2020) as a Python-based toolkit for generating embeddings for words and entities from Wikipedia, which helps in tasks such as question answering, text classification and entity linking.

Finally, Korte et al. (2023) studied the representation of CC on Wikipedia. They used sociological systems theory to understand how different social systems process information about CC. It emphasizes that different social systems (such as science or the mass media) perceive and communicate CC based on their internal logics. They analysed Wikipedia as it represents mass media and found that Wikipedia mirrors cyclical attention patterns driven by public events. According to Korte et al. (2023), CC discourse on Wikipedia has evolved from a general discussion (or “chat”) to a “crisis”, influenced by activists' movements and extreme weather events, which is not only characterized by a change of tone, but also a shift in how articles are presented. While science evolves towards solutions-oriented knowledge, Wikipedia frames CC as a public event, driven by political events and personalities. In the view of Korte et al. (2023), both systems contribute to the understanding of CC in the public.

Political events and personalities can be considered named entities from an NLP point of view. Identifying or quantifying named entities is a core component of NLP. As such, this thesis applies Named Entity Recognition (NER) to assess the hypothesis proposed by Korte et al. (2023). The methodology related to this is outlined in Section 3.3.6, and the results of the analysis are presented in Section 4.6.

1.3 Section Summary: Previous Works

Previous studies have examined the IPCC's communication style, report readability, and media representation, as well as Wikipedia's role in framing CC. They highlight how technical IPCC content is often simplified for public audiences. Building on this, the thesis compares IPCC mitigation SPMs with related Wikipedia articles using NLP techniques to quantify differences, offering a nuanced view of how each source communicates climate change.

⁴ Accessible on en.wikipedia.org/wiki/Portal:Climate_change

2 Technical State of the Art

In this section, we review the technologies and tools used to address the various aspects of the research question in this thesis, regarding the comparison between the IPCC Summaries for Policymakers (SPM) on Climate Change Mitigation (CCM) and the corresponding Wikipedia articles.

Section 2.1 focuses on text representation on computer systems, and address encoding (2.1.1) file formats (2.1.2), and the importance of raw text for NLP (2.1.3).

Section 2.2 focuses on corpus retrieval, with each subsection dedicated to a specific source. Section 2.2.1 examines the original PDF format of IPCC reports and the challenges of converting them into usable raw text, while Section 2.2.2 outlines the retrieval and conversion of Wikipedia articles from HTML and API JSON formats.

Section 2.3 covers statistical text analysis, beginning with preprocessing tasks such as tokenisation (2.3.1), followed by the use of Python (2.3.2) and NLP libraries like spaCy (2.3.3). It also introduces text analysis platforms (2.3.4), textometry (2.3.5), and lexicometric techniques (2.3.6).

Section 2.4 presents text classification methods, from bag of words and N-grams to contextual word embeddings such as ELMo and BERT. Section 2.4.1 explores different sentiment analysis and emotion detection approaches, while Section 2.4.2 lists tools such as VADER and transformer-based models trained specifically for these tasks.

Section 2.5 introduces topic modelling techniques, including Latent Dirichlet Allocation and BERTopic, along with their associated tools (2.5.1).

Finally, Section 2.6 covers NLP methods such as Sentence BERT for semantic similarity (2.6.1), and explains the difference with soft cosine similarity (2.6.2). The last section discusses Named Entity Recognition for identifying and categorising named entities. (2.6.3).

2.1 Text Representation

The first step in any NLP task is data acquisition and conversion into a format suitable for analysis. We begin by reviewing the original formats of the reports and Wikipedia content and next, we explore various conversion methods aimed at minimizing information loss. However, before proceeding further, we establish definitions related to text representation.

Text representation is a subset of the broader field that is Natural Language Processing.

Ideally, natural language processing would enable a computer to understand texts or speech and to interact accordingly with human beings. (Nugues, 2024, p. 1)

But for speech to be processed by a computer, it must be in a suitable format. The goal of Optical Character Recognition (OCR) and Speech-recognition which are also part of NLP is the transcription of natural language to a suitable format to be processed by computers. Text representation in a computer is a complex subject; however, the most important concepts are encoding and format.

2.1.1 Encoding

Encoding refers to how characters in a text are represented as bytes in a computer, most notable encoding examples are ANSI, ASCII and the modern UTF-8. Encoding is the reason special characters, like “é” sometimes appear incorrectly on foreign computers or software. Special characters and symbols were not included in early computer releases or varied depending on the location (and language) where the computer was sold. (Spolsky, 2003). So,

encoding refers to the interpretation of text characters by a system, If the system misinterprets the bytes, the result will appear garbled or incorrect. For example, the Windows operating system has used Unicode encoding since Windows 2000, but it wasn't fully adopted until the mid-2000s: In 2008, HTML5 recommended UTF-8 as the default character encoding (Hickson & Hyatt, 2008). In NLP, encoding is not the primary concern since it does not directly relate to the linguistic or semantic content of speech or text. However, it should not be completely disregarded, as some software may lack Unicode support, leading to potential information loss. In the case of this thesis, it will not be an issue and won't be discussed further.

2.1.2 Format

Format refers to the structure or arrangement of data. While encoding applies specifically to characters and symbols, format is a much broader concept. The simplest format for text is what we call "raw text" which is text as we write and read it, without any formatting. Raw text is typically stored in .txt files and lacks any structure, which can make it harder to read. Web browsers, on the other hand, use the HTML format, a markup language composed of raw text, HTML tags and metadata that displays information on a screen and facilitates navigation. Although raw text can be regarded as a "clean slate" in terms of formatting which is already high in informational content, additional tags and metadata can enhance context by indicating titles, paragraphs, sections, colours and other elements depending on the format, but are not always relevant for NLP, and can obstruct text processing software.

2.1.3 Raw Text

In text processing, metadata, literally data about data (Merriam-Webster, 2024) can refer to elements like Part-of-Speech (POS) tags in a tokenized text, sentiment labels, or even grammar rules (Lebart & Salem, 1994). However, in formats like HTML or PDF, metadata often consists of technical information for software that is largely irrelevant for NLP tasks. While the colour of a page and the format in which text is displayed can be crucial for interpretation and are often designated by tags or metadata, one could argue that this is less relevant when analysing technical documents, such as IPCC reports.

Therefore, the ideal format to begin the analysis is simple raw text, as it is compatible with NLP tools and free from any additional data that could affect the results. Of course, raw text will be pre-processed and tokenized for certain tasks, but the first step should be to acquire the IPCC reports and Wikipedia articles in a clean, metadata-free raw text format.

This subsection reviews the original format of the reports and articles we intend to analyse and compare, detailing the steps required to convert them to raw text.

2.2 Corpus Availability

This subsection reviews the original format of the reports and articles we intend to analyse and compare, detailing the steps required to convert them to raw text.

2.2.1 The IPCC Reports

The IPCC reports are distributed on the IPCC website⁵ as PDF. The Portable Document Format (PDF) is a widely used document type compatible with most browsers. PDFs are known for their unique features and versatility, as they retain their original formatting across different platforms. They can include text, images, tables, hyperlinks, and most media types,

⁵ Accessible on www.ipcc.ch

but the primary goal of a PDF is to preserve document layout, not to allow edition. To achieve this goal, PDFs are generated by specialized software, such as Microsoft Word or LaTeX, and store information as different types of “objects”, making it challenging to edit or extract raw text directly, as the text is not stored in a continuous, flowing format but rather as individually positioned elements (Adobe®, 2006).

The main challenge with automatic text extraction from PDFs is accurately separating body text from non-body elements, such as headings and footnotes. To extract text from the IPCC reports, one could manually copy and paste each title and paragraph into a .txt file. However, this approach is highly time-consuming and impractical given the length of the IPCC reports.

Most of the reports (AR1 to AR5) are already available as raw text due to previous works such as the conversion from Biros et al (2021). In the case of this thesis, we already are in possession of those texts thanks to our supervisor, Dr Pablo Ruiz Fabo, who previously worked with NLP on the IPCC reports. As such, only the conversion of AR6 will be necessary, which is a less time-consuming task if done manually.

As will be detailed in the related methodology subsection (3.2.1), the solution we decided to go with in order to obtain raw text for the IPCC reports is the Python library pypdf (Fenniak & Martin, 2024), which allows users to edit PDFs and extract text. While it supports basic text extraction, its distinctive feature is a “visitor” function that can filter out headers and footers based on their position (y-coordinates). In our case, we chose to use pypdf for basic text extraction and then applied a LLM to filter out headers, footers, and other unwanted content. This technique is detailed in Section 3.2.1

2.2.2 The Wikipedia Climate Change Portal

The acquisition of textual data from online sources such as Wikipedia, which is typically presented in HTML format, requires conversion into raw text for use in NLP applications. Various methods can be employed for this task, each with its own set of considerations.

One common technique is web scraping, which involves automatically downloading and parsing HTML pages. Libraries such as BeautifulSoup⁶, commonly used in Python, are employed to remove boilerplate content including metadata, special tags, navigation menus, and other redundant elements. This process helps isolating raw text for an analysis. Although effective, large-scale scraping can place a considerable burden on web servers.

To reduce server load and support more structured data retrieval, many platforms, including Wikipedia, offer Application Programming Interfaces (API). The Wikimedia API enables the download of articles in different format.

When acquiring a large number of articles, complete database dumps, distributed by Wikipedia, are convenient and encouraged. However, for obtaining a smaller and more specific set of articles, APIs are generally more appropriate. Access to the Wikipedia API can vary, it allows a limited number of unauthenticated requests, but more extensive use typically requires account registration and the use of an API key. Wikipedia also provides enterprise-level services, offering higher request quotas and additional functionalities.

Interaction with APIs and extraction of text can be carried out using a range of tools and programming languages. Command-line utilities such as cURL are commonly used, particularly in Linux environments, where they are often pre-installed on Ubuntu distributions (Canonical Ltd, 2024), but they are also available for other operating systems, including Windows. Scripting languages such as Python, PHP, and JavaScript are widely employed for

⁶ BeautifulSoup, developed by Leonard Richardson, is available on pypi.org/project/beautifulsoup4/

interacting with APIs. Python is particularly popular within the NLP community due to its simplicity and the availability of comprehensive libraries. (Nugues, 2024, p. 9)

When data is retrieved via an API, it is typically delivered in a structured format, most often JSON. On Wikipedia, main textual content is found within the “extract” field in the JSON response from Wikipedia’s API. Standard libraries within the chosen programming language, such as Python’s “requests”⁷ library for handling HTTP requests and its effectively named “json” library for JSON parsing, can be used to access and process the data. The resulting text is then converted into plain text and saved as a .txt file.

Using these techniques allows for the conversion of web-based corpora into raw text, making them suitable for further analysis with NLP.

2.3 Statistical Analysis of Textual Data

This section provides an overview of various aspects involved in the statistical analysis of textual data relevant for this thesis. To guide the reader through the complexity of the subject, we will begin by establishing a foundational understanding, summarising key principles and historical perspectives. This initial exploration covers crucial concepts including text segmentation, unit forms, and the significance of preprocessing (2.3.1). Building on these fundamentals, we then examine key programming environments and tools — such as Python — that are widely employed to carry out these analytical tasks (2.3.2 and 2.3.3). The section subsequently introduces Text Analysis Platforms, which offer user-friendly tools for working with large text corpora (2.3.4). Finally, we describe research domains such as Textometry (2.3.5) and related methods like Lexicometry (2.3.6), outlining their objectives and applications.

The Statistical Analysis of Textual Data is a broad field of research that has been developed since the 1950s. The subjects that significantly contributed to its development are Linguistics, Mathematics, Statistics, and Computer Science. Over time, the methodologies have been refined, and the applications have been enriched with new proposals from the segmentation of texts to the development of linguistic resources, the creation of lexicons, concordance analysis, text classification, sentiment analysis. The fields of application are the most varied, ranging from Psychology to Sociology, Marketing, Economics, Medicine, and Politics (Iezzi et al., 2020)

The fundamentals of textual data analysis are summarised and expanded upon “*Statistique Textuelle*” by Lebart & Salem (1994). According to them, the statistical method, by definition, relies on measurements made from objects we count and combine. This means we must consider these segmentation objects as equal in relevance but distinct in use. In the context of text segmentation, units such as words should be segmented appropriately based on the analysis objectives.

As Lebart & Salem (1994) point out, a chemist might want to neutralize linguistic plurality by treating “acid” and “acids” as equivalent, capturing both forms under a single search through lemmatization. Conversely, a linguist might prefer to preserve and distinguish these variations, as the morphological differences carry significant meaning in linguistic studies.

To ensure accurate representation, we must consider the appropriate segmentation level and unit form for our analysis. Segmentation levels range from words, sentences, paragraphs, and even larger structures like sections or thematic units, each providing unique insights.

⁷ The Requests library, created by Kenneth Reitz, is available on pypi.org/project/requests/.

Breaking text down into smaller units is known as tokenization: this process creates what we usually refer to as tokens, which can be words, multiple words, subwords, or even individual characters. Tokens can retain positional information if required.

We could segment the text by words, treating each word as a segment. In this format, the words become unordered, which is commonly referred to as a Bag-of-Words approach. If needed, we could tokenize each word segment into smaller tokens, such as morphemes (subwords) or graphemes (individual characters).

Tokenization is part of the preprocessing phase in NLP, other common preprocessing tasks are, but not limited to:

- Stop words removal (common words like “the”, “and”, “to” that create noise in the analysis)
- Lowercasing (standardising text to one case)
- Special character removal (such as punctuation, or replacement of characters like “é”, carriage return and new line characters (also known as CRLF)).

In addition to tokenization, the unit form we choose influences the analysis. Graphical forms treat each word exactly as it appears, maintaining morphological detail, while lemmatized forms group words by root meaning. Semantic forms groups words that share the same meaning, even if they are lexically different (Lebart & Salem, 1994). Semantic forms are often based on a controlled vocabulary like a thesaurus.

Tokenization for pre-trained language models and LLMs often differs from classical, word-based approaches, typically employing sub-word tokenisation algorithms such as WordPiece (Wu et al., 2016) where the units are statistically derived from the training corpus (Devlin et al., 2019).

2.3.1 Implementing Tokenization

In practice, tokenization can be either straightforward or challenging, depending on the objective. For example, one could use Python’s “.split()” function, which divides a string (sequence of characters) into multiple substrings at each whitespace. However, this approach does not handle punctuation or special cases like “I’m” which represents “I am” and is technically two words but will be counted as a single token if split only on whitespaces.

To tokenize, we employ much more complex “tokenizer” software that solves the tokenising problem for us. Advanced and customisable tokenizers already exist, such as those provided by the NLTK or spaCy libraries in Python.

While developing our own tokenizer is possible, it would be a highly complex undertaking, potentially requiring several weeks of work, and would constitute a separate project beyond the scope of this thesis. Therefore, it is more efficient to use the previously mentioned tools instead.

2.3.2 The Python Programming Language

Python is a high-level, interpreted programming language created by Guido van Rossum and first released in 1991.⁸ Python was designed with an emphasis on code readability and simplicity, (T. Peters, 2024) making it an accessible language for beginners. Code readability also benefits experienced programmers, as they often need to read and understand each other's code on larger projects.

⁸ According to Python’s “History and Licence” documentation available on docs.python.org/3/license.

As an interpreted language, Python is compiled at runtime by an interpreter rather than a compiler, allowing for immediate code execution without the need for a separate compilation step to convert handwritten code into a binary file which is the case for the C programming language, for example (Python Software Foundation, 2024).

This means code is stored as plain text in “.py” files, making it easy to share and read. However, this approach incurs a significant performance cost for large applications (Ziogas et al., 2021), which is why extensive desktop applications or video games are typically not written in Python. Of course, machine learning applications are among the most resource-intensive, and large language models (LLMs) cost millions to operate, yet they are still primarily run using Python.

In these cases, Python serves as a scripting language, while its libraries for data analysis and machine learning such as NumPy, pandas, scikit-learn, TensorFlow, PyTorch and others (Nagpal & Gabrani, 2019) are transmitted to faster languages backends like C, C++ or CUDA to optimize performance. For example, CuPy (Tokui, 2024) is a Python library that enables Python to employ graphical processing units (also known as GPU) for NumPy and SciPy.

Python was not always the obvious choice for NLP. Perl was once preferred due to its “*rich regular expression and a support for Unicode.*” (Nugues, 2024, p. 9). Python eventually adopted these features as well and now hosts most of the major libraries for NLP, along with support for machine and deep learning, which is a very solid environment to study natural languages.

As an additional note, the R language is also a strong contender for data analysis, with libraries such as tm (text mining) and quanteda for NLP, or Stylo for stylometry. However, Python seems to have gained greater importance as neural networks and the transformer model emerged.⁹

2.3.3 The spaCy Pipeline

In spaCy, a Python library for NLP, the pipeline refers to an ordered sequence of NLP tasks. While it includes pre-processing steps (labelled as “tokenizer”), it consists of both pre-processing and processing tasks.

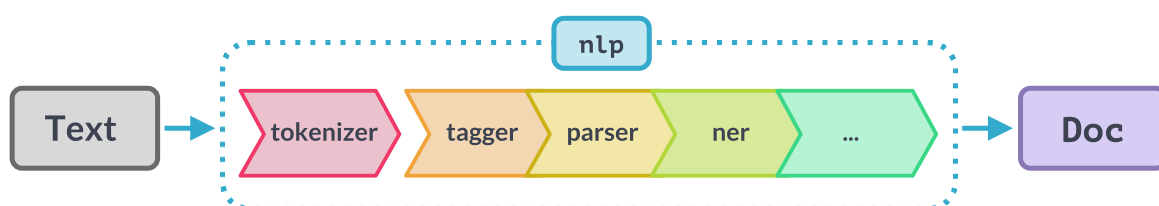


Figure 2: SpaCy NLP Pipeline

Source: spacy.io/usage/processing-pipelines

The processing pipeline can be customized as desired; the one depicted in Figure 2 includes a part-of-speech tagger, a dependency parser, and a named entity recognizer. In the case of our analysis, once the texts have been processed, metrics on NER, part-of-speech tagging, and other tasks can be evaluated and compared.

⁹ Hyperlinks to the libraries referenced in this section: [NumPy](#), [pandas](#), [scikit-learn](#), [TensorFlow](#), [PyTorch](#), [CuPy](#), [SciPy](#), [quanteda](#), [Stylo](#).

2.3.4 Text Analysis Platforms

While spaCy is a powerful tool, it is not ideal for some research-oriented tasks and exploratory visualisation. Text analysis platforms have been developed specifically to enable the exploration of linguistic patterns across corpora. For this reason, we cover such platforms in this section, and we tested the contribution of one of them to the research question in the thesis (see 4.1.1 for the results).

Numerous software tools have been developed to facilitate the statistical analysis of text. Some are available as desktop applications while others are accessible online using a web browser. Pincemin (2018), proposes a list of software tools for conducting such analyses. I explored some of them and added others to create the list below:

Developed by Université Nice Côte d'Azur, **Hyperbase**¹⁰ was developed with a focus on French-language, but most of the features (word frequency, collocations, KWIC, concordance) are language-agnostic and English corpora can be used.

Previous versions of Hyperbase were desktop applications, while the latest version is accessible online through a browser. Previous versions had more features as Hyperbase web as *"In 2015, Hyperbase was released in a web version with a new interface that retained the main features of the original version"* (Hyperbase, 2024). Unfortunately, the previous versions appear to be inaccessible.

IRaMuTeQ¹¹ is an R and Python based software tool for performing multidimensional text analyses. It primarily analyses corpus themes and offers visualization of thematic classes according to the "Reinert classification method", a statistical text analysis technique designed to identify and classify the main themes present in a corpus by grouping text segments such as sentences or paragraphs according to the co-occurrence of words within them (Reinert, 1983). IRaMuTeQ is available in multiple languages and can analyse English texts but only if they are properly preprocessed. Fortunately, IRaMuTeQ allows imports from TXM, which can assist with preprocessing.

Le Trameur (Fleury & Zimina, 2014) is a tool designed for the analysis of structured and annotated texts. It segments texts into "Trame", which refers to a sequence of positions within the text, and "Cadre", representing the partitions or frames that structure the text. The software supports regular expressions and specializes in analysing aligned corpora, including co-occurrence and concordance analyses, with elements of correspondence analysis and visualisation.

TXM (Heiden et al., 2010) is an open-source project primarily developed by the ICAR research lab in France. It aims to provide a comprehensive platform for corpus analysis, offering tools for both quantitative and qualitative text analysis, including statistical analysis, visualisation, and textual exploration. Users can perform frequency analyses, identify co-occurrences, cluster words, and map thematic patterns within a corpus. It allows for text and of corpora from various formats, including raw text, and offers customisation and scripting options for advanced users.

AntConc (Anthony, 2005) can be essentially seen as a beginner-friendly version of TXM, as it covers TXM's basic functions. However, Antconc is unrelated to TXM and was released in 2002, which is almost 10 years earlier. It handles plain text, supports regular expressions, and can perform essentially the same tasks as TXM (minus some visualisation techniques) while offering a comprehensive interface and ease of use.

¹⁰ Accessible on hyperbase.unice.fr

¹¹ Accessible on www.iramuteq.org

Sketch Engine (Kilgarriff et al., 2014) is one of the most accessible platform for text analysis. It is an online platform accessible via any browser and provides textometry tools as well as pre-made corpora. Users can import their own texts and expand their corpus through web scraping performed by the platform itself. Sketch Engine offers additional features, such as Word Sketch, which automatically generates a summary of a word's collocations, and a Thesaurus, however, it is a paid tool. The advantage Sketch Engine has over other platforms is its accessibility and the exceptional user experience it offers for beginners. In my case, it seems unnecessary.

2.3.5 What is Textometry?

According to Pincemin & Heiden (2008), Textometry was essentially developed in France during the 1970, it “*applies a wide range of linguistically significant and mathematically based calculations for methodical and renewed analysis of text collections.*”. Textometry borrows techniques from multiple fields and applies them to linguistic content to uncover information, identify patterns, and reveal relationships within texts. Textometry can be a broad subject, typically including sentiment analysis, stylometry, or topic modelling. However, platforms like TXM focus solely on lexical analysis. We will start by reviewing the lexical analysis techniques.

2.3.6 Lexicometry

Lexicometry is the quantitative study of vocabulary within a text. It measures linguistic features through statistical methods, with the usual measurements as follows:

Word Frequency list: calculates the frequency and rank of each word in a text or corpus.

Type-Token Ratio (TTR): TTR is shown in Equation 1. It measures lexical richness by dividing the number of unique words V (Types) by the total number of words N (Tokens) in a

$$\text{TTR} = \frac{V}{N}$$

Equation 1: Type-Token Ratio

Source: Lissón & Ballier (2018, p. 9)

text. The higher the TTR, the greater the lexical diversity.

However, TTR has a known limitation: it decreases as text length increases. Variations of TTR that are less sensitive to text length do exist, but according to Lissón & Ballier (2018), the extent to which they effectively neutralize the influence of text length remains unclear.

Additional measures can be employed such as **Herdan's C**, which is less sensitive to changes in length. Herdan's C is shown in Equation 2:

$$C = \frac{\log V}{\log N}$$

Equation 2: Herdan's C

Source: Lissón & Ballier (2018, p. 9)

Guiraud's R is useful for comparing lexical richness across texts of different lengths as it accounts for text length by normalizing the count of unique words against the square root of the total number of words. Guiraud's R is shown in Equation 3:

$$R = \frac{V}{\sqrt{N}}$$

Equation 3: Guiraud's R

Source: Lissón & Ballier (2018, p. 9)

Yule's K focuses on repetition a high score indicates high level of repetition, which suggest lower lexical diversity and vocabulary (Lissón & Ballier, 2018). Yule's K is available in Equation 4:

$$K = 10^4 \times \left[-\frac{1}{N} + \sum_{i=1}^v f_v(i, N) \left(\frac{i}{N} \right)^2 \right]$$

Equation 4: Yule's K

Source: quanteda.io/reference/textstat_lexdiv

N is the total number of words.

v is the total number of distinct frequency classes

i is a word frequency class

10^4 is a scaling factor.

The **Lexical Density (LD)** is the proportion of lexical words or content words (opposite of function words) divided by the total words in a text. Higher LD indicates that information is more densely packed, suggesting greater textual complexity. While not a perfect measure on its own, academic texts demonstrate higher LD. Equation 5 is Ure's (1971) LD:

$$LD = \frac{N_{LEX}}{N_{TOTAL}}$$

Equation 5: Lexical Density

Source: Ure (1971, p. 445)

While Ure focuses on a word-level ratio, there is a different approach to lexical density: Equation 6 is Halliday's (1989) clause based approach:

$$LD = \frac{L}{C}$$

Equation 6: Lexical Density (clause based)

Source: Halliday (1989)

L represents the number of lexical items (nouns, lexical verbs, adjectives, adverbs).

C refers to the number of clauses. To clarify, a clause is a group of words that contains at least a subject and a verb.

Distribution of Word Classes: An examination of the distribution of word classes (nouns, verbs, adjectives, adverbs) reveals that different genres exhibit distinct distributions. For instance, scientific texts, such as IPCC reports, may contain a higher proportion of nouns.

Lexical Dispersion: An examination of the spread or distribution of words across a text. Patterns of distribution may carry stylistic or thematic significance and are typically visualized using a dispersion plot.

Cluster Analysis: Grouping of similar words, phrases, or documents based on previously mentioned linguistic features.

N-gram Analysis: Analysis of sequences of “n” words (where “n” is an unknown) that appear together in a text.

Collocation and Co-occurrence: Analysis of words that frequently appear together. Collocation is usually larger, while co-occurrence focus on smaller groups such as pairs.

Key Word in Context (KWIC): A technique that displays a keyword within its immediate textual environment, showing a predefined number of words on either side. This allows for the examination of the keyword’s context. Specialized software typically allows users to click on the keyword to navigate to the actual text.

Readability levels: Readability measures the ease with which a reader can comprehend a written text. In our context, assessing readability is crucial for quantifying expected differences in linguistic complexity. Typically, readability formulae provide a score based on characteristics that can be quantified, such as:

- Sentence Length, as longer sentences are assumed to be harder to process syntactically.
- Word Complexity, as longer words are often associated with lower frequency and greater difficulty.

However, readability doesn’t assess conceptual difficulty or ambiguity.

The most commonly known formula is the Flesch Reading Ease (FRE), introduced by Flesch (1948). It scores readability from 0 to 100, and higher scores indicates easier readability. Simply put, the formula is a weighted average of sentence length and average syllables per word, combined and converted into a score.

The Flesch-Kincaid Grade Level (FKGL) is an evolution of Flesch’s work, developed for the U.S. Navy by Kincaid et al. (1975). Like the previous iteration, it uses average syllable and sentence lengths, but with updated weightings and a different scoring system that results in a U.S. school grade level instead of a score out of 100. The FKGL is still in use today, for example in legal texts. (Han et al., 2024)

Stylistic Markers: Linguistic features that contribute to the unique style of a text. Common features that can be calculated are:

- average sentence length
- average word length
- frequency of function words
- Usage of specific grammatical structures

These features can contribute to the stylometric analysis of a text, for tasks such as identifying the author.

Visualization Techniques: A visual representation of the linguistic features of a text. Certain linguistic features are more easily identified in a 2D space. Different representations exist for different measurements:

- Word cloud: (frequency)
- Heatmap: (co-occurrence patterns)
- Dispersion plots (lexical spread)
- Histogram (frequency distribution)

2.4 Classification

Classification is the process of assigning predefined categories or labels, such as genre, to data. In NLP, this data can range from words to entire texts. It is regarded as a supervised machine learning technique, as it relies on labelled data to train a model that can assign appropriate labels to future, unseen data segments.

For this thesis, classification does not appear to be particularly effective, as the focus is on comparing texts, a task unrelated to classification. However, sentiment analysis, which is a classification task, offers an interesting approach for comparison. Analysing the sentiment of sentences, paragraphs, and entire texts could provide valuable insights.

The factors that a classification model uses to make labelling decisions are known as features. These features are extracted from the segments and may include statistical properties leveraged from lexicometry, such as word frequency. The following are the most common features utilized in classification:

Word Frequency (Bag of Words): Simply counts the occurrences of each word in the text, ignoring grammar and word order.

N-grams: Considered a more extensive version of the bag-of-words approach, N-grams are used to capture local context. By adjusting the value of N, we can control the amount of context captured. A single-word N-gram is referred to as a unigram, two words form a bigram, and three words make a trigram. N-grams are particularly important for sentiment analysis, as they preserve context. For instance, a sentence containing the words “not” and “happy” might be processed separately, leading to an entirely different sentiment interpretation without contextual information.

Term Frequency-Inverse Document Frequency (TF-IDF): balances term frequency within a document against its uniqueness across documents to assign weights to words based on their distinctiveness. It typically eliminates stop words, provided they have not already been removed during preprocessing, as well as domain-specific terms like “climate” in the case of IPCC reports. While “climate” is undeniably significant, its frequent occurrence makes it an expected result, whereas TF-IDF is designed to allow more distinctive patterns to emerge.

Part-of-Speech Tags: Identifies grammatical roles of the words for a more semantic understanding. This way, nuances in style and argumentation can be uncovered.

Syntactic Patterns: Measures sentence complexity using parse trees. It identifies the sentence structure (simple, compound, complex) and count the average number of clauses. More complex sentences suggest a more sophisticated style, which is typically the case with technical, academic documents.

Word Embeddings (Word2Vec, GloVe): Words are represented as word embeddings: each word is represented as a vector in hundreds of dimensions. Words with similar meaning are closer in the vector space (Mikolov et al., 2013) (Pennington et al., 2014). The issue with this technique is that it fails to take context into account, which has limitations for sentiment analysis. This is fixed by contextual embeddings, such as ELMo or BERT.

Contextualized Word Embeddings (ELMo): A more contextualized version of word embeddings. ELMo embeddings are dynamic, and change based on the surrounding context of the word in a sentence (M. E. Peters et al., 2018). This is especially useful for sentiment analysis, machine translation or NER.

Transformer-Based Representations (BERT): While ELMo relies on recurrent neural networks, BERT uses the transformer architecture (Vaswani et al., 2017). BERT outperforms ELMo by capturing a richer context through self-attention mechanisms, while also being more cost-efficient and fine-tunable (Devlin et al., 2019). This has led to the development of multiple models inspired by BERT, such as RoBERTa, ALBERT, and BioBERT (designed for biomedical text).

2.4.1 Sentiment Analysis and Emotion Detection

According to Liu's (2012) work, polarity-based sentiment analysis is a classification technique that categorises segments as positive, negative, or neutral. To classify, a trained model utilizes data about the segments, applying the techniques previously mentioned. The performance of such classification depends on the techniques employed, with transformer-based embeddings being the most effective due to their ability to capture extensive context.

There are variations to this technique: fine-grained sentiment analysis provides a more granular assessment of sentiment, for example on a scale of 1 to 5.

Aspect-based sentiment analysis focuses on determining sentiment polarity towards specific features within a text, such as the polarity of sentiments regarding an object or person.

Intent-based sentiment analysis focuses on identifying the user's intent alongside their sentiment.

Finally, emotion detection aims to identify specific emotions within a text, extending the 3 classes of polarity-based sentiment analysis. Emotions, such as "happiness", "sadness", "anger", "fear", "surprise" can be assigned to segments. (Nandwani & Verma, 2021)

2.4.2 Tools for Sentiment Analysis

The Python library NLTK includes a rule-based model known as VADER, which is designed for sentiment analysis of social media text (Hutto & Gilbert, 2014). Although intended for short segments, it can also be applied to larger texts. While VADER serves as a solid starting point, it relies solely on rules and considers minimal context.

TextBlob (Loria, 2013/2024) is another option for Python, which performs sentiment analysis using a Naive Bayes classifier to generate sentiment scores ranging from -1 to 1.

However, for more accurate results, specialized libraries that utilize word embeddings should be used in addition to VADER. Numerous fine-tuned models based on BERT, tailored for various types of sentiment analysis, are available on Hugging Face and can be integrated into Python using the transformers library.

2.5 Topic Modelling

Topic modelling is an unsupervised machine learning technique used to identify and extract hidden topics from text by clustering words that frequently appear together, facilitating the understanding of the text's underlying structure.

There are several approaches to topic modelling. One common statistical method is Latent Dirichlet Allocation (LDA), which requires specifying the number of topics in advance (Härdle & Chen, 2016). Alternatively, other techniques such as the Hierarchical Dirichlet Process (HDP) do not require this parameter to be set manually, which is useful in cases where the optimal number of topics is unknown (Teh et al., 2006).

More recent methods make use of word embeddings, with one of the latest being BERTopic, a transformer-based approach to topic modelling that also does not require the number of topics to be set manually.

2.5.1 Tools for Topic Modelling

Gensim is an open-source Python library designed for topic modelling and document similarity analysis. Gensim supports LDA and HDP, but also implements word embeddings algorithms such as Word2Vec or Doc2Vec that can be used for topic modelling (Řehůřek, 2024). BERTopic is also a Python library designed for topic modelling but leveraging transformer-based embeddings instead (Grootendorst, 2022), which have been argued to lead to better results than earlier methods due to the semantic capacity of these embeddings.

2.6 Other Techniques

2.6.1 Sentence-BERT and Cosine Similarity Score

Sentence-BERT, sometimes abbreviated as SBERT, is a variation of BERT designed to create fixed-size vector embeddings for entire sentences, allowing for comparison between sentences using a metric named cosine similarity score. SBERT can be expanded over entire paragraphs, rather than individual sentences. (Reimers & Gurevych, 2019)

The cosine similarity score is a method for calculating the similarity between two vectors in a vector space model, introduced in NLP by Salton, Wong, and Yang (1975). When converted to embeddings, words, sentences, or any transformed entity occupy a position in a multidimensional space based on vector coordinates. This space can be 2D, 3D, or N-dimensional, sometimes reaching hundreds or even thousands of dimensions in NLP.

Cosine similarity between two vectors is then measured by calculating the angle between them. In the context of embeddings, a smaller angle indicates that the vectors are more closely aligned, revealing a higher degree of semantic similarity. (Figure 3)

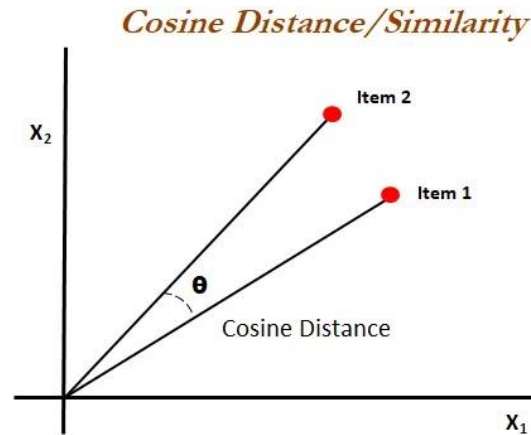


Figure 3: Cosine similarity in a 2-dimensional space

Source: [Medium.com 'How to leverage cosine similarity' by Thao Quach](#)

Thus, cosine similarity can be calculated using *Equation 7*:

$$\cos(\theta) = \frac{A \cdot B}{\|A\| \cdot \|B\|}$$

Equation 7: Cosine similarity

Source: Salton, Wong, and Yang (1975)

The dot product measures the alignment of vectors A and B , while the denominator normalizes the vectors by dividing by the product of their magnitudes. Normalisation is necessary to remove the effect of vector sizes. This is relevant in NLP, as the lengths of the texts whose vectors we compare may differ, leading to higher frequencies in longer texts, whereas this difference is not relevant for semantic comparison.

The result is a score between -1 and 1:

- A score close to 1 indicates a small angle between the vectors, suggesting similarity.
- A score close to 0 indicates a 90° angle, suggesting no similarity.
- A score close to -1 indicates that the vectors point in opposite directions, suggesting they are dissimilar.

2.6.2 Soft-cosine Similarity

Now less commonly used, soft-cosine similarity (Sidorov et al., 2014) was once a strong contender to cosine similarity, as it could encode semantic similarity when provided with word vectors and a term-similarity matrix. Modern techniques, however, rely on powerful transformer-based embeddings that inherently capture semantic relationships and perform very well with cosine similarity, reducing the need for a soft-cosine approach.

2.6.3 Named Entity Recognition

As we mentioned earlier, Korte et al. (2023) have shown that Wikipedia is most likely to rely on political events and public personalities to frame CC. In NLP, these fall under the category of named entities: specific terms typically referring to people, organisations, places, or events, though the scope is not limited to these. The classification and categorisation of such entities vary depending on the thematic focus or context of the corpus.

While Wikipedia tends to rely more on public figures and events, the IPCC is also likely to use a wide range of named entities, though these could be more often associated with specialized CC terminology. As such, extracting named entities with a CC perspective from our corpus enables a comparative analysis of their types and usage across Wikipedia and the IPCC. As an example, English models for spaCy NER (Explosion, 2025), based on OntoNotes 5.0 (Weischedel, Ralph et al., 2013) use the categories detailed in *Table 1*:

Category	Definition (“example”)
CARDINAL	Numerical values (“one”, “2”, “one million”)
DATE	Dates or periods (“July 14”, “last week”).
EVENT	Named events (“Fashion Week”, “World War II”)
FAC	Facilities such as buildings, roads or bridges (“Charles de Gaulle Airport”)
GPE	Geopolitical entities such as countries or cities (“Belgium”, “Marseille”)
LANGUAGE	Natural languages (“English”, “Sanskrit”)
LAW	Documents related to laws (“Constitution”, “Magna Carta”)
LOC	Locations that aren’t GPE (“Mount Everest”, “Atlantic Ocean”)
MONEY	Monetary values with units (“20€”, “£10”)
NORP	Nationalities, religious/political groups (“French”, “Buddhist”, “Socialist”)
ORDINAL	Ordinal numbers (“First”, “2 nd ”, “third”)
ORG	Organizations (“Microsoft”, “Mercedes-Benz”, “Dassault”)
PERCENT	Percentage values (“42%”, “forty-two per cent”)
PERSON	Person or fictional character (“Barack Obama”, “Luke Skywalker”)
PRODUCT	Named manufactured products (“iPhone”, “Zotero”, “Renault Twingo”)
QUANTITY	Unit based measurements (“10km”, “50 kilograms”)
TIME	Specific times and small periods (“01:00 pm”, “midnight”, “morning”)
WORK_OF_ART	Titles of creative works (“Mona Lisa”, “2001: A Space Odyssey”)

Table 1: spaCy Named Entities Categories Available in Their Models.

An issue with NER is that it involves inherent ambiguities in classification. For instance, consider Stanley Kubrick’s film “Paths of Glory”. How would this title be classified within such a system? It would most likely be classified as “WORK_OF_ART”, as Kubrick is considered an auteur filmmaker (Kobaliani, 2023). However, how might one classify a “blockbuster” film? Would it be categorised as “WORK_OF_ART” or “PRODUCT”? Furthermore, the

classification of video games presents a complex case. Although they are undoubtedly commercial products, the significant artistic contribution should also be acknowledged.

In our specific context, which focuses on CC, the NER classification task is expected to present relatively few ambiguities. However, the precision offered by spaCy's categories may be insufficient for a clear comparison. Therefore, the current classification system could be enhanced with subcategories specifically tailored to our corpus. For example, as ORG is a particularly large category in the context of climate change, it could be subdivided into the following subcategories:

- org.NGO (Non-governmental organization),
- org.IGO (Intergovernmental organization)
- org.CMP (Company)

To incorporate such subcategories automatically into a NER analysis, we would need either a spaCy or BERT model specifically trained on climate change content, or a climate change annotated NER dataset for fine-tuning a model. We used such a model (Duran, 2024), and discussed the methodology in Section 3.3.6, while the results are available in Section 4.6.

2.7 Section Summary: Technical State of the Art

This section, Technical State of the Art, has outlined the essential NLP technologies and concepts for this thesis. From there, we examined a comprehensive set of analytical methods, including lexicometry, topic modelling, sentiment analysis, emotion classification, and named entity recognition. These approaches encompass both classical techniques and modern embedding-based methods, enabling the assessment of semantic alignment and the enhancement of analytical precision. Having defined the capabilities of NLP tools, we will, in the next section, Corpus and Methods, describe their concrete implementation in our analysis, bridging the gap between theory and practice.

3 Corpus and Methods

3.1 Corpus

The main objective of this thesis is to compare the IPCC reports to the Wikipedia CC portal to uncover potential differences. The volume of the IPCC reports and the Wikipedia climate change portal is immense; comparing them directly in their raw form would be illogical. As the IPCC reports are primarily intended for policymakers and researchers, whereas Wikipedia targets the general public, it could be insightful to compare them on the highly politicised aspect of climate change mitigation, to examine how political influences from nations shape the content and framing of the IPCC reports in comparison to the collective influence of the general public on Wikipedia. By analysing these discrepancies, this thesis aims to uncover potential divergences between the perspectives and priorities of state actors and the public, shedding light on how political interests may influence the representation of climate change mitigation strategies across these two platforms.

Proposed Methodology: Since Working Group III of the IPCC focuses on climate change mitigation, this study compares the SPM from each WG3 AR with selected revisions of the Wikipedia page “Climate Change Mitigation” that correspond to the same time period. As detailed in the next section (3.1.1) these Wikipedia revisions were carefully chosen based on their proximity to the publication dates of the respective IPCC reports.

Also, by anchoring the analysis to key publication dates, we enable an additional layer of comparison that tracks and examines the evolving discourse on climate change mitigation across both sources over time.

Unfortunately, the earliest archived version of Wikipedia’s Climate Change Mitigation (CCM) article dates back to 2005 (Figure 4), which limits direct comparisons with the first two IPCC Assessment Reports (AR1 and AR2 were published in 1990 and 1995).

• (cur prev) ○	23:39, 26 June 2005	Rd232 (talk contribs) m .. (9,778 bytes) (+4) ..
• (cur prev) ○	23:38, 26 June 2005	Rd232 (talk contribs) .. (9,774 bytes) (+38) ..
• (cur prev) ○	23:37, 26 June 2005	Rd232 (talk contribs) .. (9,736 bytes) (+212) ..
• (cur prev) ○	23:33, 26 June 2005	Rd232 (talk contribs) .. (9,524 bytes) (+1,089) ..
• (cur prev) ○	23:25, 26 June 2005	Rd232 (talk contribs) .. (8,435 bytes) (+187) ..
• (cur prev) ○	23:24, 26 June 2005	Rd232 (talk contribs) .. (8,248 bytes) (+125) ..
• (cur prev) ○	23:23, 26 June 2005	Rd232 (talk contribs) .. (8,123 bytes) (+626) ..
• (cur prev) ○	23:17, 26 June 2005	Rd232 (talk contribs) .. (7,497 bytes) (+217) ..
• (cur prev) ○	23:10, 26 June 2005	Rd232 (talk contribs) .. (7,280 bytes) (+108) ..
• (cur prev) ○	22:54, 26 June 2005	Rd232 (talk contribs) .. (7,172 bytes) (+388) ..
• (cur prev) ○	20:23, 26 June 2005	Rd232 (talk contribs) m .. (6,784 bytes) (+5) ..
• (cur prev) ○	20:23, 26 June 2005	Rd232 (talk contribs) .. (6,779 bytes) (−390) ..
		<i>version here after all) (undo)</i>
• (cur prev) ○	16:40, 26 June 2005	Rd232 (talk contribs) m .. (7,169 bytes) (−2) ..
• (cur prev) ○	15:11, 26 June 2005	Rd232 (talk contribs) m .. (7,171 bytes) (−17) ..
• (cur prev) ○	15:10, 26 June 2005	Rd232 (talk contribs) .. (7,188 bytes) (+7,188) ..

Figure 4: Earliest Version Available of CC Mitigation (26th of June 2005)

Source: en.wikipedia.org/w/index.php?title=Climate_change_mitigation&action=history

Nevertheless, the AR1 and AR2 WG3 SPMs have still been analysed to trace the entire historical evolution of climate mitigation discourse within the IPCC reports themselves, which provided notable findings and additional context to draw upon in this comparison.

3.1.1 Selected Article Revisions

Given that Wikipedia articles are often edited multiple times per day, it is necessary to identify a specific, appropriate revision to compare with each AR.

Typically, we searched for revisions published after the release of an AR, including its findings. However, this was not possible for the 2005 revision, for which we simply used a version published on the day of the article was created.

Following this methodology, we found revisions may relate to IPCC findings, presented in Table 2. Links to their online versions on Wikipedia are available in Appendix B.

<i>Wikipedia article</i>	<i>Corresponding AR</i>	<i>Exact date chosen</i>	<i>Reasoning</i>
<i>CC Mitigation (2005)</i>	<i>AR1 (1990), AR2 (1995), AR3 (2001)</i>	<i>26 June 2005, 23:39 (UTC)</i>	<i>Late at night on the day of the article's creation to ensure the author had finished editing</i>
<i>CC Mitigation (2007)</i>	<i>AR4 (2007)</i>	<i>7 May 2008, 10:15 (UTC)</i>	<i>Addition of a citation from AR4 to define global warming, confirming that AR4 influenced the article's content.</i>
<i>CC Mitigation (2014)</i>	<i>AR5 (2014)</i>	<i>2 September 2014, 07:44 (UTC)</i>	<i>Major revision which added multiple references to AR5</i>
<i>CC Mitigation (2022)</i>	<i>AR6 (2022)</i>	<i>13 June 2022, 22:16 (UTC)</i>	<i>While AR6 WG3 was referenced in the article on the day of its release, the 13 June revision added a substantial amount of information from AR6.</i>

Table 2: Selection of Climate Change Mitigation Article Revisions from Wikipedia

3.1.2 Corpus Size

The proposed corpus, consisting of the six WG3 SPMs and the corresponding versions of the CCM Wikipedia articles (when applicable), offers a focused dataset that facilitates comparison between the two entities and over time.

While this selection enables precise comparison on the specific topic of climate change mitigation, it yields a corpus that may be too limited in size for certain NLP applications, such as topic modelling. Latent Dirichlet Allocation (LDA), for instance, is a topic modelling technique that performs optimally on large datasets (Härdle & Chen, 2016), as such, it wouldn't work in this case.

Although this limitation may seem like an oversight, it is in fact a deliberate methodological choice as it offers the following advantages:

- **Controlled comparison:** This selection ensures a meaningful comparison between functionally analogous documents, aimed at different audiences across consistent time periods.
- **Quality and depth:** It allows for a more detailed qualitative and quantitative analysis of linguistic features, framing and semantic nuances.
- **Practicality:** It maintains a manageable scope suitable for a master's thesis, allowing sufficient time for potentially time-consuming processing steps, such as the Named Entity Recognition sub-categorisation (detailed in Section 2.6.3)

Therefore, while we acknowledge certain limitations associated with specific techniques, the defined corpus is considered sufficient and appropriate for the primary aims of this thesis. The selection and interpretation of NLP methods has taken the corpus size into account and emphasis has been placed on analyses that are robust to medium-sized corpora, including detailed lexicometric comparisons, readability assessments, Sentence-BERT based similarity measures, and NER.

For topic modelling, approaches such as BERTopic were prioritised over traditional methods, as pre-trained models generally perform more effectively than LDA on smaller datasets. The topic modelling methodology is detailed in Section 3.3.4, with results presented in Section 4.4.

3.2 Preprocessing

The following sections (3.2 and 3.3) are structured to provide a comprehensive overview of the analytical pipeline, progressing from data acquisition to the execution of specific textual analyses.

The Preprocessing section includes the following subsections:

- Section 3.2.1, entitled Data Acquisition and Cleaning, describes the procedures employed to compile and standardise textual data from both IPCC reports and Wikipedia, thereby ensuring a comparable corpus.
- Section 3.2.2, Preprocessing and Linguistic Annotations, which details the utilisation of the spaCy library for essential NLP tasks, including tokenisation, lemmatisation, and part-of-speech tagging.

All subsections are presented with an explicitly stated “Objective”, followed by a thorough account of the tools and methodologies applied.

Software and library versions are outlined in Appendix A.

3.2.1 Data Acquisition and Cleaning

Objective: To obtain clean, comparable raw text from both sources.

As previously mentioned, a total of 10 documents (6 for IPCC, 4 for Wikipedia) had to be collected and converted into raw text format to ensure compatibility with an NLP preprocessing pipeline.

For the IPCC WG3 SPMs, while versions in raw text form existed online, such as the version from Biros et al. (2021), their edition did not fully meet my requirements.

Specifically, we needed the WG3 SPM from each AR, and this document was not always included for each AR in their compilation. We were already in possession of five out of the six reports converted to raw text, thanks to my supervisor, Dr Pablo Ruiz Fabo, who had previously worked with NLP on the IPCC. However, the most recent report, AR6, released in

2022–2023, has no raw text conversion available online. To convert the report, we first downloaded the original PDF document from the official IPCC website, we then carried out the conversion using Python.

The safest way to work with Python is to create a new virtual environment for each project, to minimise conflicts between dependencies and avoid unpredictable behaviour. While various software solutions offer this functionality, we found Python’s Anaconda distribution to be the most practical. We worked with Python 3.11.11, as using the latest or an older version is not always the best choice in programming, due to compatibility and security considerations. Python 3.11.11 seemed to be the most balanced choice in this regard.

Using Python 3.11.11, we began by installing and using pypdf to extract raw text from the report. This approach was straightforward, and saving a new file with the extracted raw text required only 15 lines of code (Python code for PDF extraction is available in Appendix C). However, the issue with this method was that pypdf extracted all the text without cleaning it: many elements that could be considered unnecessary remained, such as metadata, authors, headers, footers, and figure captions. While several solutions exist to clean the file, one possibility in this case was to use a state-of-the-art LLM, due to the file’s length.

LLMs can be exceptionally effective at a wide range of repetitive tasks, provided they are supplied with examples and a clear prompt. This approach is known as *few-shot learning* or *few-shot prompting*. In this instance, we already had examples: the cleaned SPMs from AR1 to AR5. Therefore, all that was needed was a well-crafted prompt and an LLM with a very large context window.

As of April 2025, Google has released *Gemini 2.5 Pro Preview 03-25* on their AI Studio. Google’s AI Studio is a platform for developers to test and deploy software solutions using Google AI models. Notably, Gemini supports an exceptionally large context window of 1,048,576 tokens. Token counts may vary depending on the task and model. In our case, token count for the raw text SPMs, as reported by Google’s AI Studio, are listed in Table 3:

<i>AR1 WG3 SPM</i>	<i>21,024 tokens</i>
<i>AR2 WG3 SPM</i>	<i>12,555 tokens</i>
<i>AR3 WG3 SPM</i>	<i>12,603 tokens</i>
<i>AR4 WG3 SPM</i>	<i>18,595 tokens</i>
<i>AR5 WG3 SPM</i>	<i>23,461 tokens</i>
<i>AR6 WG3 SPM (uncleaned)</i>	<i>65,508 tokens</i>

Table 3: Token Count of the Raw Text WG3 SPMs from AR1 to AR6 According to Gemini 2.5 Pro Preview 03-25

All reports totalled 153,746 tokens, meaning Gemini’s context window was more than large enough to support a few-shot learning approach.

The conversion methodology with the LLM was the following:

- The SPMs from AR1 to AR5 were sent to Gemini to establish examples context.
- A custom prompt was then submitted, including the raw text of AR6, instructing Gemini to return a cleaned version in the same style as the previously provided files.

Thankfully, Gemini supports an output limit of 65,536 tokens, which was sufficient to return the entire cleaned document in a single response. This minimised the need for output segmentation and reduced the risk of truncation or context loss. Therefore, *Prompt 1* was submitted to Gemini:

You are an NLP computer scientist.

The previously sent texts comprise the summaries for policymakers from Working Group III in Assessment Reports 1 to 5. They are formatted as raw text, without metadata, tables, headers and footers, document references, unnecessary line breaks, as these elements are not useful for NLP tasks. However, keep figures and tables captions. I am now sending you the Summary for Policymakers from the Sixth Assessment Report as a raw text file. Please clean the file in the same manner as the previous ones, as it has already been converted to text but still contains unnecessary information for NLP. Return the cleaned text.

Prompt 1: Cleaning AR6 with Gemini 2.5 Pro

After carefully reading the results, this approach returned a very satisfactory outcome. The text had been cleaned of unnecessary elements, and a few minor fixes had to be applied, in order to resolve formatting issues (spaces before commas and full stops, consecutive spaces, unnecessary line breaks). Although we could have proceeded to the next step, several additional cleaning procedures were applied across all SPMs. These included the manual removal of special characters (such as bullets “•”), the merging of hyphenated words that had been split across lines (e.g. “in- tellectual” in AR1–AR4), and the insertion of spaces between words in a few cases where they were missing (notably in AR4). We also addressed words fragmented by individual characters (e.g. “S T R A T E G I E S”), consolidating them into proper form (“STRATEGIES”) using *Regular Expression 1*:

$\backslash\text{b}[a-zA-Z](?:[a-zA-Z])+\backslash\text{b}$

Regular Expression 1: Detecting Fragmented Words in Raw Text

While this was a time-consuming manual task, the small size of our corpus demanded the highest possible quality to ensure robust results. For larger corpora, recent advancements in the information retrieval field have led to the development of several tools that are highly effective at processing PDFs and preparing them for NLP applications. By combining multiple processing techniques, such as basic text extraction, OCR and layout identification, Docling (Auer et al., 2024) is also able to extract text from all types of PDFs along with images, tables, and charts. While pypdf and Gemini were sufficient for our task, Docling could be a valuable tool for future work, offering faster document conversion and the ability to extract information from data that was previously inaccessible to NLP.

For Wikipedia articles, we used the requests library and Python code provided in Appendix D. We called Wikipedia’s API and retrieved the articles in JSON format and extracted the content to save it as raw text. However, the output included Wikipedia’s markup language, which is not compatible with NLP applications and had to be cleaned.

Fortunately, the Python library mwparserfromhell¹² allowed us to parse and remove Wikipedia’s markup, cleaning a large part of the files before saving them.

¹² Mwparserfromhell, developed by Ben Kurtovic, is available on pypi.org/project/mwparserfromhell/.

After a careful evaluation of the files, we still identified a few elements that needed to be removed, such as excessive line breaks and special formatting of figure captions. Fortunately, most captions were easy to detect, as they always contained the word “thumb” and one of several “|” symbols, which can be automatically identified using *Regular Expression 2*:

$$(.+?)\|$$

Regular Expression 2: Detecting Wikipedia Captions in Raw Text.

However, we noticed later a few of these captions were not completely removed by that step, though this did not noticeably affect the analysis.

There were also URLs that were manually removed. While we could have used regular expressions to locate them quickly, only a few appeared in the 2014 and 2022 versions of the report, and they were easy to remove manually.

Finally, the earliest revisions contained some irrelevant words typical of Wikipedia, such as “See also:”, which were trivial to identify and remove.

In the next section, we will discuss the preprocessing steps of those documents, a crucial step in NLP applications.

3.2.2 Preprocessing and Linguistic Annotations

Objective: To prepare the text for deeper analysis by performing fundamental NLP tasks.

Raw text documents will be processed using the spaCy Python library, employing its large English model: `en_core_web_lg`

This Python library is multifunctional and will automatically perform the following tasks:

- Sentence segmentation
- Tokenization
- Lemmatization
- Part-of-Speech tagging (POS tagging)
- Dependency parsing
- Named Entity Recognition

This process is non-destructive, spaCy is designed to handle all annotations and stores the results in a Doc object in Python. Specific components, such as NER, can then be extracted from this Doc object and either used as they are or further processed.

Using spaCy as an initial step to generate multiple processed versions of the document was highly efficient and required very little code. Although we could have used different software for each task, such as tokenization, lemmatization, NER and so on, spaCy proved significantly faster, and its output was immediately usable by other Python components.

Preprocessing with spaCy could still be time-consuming if it had to be repeated across multiple texts each time a script or notebook was run. Ideally, each text would be processed once, and the resulting Doc object saved for later use.

Fortunately, spaCy provides a dedicated class, DocBin, which allows Doc objects to be saved to disk in a binary format, helping to avoid reprocessing large corpora. Technically, multiple Doc objects can be stored in a single DocBin object and saved to disk. This means the entire preprocessed corpus can be stored in a single file. However, due to an unknown issue encountered when handling a large DocBin object, we chose to store a single Doc per

DocBin object instead, resulting in one file per document on disk. This did not affect the analysis in any way.

Code demonstrating how the documents were preprocessed and how the resulting DocBin objects were saved to disk is available in Appendix E.

However, spaCy preprocessing was not always sufficient for certain tasks that required tailored treatment to produce meaningful results. For example, while lexicometric analysis typically involved stop-word removal for tasks such as calculating TF-IDF, sentence similarity using SBERT required the original form of the text to be preserved.

For this reason, any additional preprocessing steps will be addressed at the beginning of each respective section under 3.3 below (Methodology), where we will specify whether spaCy's output was used as a starting point and outline the further steps taken.

3.3 Methodology

Methodology subsections (3.3.1 to 3.3.6) are each dedicated to a specific analytical approach. The analysis encompasses the following domains:

- Lexicometric analysis and readability (3.3.1)
- Modality analysis (3.3.2)
- Semantic similarity assessment (3.3.3)
- Topic modelling (3.3.4)
- Sentiment and emotion analysis (3.3.5)
- Named entity recognition (3.3.6)

3.3.1 Lexicometry, Stylistic, and Readability Analysis

Objective: To quantify and compare fundamental textual characteristics such as vocabulary richness or readability.

To get an initial view of the lexical features of our corpus, we began by **using TXM**, an open-source platform designed for corpus analysis. As we mentioned in Section 2.3.4, TXM includes a variety of tools for statistical analysis, visualisation, and detailed textual exploration. We used the latest TXM version, 8.4, released in February 2025. Although TXM supports the import of corpora in a variety of formats and allows for the addition of metadata, we imported only unprocessed raw text into the software. This was due to the limited metadata available and the fact that text exploration using TXM was not the primary focus of our analysis.

- The TXM exploration results are available in Section 4.1.1.

After this initial exploration with TXM, we used the texts preprocessed with spaCy to begin the analysis with **a word count for each text**, corresponding to a simple token count. To achieve this, we imported the saved Doc objects from disk using spaCy's DocBin class for each document and counted the total number of tokens generated, including both the raw count and a count excluding stop words and punctuation marks.

- Word counts results are available in Section 4.1.2.

A different approach was used for **assessing lexical diversity**, as TTR, Herdan's C, and Guiraud's R were computed using lowercased and lemmatized tokens. Lemmatizing and lowercasing improve the reliability of vocabulary richness measures compared to using raw text.

- Lexical diversity results are available in Section 4.1.3.

For Ure's (1971) **lexical density**, we used spaCy's POS tags to identify nouns, proper nouns, verbs, adjectives, adverbs, and numbers, and divided them by the total number of tokens (excluding punctuation). Numbers are considered lexical items because they function as content words by conveying quantitative information that can be very relevant for technical documents.

- Lexical density results are available in Section 4.1.4.

Next, we calculated the **Relative Frequency of Function Words** (RFFW). RFFW can be viewed as a reversed form of lexical density. Technical texts typically have a lower RFFW, which corresponds to a higher lexical density. RFFW was calculated by grouping function words using spaCy's POS tags to identify them and dividing their total by the overall word count (excluding punctuation). We considered the following POS tags as function words: "ADP" (adposition, which are prepositions and postpositions), "AUX" (auxiliaries), "CCONJ" (coordinating conjunction), "SCONJ" (subordinating conjunction), "DET" (determiner), "PRON" (pronoun), "PART" (particle) and "INTJ" (interjection). While it is not particularly useful to compute both lexical density and RFFW, it was used to verify previously obtained information.

- RFFW results are available in Section 4.1.5.

While not particularly useful on its own, **sentence count** per text is based on spaCy's sentence segmentation and reflects the total length of the texts. They are included here for reference only. **Average sentence and word length** per text were also calculated and provide insight into a text's difficulty, as they form the basis for computing readability levels.

- Sentence metrics are available in Section 4.1.6.

Readability levels were calculated using textstat, a Python library dedicated to textual metrics such as "*readability, complexity, and grade level*" (Shivam & Chaitanya, 2014). We focused on calculating FRE (Flesch, 1948) and FKGL (Kincaid et al., 1975) metrics which are based on sentence and word length, as we previously described in Section 2.3.6.

- Readability levels results are available in Section 4.1.7.

POS Tags Distribution: during preprocessing, spaCy assigned a part-of-speech tag to each token, which enabled us to compute a probability score for each tag.

- POS Tags Distribution results are available in Section 4.1.8.

For the **TF-IDF** analysis, the documents were lowercased and stripped of punctuation and stop words before lemmatization. We used scikit-learn and limited the vocabulary to 2,000 terms (*max_features=2000* in *scikit-learn's* *TfidfVectorizer*) to reduce noise and exclude very rare terms unlikely to be significant across the corpus. *TfidfVectorizer* was fitted to the entire corpus, and for each document, a list of ten lemmatized words was then selected based on their TF-IDF scores.

- TF-IDF results are available in Section 4.1.9.

3.3.2 Modality Analysis

Objective: To analyse the expression of certainty and uncertainty based on the studies by Roeder (2011) and Herrando-Pérez et al. (2019)

Modality is expressed through modal verbs and adverbs, as well as likelihood and confidence expressions. All these elements play a crucial role in effective CC communication. According to Roeder (2011) and Herrando-Pérez et al. (2019) The IPCC's use of modality and expressions of uncertainty reflects the scientific rigour and complexity involved in

generating its reports, as such choices inevitably shape how the public and policymakers perceive the urgency of climate change.

For modality analysis, we first calculated the raw frequency of modal verbs and adverbs for each document in our corpus and compared their normalized frequency over time.

The analysis used the previously pre-processed spaCy Doc objects (Section 3.2.2) and filtered modal expressions using a combination of POS tags and a curated list of verbs and adverbs. Likelihood and confidence expressions were also retrieved using a list borrowed directly from the IPCC. This approach is considered rule-based, as it does not take context into account, relying solely on simple word filtering and detection.

For modal verbs, the curated list was the following: "can", "could", "may", "might", "must", "shall", "should", "will", "would". These are the core modal auxiliaries and their preterit version in English. For modal adverbs, the list is a little longer. In the Cambridge Grammar of the English language, (Huddleston & Pullum 2002, p. 102) the following list was adopted for our analyses: "apparently", "arguably", "assuredly", "certainly", "clearly", "conceivably", "definitely", "doubtless", "evidently", "hopefully", "indubitably", "ineluctably", "inescapably", "incontestably", "likely", "manifestly", "maybe", "necessarily", "obviously", "patently", "perhaps", "plainly", "possibly", "presumably", "probably", "seemingly", "surely", "truly", "unarguably", "unavoidably", "undeniably", "undoubtedly", "unquestionably".

We filtered all verbs and adverbs by selecting tokens with the POS tags "AUX" for verbs and "ADV" for adverbs, then refined the selection using the previously presented lists of modal verbs and adverbs. Finally, we calculated the normalized frequency of each item per text.

While it would have been possible to analyse the use of each modal verb and adverb individually, such a detailed analysis falls outside the primary scope of this master's thesis. Instead, we opted to compute the total frequency of both categories as the basis for our comparison.

- Results for modal verb and adverbs analysis are available in Section 4.2.1

Likelihood and confidence expressions are a specific category of standardized language introduced by the IPCC to convey degrees of certainty through quantified modalities. They are listed in *Table 4* and *Table 5* (Le Treut & Somerville, 2007)

Confidence Terminology	Degree of confidence in being correct
Very high confidence	At least 9 out of 10 chance
High confidence	About 8 out of 10 chance
Medium confidence	About 5 out of 10 chance
Low confidence	About 2 out of 10 chance
Very low confidence	Less than 1 out of 10 chance

Table 4: IPCC Confidence Terminology¹³

¹³ Original tables accessible on archive.ipcc.ch/publications_and_data/ar4/wg1/en/ch1s1-6

Likelihood Terminology	Likelihood of the occurrence / outcome
Virtually certain	> 99% probability
Extremely likely	> 95% probability
Very likely	> 90% probability
Likely	> 66% probability
More likely than not	> 50% probability
About as likely as not	33 to 66% probability
Unlikely	< 33% probability
Very unlikely	< 10% probability
Extremely unlikely	< 5% probability
Exceptionally unlikely	< 1% probability

Table 5: IPCC Likelihood Terminology

Expressions of likelihood and confidence were retrieved using the entries in Tables 4 and 5, along with a simple rule-based text matching approach.

- Results for likelihood and confidence modal expressions are available in Section 4.2.2.

3.3.3 Semantic Similarity Comparison

Objective: To evaluate the semantic correspondence between paired IPCC and Wikipedia documents.

As mentioned in Section 2.6.1, we assess the semantic similarity between the IPCC WG3 SPM and Wikipedia articles on Climate Change Mitigation (CCM) by encoding the texts as vectors using Sentence-BERT embeddings (Reimers & Gurevych, 2019) and applying cosine similarity to evaluate their degree of alignment in a multi-dimensional vector space.

While we previously measured and evaluated each document to compare them collectively and track the evolution of both the IPCC and Wikipedia, directly comparing the IPCC documents to one another would have been of limited value, as semantic differences between them are expected.

Instead, we focused on comparing pairs of documents: the IPCC SPMs and their corresponding Wikipedia articles from the same period, as shown in *Table 6*:

<i>AR3-WG3-SPM (2001)</i>	<i>Wikipedia:Climate Change Mitigation (2005)</i>
<i>AR4-WG3-SPM (2007)</i>	<i>Wikipedia:Climate Change Mitigation (2008)</i>
<i>AR5-WG3-SPM (2014)</i>	<i>Wikipedia:Climate Change Mitigation (2014)</i>
<i>AR6-WG3-SPM (2022)</i>	<i>Wikipedia:Climate Change Mitigation (2022)</i>

Table 6: Document Pairs for Semantic Similarity Comparison

As the first two IPCC WG3 SPM from 1990 and 1995 do not have corresponding Wikipedia articles, they are not included in this analysis.

There are various methods for computing semantic similarity. Some are more qualitative and involve first identifying relevant or similar paragraphs or sections to compare, before calculating a similarity score.

Since the structures of the documents differ significantly, identifying comparable paragraphs would have introduced additional complexity and potential room for error, particularly if unsuitable sections were selected. Instead, we adopted an automatic approach by calculating a similarity score for each document using sentence-level similarity:

Each document was segmented into sentences using spaCy, and individual sentences were stripped using `.strip()` (which removes spaces, tabs, newline characters like “\n”). Once stripped, embeddings were generated for every sentence in both documents. Semantic similarity was then calculated for each sentence, and the highest similarity score (i.e. the closest semantic match) was recorded. The final score was obtained by averaging these maximum sentence similarity values.

For this analysis, we used documents that have been preprocessed with spaCy (Section 3.2.2), and the sentence-transformers 4.1.0 Python library using the “all-MiniLM-L6-v2” model (Aarsen, 2024). This is an efficient 384-dimensional model, but it is limited to 256-word pieces per sentence. As a result, although it could have been interesting for the comparison, we did not compute entire documents as single embeddings.

- Semantic similarity results are available in Section 4.3.

3.3.4 Topic Modelling

Objective: To identify, group and compare thematic structures within the documents

Topic modelling typically identifies clusters of words that frequently co-occur, helping to reveal distinct underlying topics that are usually imperceptible to the human eye.

As mentioned in Section 2.5, since the corpus is not large enough for classic topic modelling, the Python library BERTopic (Grootendorst, 2022) will be the principal tool for this analysis. BERTopic can use embeddings and clustering techniques to extract thematic structures or “topics”. While BERTopic should produce better results than LDA on this corpus due to its reliance on sentence embeddings (which tend to perform better on smaller corpora) it does not necessarily guarantee that our results are of exploitable value given the very limited size of our corpus.

For topic modelling, we used lowercased, lemmatized text with punctuation and stop words removed, as pre-processed with spaCy in Section 3.2.2. Lemmatization seemed necessary in the case of this corpus, even though it further reduces an already limited vocabulary. Without lemmatizing the text, the semantic impact of words would have been diminished, as topics would have been influenced by word forms. For example, BERTopic would have treated “emission” and “emissions” as distinct, rather than recognising them as part of the same concept.

The “all-MiniLM-L6-v2” sentence-transformer model (Aarsen, 2024) was used to generate embeddings for BERTopic, which has all other parameters set to default (UMAP for dimensionality reduction, HDBSCAN for clustering)¹⁴. While we could have manually selected the number of topics, BERTopic can determine this automatically, so we decided to leave it

¹⁴ BERTopic documentation, which gives some details about these methods, is accessible on maartengr.github.io/BERTopic/

on “auto”. The minimum topic size was set to 2 to ensure that topics were formed by at least two documents.

- Topic modelling results are available in Section 4.4.

3.3.5 Sentiment Analysis and Emotion Detection

Objective: To compare sentiment polarity and emotions expressed in the documents.

While working on the SPM and associated coverage, Barkemeyer et al. (2016) previously found that the IPCC aims for a neutral tone, while the media had become increasingly pessimistic. From this perspective, we expected the SPM to remain neutral and Wikipedia to convey more emotions and/or express a negative polarity. To have different perspectives, we decided to perform sentiment analysis using a rule-based model called VADER, that is part of NLTK library, and both sentiment analysis and emotion detection using transformer-based models.

Firstly, we had to preprocess the texts and create paragraph-level extracts, as sentiment and emotion analysis cannot be effectively performed on individual sentences due to the lack of context. Sentiment analysis or emotion detection with an entire document at once is also not practical, both because it offers limited insight and because of limitations related to embedding size, as discussed in Section 3.3.3.

The original documents contained some paragraphs that were lost during the conversion to raw text. In fact, all newline characters were removed, and sentences were separated by a single space. This means that while spaCy's preprocessing worked for sentence separation, as sentences are easily detected by full stops (.), it wouldn't reliably work for paragraph detection.

Since we had no means of detecting and retrieving the original paragraphs, we needed to either:

- Read the original documents and manually separate the paragraphs for each text, which is very time-consuming.
- Divide the text into artificial paragraphs every N sentence, which would return unreliable results for analysis, as the context would be truncated.
- Use the Text Tiling method, introduced by Hearst (1997). Text Tiling refers to a text segmentation technique that detects topic shifts using bag-of-words-based cosine similarity between adjacent blocks. However, the method is purely lexical and its ability to capture semantic information is more limited than what most recent approaches based on embeddings can offer.
- Semantically chunk the text using transformer-based embeddings, this type of chunking is a complex technique typically used in Retrieval-Augmented Generation (RAG) applications.¹⁵ This would recreate semantically linked paragraphs, which should be more effective than text tiling or splitting paragraphs every N sentences.

While reading the original documents would have been the most accurate option, it was also the most tedious and unrealistic. In theory, semantic chunking is a close alternative to the original documents, as paragraphs are usually separated according to shifts in topic. Considering this, we decided to semantically chunk the texts.

¹⁵ RAG is a technique designed to improve LLMs responses by retrieving relevant information from an external knowledge source to generate a more accurate answer.

Semantic chunking can be a highly complex task involving multiple steps. To facilitate this process, we decided to use an LLM. As mentioned in Section 3.2.1, LLMs are effective at a wide range of tasks, with semantic representation being their primary strength. Therefore, given a precise prompt and a sufficiently large context, an LLM can return a document with clearly separated paragraphs.

Given the large context window and output length of Google's Gemini 2.5 Pro, we decided to use it again for this task. As of May 2025, Gemini 2.5 Pro was updated to Preview 05/06, so this specific version was used for semantic chunking with a few-shots learning approach using *Prompt 2*:

This a natural language processing task.
For an analysis, semantic chunking of a text is required.
For each raw text we send you, return the full text semantically chunked. Separate chunks with a blank line. Texts have been cleaned for NLP. There are no natural paragraphs you can rely on.
Ensure that no chunk exceeds 500 tokens in length.
 ----- OUTPUT EXAMPLE -----
 [Wiki_CCM_2005-06-26 as raw text split manually as an example]
 ----- OUTPUT EXAMPLE -----
 Are you ready to process texts?

Prompt 2: Semantic Chunking with Gemini 2.5 Pro Preview 05-06

This process took several attempts, and many different prompts were tried. After a few tries, this resulted in the most efficient version, shown in Prompt 2. Context and goal are mentioned first, and a clearly defined, hand-crafted example allows the model to truly understand the task. Without an example (zero-shot learning setting), the model was not able to begin any file processing.

Since this was a very intensive task, there was a risk of documents leaking into one another if processed within the same context window. The context window was therefore reset after each document was processed.

Content was verified after processing by reading some extracts, ensuring paragraphs were split semantically, and by checking the file size, which was confirmed to be very similar to the previous version,¹⁶ indicating that only a few lines were modified but the overall content remained unchanged.

Once all documents were processed, the document used as the example (initially converted manually) was also processed by the model, by using another document as an output example instead, to standardise the formats for analysis.

Documents were not subjected to further pre-processing, such as lowercasing, as VADER automatically lowercases text for analysis, whereas transformer-based models used in this thesis are case-sensitive.

While VADER is rule-based and comes in a standard static release, there are many different transformer-based models available for sentiment and emotion analysis. This meant we had to select one suited to our task. Our reasoning was as follows: since we were working with scientific documents, BERT models trained on social media data would not be appropriate, as social media typically expresses far more emotion than scientific writing. Therefore, we

¹⁶ A difference of 1 kB, which is approximately a 1% size difference per document on average.

considered that models such as “*twitter-roberta-base-sentiment*” (Barbieri et al., 2024) which are specifically trained on Twitter data, would not be suitable for our task and could produce biased results.

On the other hand, models specifically trained on climate change contexts were potentially risky, as they are often trained on the IPCC reports themselves, which could have led to inaccurate results due to performing inference on data included in the training set.

Finally, we decided to use the following models:

- For sentiment analysis, we used “*distilroberta-finetuned-financial-news-sentiment-analysis*” (Romero, 2024), as it is a very general model trained on financial news, which is similar in tone and register to climate change discourse.
- For emotion detection, we selected “*emotion-english-distilroberta-base*” (Hartmann, 2022) which appeared to be an industry standard, judging by its popularity on HuggingFace.

Both models are also distilled,¹⁷ which was a welcome advantage given hardware limitations.

- Sentiment analysis and emotion detection results are available in Section 4.5.

3.3.6 Named Entity Recognition Analysis

Objective: To investigate the different entities mentioned in the documents and provide a comparative analysis of entity distribution, while assessing the findings of Korte et al. (2023).

As discussed in Section 2.6.3, Named Entity Recognition (NER) is a classification task that categorises words into named entity classes. By applying NER and examining the distribution of entities across the documents, we were able to compare patterns in the IPCC WG3 SPMs and Wikipedia CCM articles, which allowed us to assess Korte et al. (2023) claim that Wikipedia tends to frame climate change through events and personalities.

During preprocessing (Section 3.2.2), spaCy performed several tasks, including automatically tagging named entities, using the OntoNotes 5.0 categories discussed in Section 2.6.3. This means NER results were saved to disk along with preprocessed texts by using spaCy’s DocBin class and can be retrieved and analysed immediately.

- spaCy’s NER results are available in Section 4.6.1

spaCy’s NER provided interesting insights into the texts, but we considered that the OntoNotes 5.0 annotations were not sufficiently specific for an analysis focusing on climate change discussions.

Therefore, we decided to conduct NER again using a model specialised in identifying and labelling entities related to CC. As such, for the second analysis, we employed a specialised BERT model trained on a dedicated climate change NER dataset available on HuggingFace.

- The selected model was “*nicolauduran45/specter-climate-change-NER*” (Duran, 2024), which is based on the Specter2 model architecture (Singh et al., 2022).
- This model was trained on the *Ibm-Research/Climate-Change-NER* (2024) dataset developed by Bhattacharjee et al. (2024).

Instead of the OntoNotes categories, this NER dataset has specific categories tailored to climate change, they are listed in *Table 7* below.

¹⁷ Distillation is a process that reduces the size of a model while preserving most of its quality,

Categories	Description
climate-assets	Objects or services of value to humans that can get destroyed or diminished by climate-hazards. Key categories are: health, buildings, infrastructure, and crops or livestock.
climate-datasets	Specific collections of climate data with a name. A climate dataset can be the result of observations or of a model. e.g., as a prediction or reanalysis. The data may be lists, tables, databases, inventories or historical records, where the data dominate over attached code.
climate-greenhouse-gases	Gases that cause heating of the atmosphere (greenhouse gases).
climate-hazards	Hazards with potential negative impact on climate, such as floods, wildfires, droughts, and heatwaves. Where a hazard is named in more detail in a text, the entire term is annotated. e.g., surface water flood or soil liquefaction.
climate-impacts	Effects of hazards, primarily negative effects on humans. We also consider impacts on livestock as impacts, as it indirectly affects humans.
climate-mitigations	Activities to reduce climate change or to better deal with the consequences.
climate-models	Specific physical, mathematical, or artificial intelligence objects. Nowadays always computer-executable, used to analyze and usually predict climate parameters.
climate-nature	Aspects of nature that are not alive, such as oceans, rivers, the atmosphere, winds, and snow.
climate-observations	Climate observation tools with a name. (satellites, radiospectrometers, rain gauges, wildlife cameras, and questionnaires)
climate-organisms	Animals, plants, and other organisms that are considered for their own sakes (in contrast to as food for humans) as climate organisms.
climate-organizations	Real-world organizations with climate-related interests.
climate-problem-origins	problems that describe why the climate is changing. Key examples are fossil fuel and deforestation. We also mention sectors that can be cited as causes of energy use. For instance, in a text about the energy consumption by the transport sector, transport sector is annotated as problem.

climate-properties	properties of the climate itself (not abstract objects like models and datasets) that typically come with values and units.
--------------------	---

Table 7: Climate-Change-NER Categories

Source: huggingface.co/datasets/ibm-research/Climate-Change-NER

Transformer-based models we used in this analysis are highly context-dependent and case-sensitive. As such, we needed raw, unprocessed text for NER. However, as with sentiment analysis (Section 3.3.5), the model is unable to process an entire document at once.

Theoretically, some NER models allow for the input of an entire raw text without the need to split the document into sentences or paragraphs, as they use a “sliding window technique” (Gallo et al., 2008) for segmenting large documents. However, “*nicolauduran45/specter-climate-change-NER*” did not work with individual documents, and consequently, this suggests that the model is limited to a typical 512-token context window, as it is a fine-tuned version of *specter2_base* which, according to its HuggingFace entry, was trained with a 512-token context limit.¹⁸

To address the context window limitation, the most straightforward approach was to reuse the semantically segmented texts previously employed for sentiment and emotion analysis. These were used to run inference with the model, count entities within each paragraph of each text, and then calculate their frequency.

As a final methodological note, for both spaCy’s and Climate NER categories, our calculation of the proportions of entities for each type (see Sections 4.6.1 and 4.6.2) is based on word counts derived from spaCy’s token objects. These objects include attributes indicating whether a token is punctuation or a stop word. This allowed us to easily exclude such tokens from the count, as their inclusion would otherwise distort entity frequency measurements in NER analyses.

3.4 Section Summary: Corpus and Methods

To conclude this section, we outlined the methodological approach that guided our comparative analysis. We described the creation of a diachronic corpus comprising IPCC SPMs and Wikipedia CCM articles, the rigorous process of data acquisition, preprocessing, and the deployment of a diverse range of NLP techniques. These techniques include lexicometry, modality analysis, and state-of-the-art models for tasks such as semantic comparison, sentiment analysis, emotion detection and NER. This comprehensive approach ensures the reliability of the findings presented in the following section, “Results and Discussion”.

¹⁸ Specter2_base HuggingFace page is accessible on huggingface.co/allenai/specter2_base

4 Results and Discussion

The preceding sections have established the purpose of the analyses presented in this thesis. These analyses are guided by the following research question, which we now formulate in explicit terms: How does the portrayal of the same issue differ between the IPCC WG3 SPMs and the Wikipedia CCM articles, with particular attention to potential bias in the Wikipedia texts?

Previously, Section 1 reviewed the relevant literature. Section 2 defined core NLP concepts, and Section 3 outlined the methodology used to address the research question. The current section presents our results and their discussion. We acquired textual data and applied the previously outlined methodology, leading to the interpretation of the resulting findings.

Reproducibility is a crucial factor in any scientific experiment. Although computers are designed to operate deterministically, variations in hardware, operating systems and software versions can introduce unpredictability. This challenge is particularly intense with Large Language Models, often described as black boxes due to potential variability in output despite their mathematical foundations.

To enhance experimental stability and maximise the potential for reproducibility, detailed information about software versions is provided in Appendix A. Python code used in the analyses is also included in the appendices, with each subsection indicating the corresponding appendix.

In addition, the Python scripts and (a selection of¹⁹) the raw texts will be published on GitHub. All links to documents and web pages referenced in this thesis, including the GitHub repository, are compiled in Appendix B for convenience.

We begin with lexicometric, stylistic, and readability analysis (4.1), before turning to modality analysis (4.2). We then focus on techniques involving embeddings with semantic similarity comparison (4.3), topic modelling using BERTopic (4.4), sentiment analysis and emotion detection (4.5), and conclude with NER (4.6).

4.1 Lexicometry, Stylistic, and Readability Analysis

This subsection explores the quantitative and stylistic characteristics of our corpus. By applying a suite of lexicometric, stylistic, and readability measures, we aim to objectively quantify variations in vocabulary richness, syntactic patterns, and overall textual complexity. These metrics allow us to observe how such linguistic features evolve over time and across texts.

We begin with an initial exploration using TXM (4.1.1), followed by detailed analyses of word count (4.1.2), lexical diversity (4.1.3) and density (4.1.4), the frequency of function words (4.1.5), sentence and word metrics (4.1.6), readability levels (4.1.7) and part-of-speech (POS) tag distributions (4.1.8). Finally, we compute TF-IDF scores to highlight term significance across the corpus (4.1.9).

Recall that the different measures were defined in the Section 2.3.6 above, and their implementation for this thesis is described in the related methodology section (3.3.1 above). As an additional note, Python code for Section 4.1 and its subsections is available in Appendix F.

¹⁹ The IPCC copyright policy prohibits the redistribution of their work.

4.1.1 Initial TXM Exploration

The first significant aspect to observe is the size of the texts, text size in number of words is shown in Figure 5:

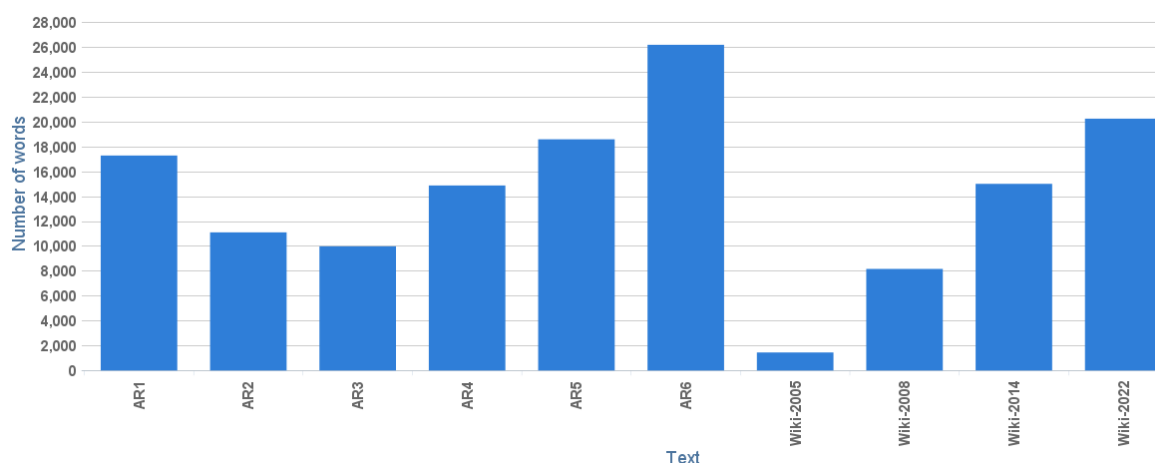


Figure 5: Text size by Number of Words as Computed with TXM’s “Dimensions” Tool

We observed that while SPMs show more gradual changes in terms of document sizes, with a decline after AR1 and increase after AR3, Wikipedia CCM articles displayed a more abrupt upward trajectory, possibly indicating a growing interest about CC over time. But this is hardly the only possible explanation, as it could also be due to the growing interest in Wikipedia, or the increased accessibility of the Internet, which is far more common now than it was in 2005-2008. As such, we do not consider document size a reliable factor in this corpus, and its impact is mitigated through normalisation of any measures or analysis.

The most compelling aspect of TXM is how easily it facilitates the exploration of a corpus using features such as the lexicon, concordance, and co-occurrences. However, we were unable to identify any meaningful differences between the texts, apart from the observation that the Wikipedia texts seemed to use simpler vocabulary.

To pursue the analysis further in TXM, unprocessed raw text was no longer sufficient. To obtain more insightful results with TXM, we would have needed to import different versions of the texts: one with lowercased text to standardise the lexicon, and one with stop words removed.

Although preparing multiple processed versions of the data for use in TXM was possible, it would have added an additional layer of preparation that was not necessary for the specific quantitative comparisons addressed in this study.

Since our main goal was to systematically compute and compare predefined metrics mentioned at the beginning of this section, a straightforward approach using Python offered a more efficient and targeted alternative.

4.1.2 Word Count

Word count per document obtained with spaCy is shown in *Table 8*.

Document	Total Tokens	Excluding Punct/stop-words
AR1_WG3_SPM	16983	9043
AR2_WG3_SPM	11224	5767
AR3_WG3_SPM	9739	5261
AR4_WG3_SPM	14496	8086
AR5_WG3_SPM	17667	9399
AR6_WG3_SPM	26100	14328
Wiki_CCM_2005-06-26	1459	769
Wiki_CCM_2008-05-07	8156	4466
Wiki_CCM_2014-09-02	14888	7997
Wiki_CCM_2022-06-13	20477	10835

Table 8: Token Count per Document Obtained with spaCy

While we previously observed an unprocessed word count using TXM, we are now able to exclude specific categories of words thanks to spaCy's preprocessing. We excluded punctuation and stop-words in a second column. While not useful on its own, this metric is a foundation for other metrics such as average sentence length and POS tag frequencies, that we will observe later.

4.1.3 Lexical Diversity

Table 9 presents the TTR, Herdan's C and Guiraud's R which were defined in Section 2.3.6 above. Recall from 3.3.1 that we computed the values based on lowercased lemmatized tokens. Values were rounded to 3 decimal places to facilitate reading. In this table, TTR should not be considered a reliable basis for comparison between the texts, as it does not account for text length.

Document	TTR	Herdan's C	Guiraud's R
AR1_WG3_SPM	0.220	0.834	20.895
AR2_WG3_SPM	0.225	0.828	17.119
AR3_WG3_SPM	0.248	0.837	18.019
AR4_WG3_SPM	0.222	0.833	19.939
AR5_WG3_SPM	0.196	0.822	19.041
AR6_WG3_SPM	0.147	0.799	17.561
Wiki_CCM_2005-06-26	0.563	0.914	15.614
Wiki_CCM_2008-05-07	0.364	0.880	24.346
Wiki_CCM_2014-09-02	0.295	0.864	26.368
Wiki_CCM_2022-06-13	0.254	0.853	26.486

Table 9: TTR, Herdan's C, Guiraud's R (rounded values, 3 d.p.)

By contrast, Herdan's C and Guiraud's R appeared to be more robust, except for the 2005 Wikipedia article for Guiraud's R, which produced an unexpected result.

Nevertheless, focusing on Herdan's C revealed a slight decline in lexical diversity over time in both Wikipedia and IPCC documents, although Wikipedia consistently exhibits greater lexical diversity overall.

Lower lexical diversity in the IPCC reports may reflect repetitive or technical phrasing, which is typical of scientific writing, whereas the slightly broader vocabulary observed in Wikipedia articles could be attributed to the varied authorship inherent to the platform.

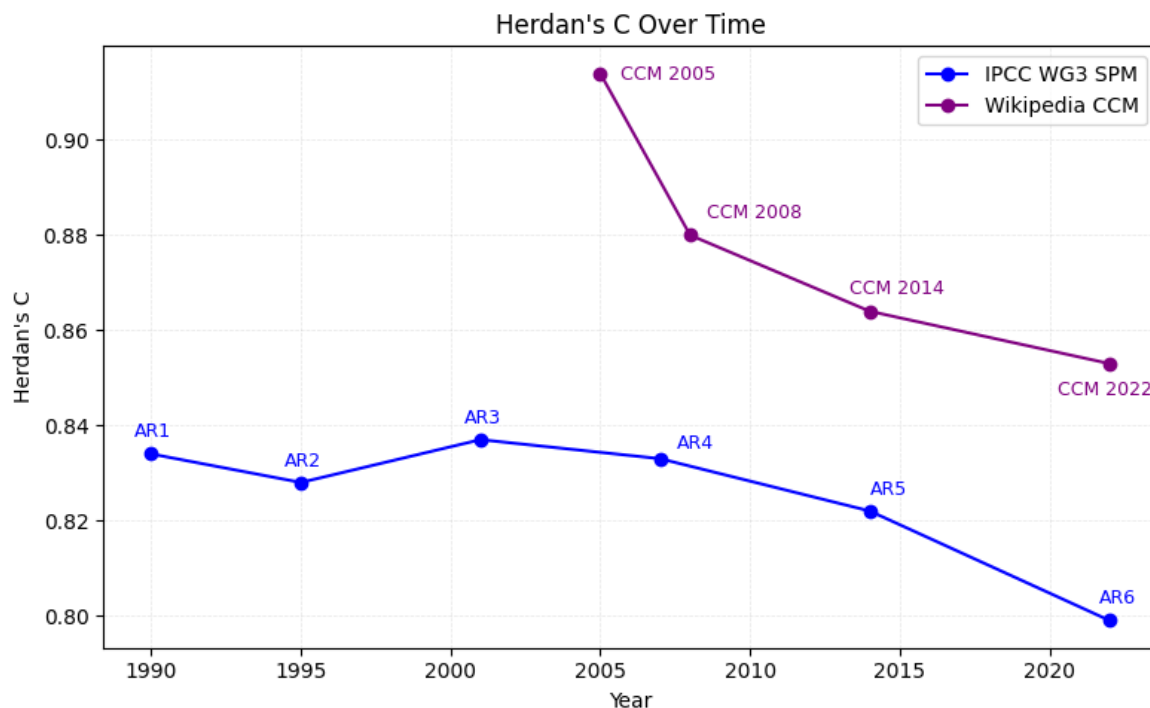


Figure 6: Evolution of Herdan's C Over Time

Source: Figures are generated using the Python script available in Appendix O

While Figure 6 shows a slight trend of declining lexical diversity over time in the WG3 SPMs (in blue), a larger similar trend can be observed in Wikipedia (purple), which could be linked to improvements in article quality and fidelity to the IPCC over time.

4.1.4 Lexical Density

Lexical Density results are available in Table 10. Figure 7 provides a visual representation of the results. Recall that lexical density (2.3.6) refers to the proportion of content words over total words, and that the status of each word was based on spaCy POS tags (see 3.3.1 above), considering numbers as content words given their informative role when discussing scientific issues.

Document	Lexical Density
AR1_WG3_SPM	0.641
AR2_WG3_SPM	0.621
AR3_WG3_SPM	0.658
AR4_WG3_SPM	0.697
AR5_WG3_SPM	0.678
AR6_WG3_SPM	0.683
Wiki_CCM_2005-06-26	0.642
Wiki_CCM_2008-05-07	0.664
Wiki_CCM_2014-09-02	0.667
Wiki_CCM_2022-06-13	0.648

Table 10: Ure's Lexical Density per Document (rounded, 3 d.p.)

The closer lexical density is to 1, the denser (i.e. information heavy) are the documents.

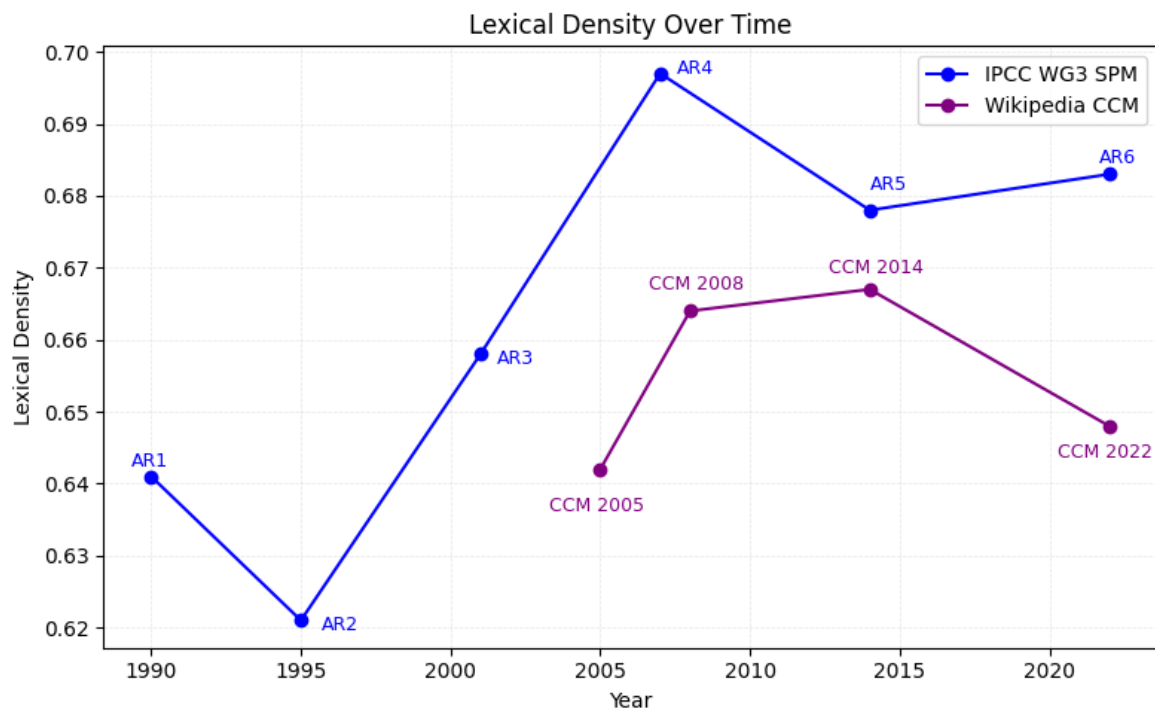


Figure 7: Evolution of Ure's Lexical Density Over Time

According to Figure 7, the IPCC maintains a consistently higher lexical density than Wikipedia across all time points, except for AR1 and AR2, when there was no equivalent. This is likely due to the IPCC's more technical language. Although Wikipedia's articles have increased slightly in lexical density over the years, the 2022 version remains below that of the IPCC AR6, indicating that Wikipedia has not yet matched the IPCC's level of technical language. This suggests that Wikipedia still maintains a less technical and more accessible tone compared to the IPCC.

4.1.5 Relative Frequency of Function Words

Document	Rel. Freq. Function Words (%)
AR1_WG3_SPM	35.55
AR2_WG3_SPM	37.67
AR3_WG3_SPM	33.78
AR4_WG3_SPM	29.23
AR5_WG3_SPM	32.00
AR6_WG3_SPM	31.18
Wiki_CCM_2005-06-26	35.69
Wiki_CCM_2008-05-07	33.47
Wiki_CCM_2014-09-02	33.12
Wiki_CCM_2022-06-13	35.01

Table 11: Relative Frequency of Function Words per Document (rounded, 2 d.p.)

As mentioned in Section 3.3.1, RFFW can be seen as a reversed lexical density. As such, we used the RFFW values from Table 11 to produce Figure 8, but we reversed the Y axis which represents the values, and observed if Figure 8 and Figure 7 are visually similar.

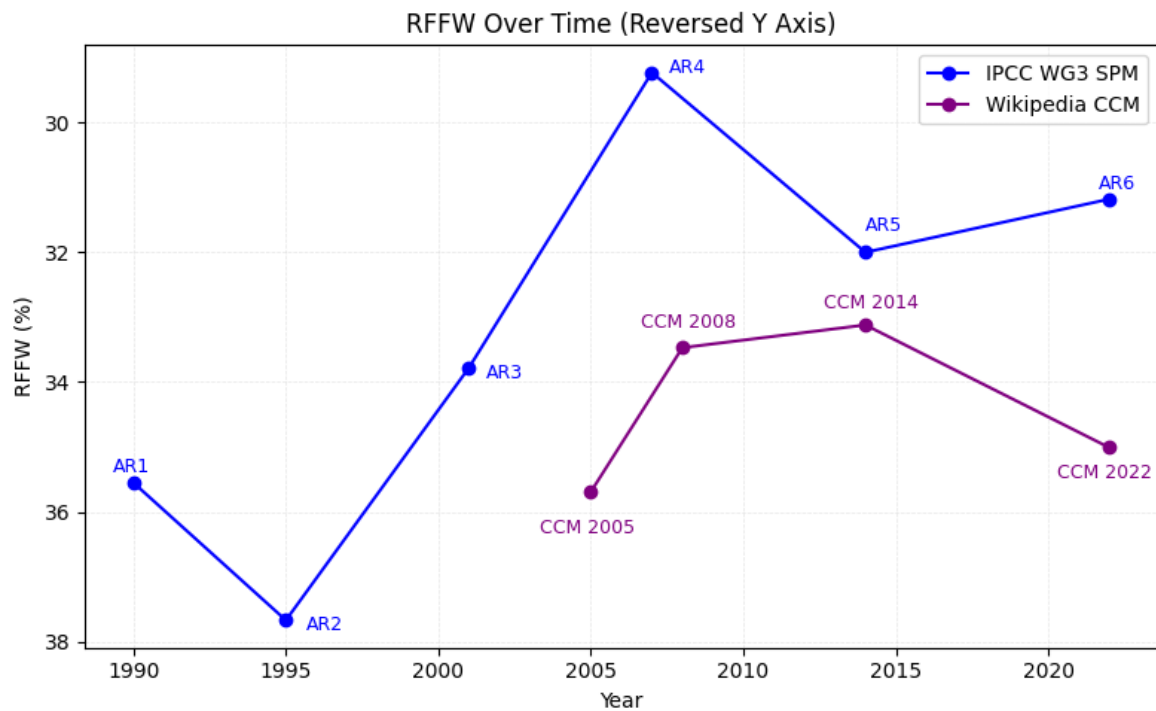


Figure 8: Relative Frequency of Function Words Over time (reversed Y Axis)

This hypothesis was confirmed, as Figure 8 appeared to be an exact replica of Figure 7, thereby supporting the results on lexical density just presented (4.1.4).

4.1.6 Sentence Count, Length, and Word Length

Sentence count based on spaCy's segmentation, average length (in words), and average word length (in characters) are available in Table 12.

Document	Sentence count	Avg. Sentence Length (words)	Avg. Word Length (chars)
AR1_WG3_SPM	555	26.9	5.7
AR2_WG3_SPM	420	23.9	5.7
AR3_WG3_SPM	343	24.6	5.6
AR4_WG3_SPM	558	21.7	5.4
AR5_WG3_SPM	580	25.2	5.5
AR6_WG3_SPM	896	24.2	5.6
Wiki_CCM_2005-06-26	48	26.8	5.2
Wiki_CCM_2008-05-07	263	27.3	5.4
Wiki_CCM_2014-09-02	583	21.8	5.4
Wiki_CCM_2022-06-13	786	22.8	5.3

Table 12: Sentence Count, Length (in words) and Word Length (in characters) per Document (rounded, 1 d.p.)

Unfortunately, these metrics alone were not sufficient to provide insight into a text's readability. That is why, in the next section, we applied two different methods to assess readability based on sentence and word length.

4.1.7 Readability

As mentioned in 3.3.1, readability levels were calculated using textstat. Results are available in Table 13. The measures were defined in Section 2.3.6.

Document	FRE	FKGL
AR1_WG3_SPM	22.34	16
AR2_WG3_SPM	13.68	17.2
AR3_WG3_SPM	22.45	15.9
AR4_WG3_SPM	25.49	14.7
AR5_WG3_SPM	24.68	15.1
AR6_WG3_SPM	25.49	14.7
Wiki_CCM_2005-06-26	27.76	15.9
Wiki_CCM_2008-05-07	29.79	15.2
Wiki_CCM_2014-09-02	25.8	14.6
Wiki_CCM_2022-06-13	33.34	13.8

Table 13: Readability Scores Based on the Flesch Reading Ease (FRE, 1948) and its Later Development, the Flesch-Kincaid Grade Level (FKGL, 1975)

FRE produces a score out of 100. The higher the score, the easier the text is to read. A score between 0 and 30 is generally considered to indicate college-level difficulty.

FKGL is expressed as a U.S. school grade. In the United States, grade 13 corresponds to the first year of university, while grade 17 refers to the first year of a postgraduate programme (Master's or PhD).

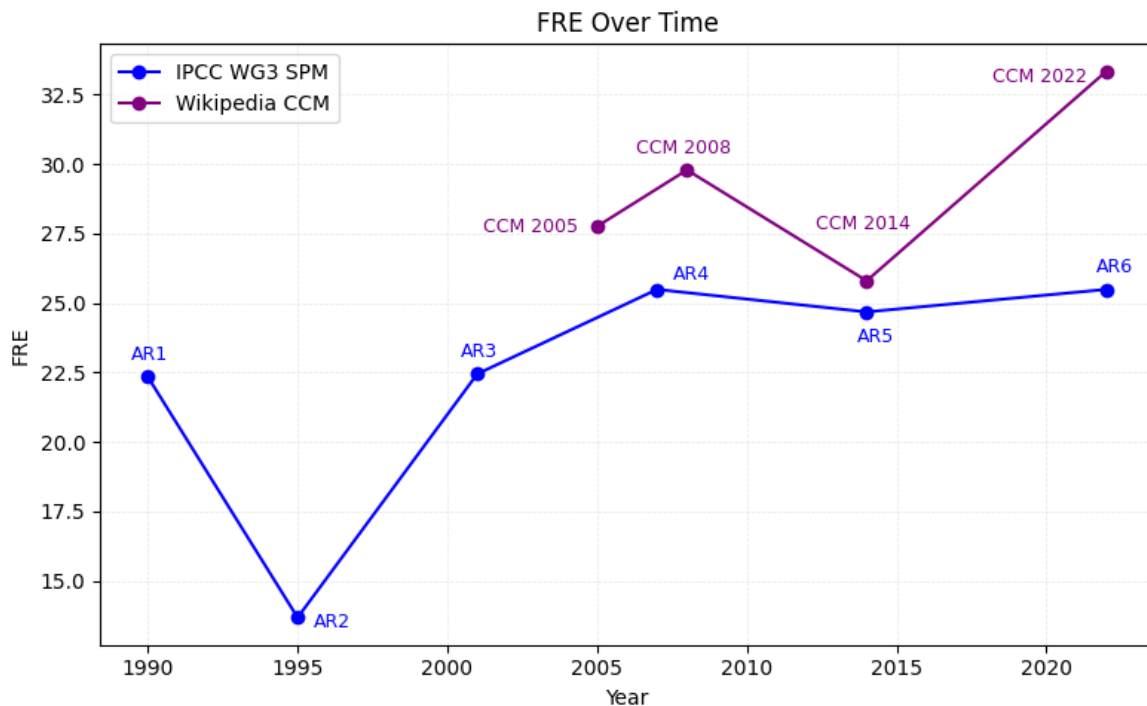


Figure 9: Flesch Reading Ease Over Time

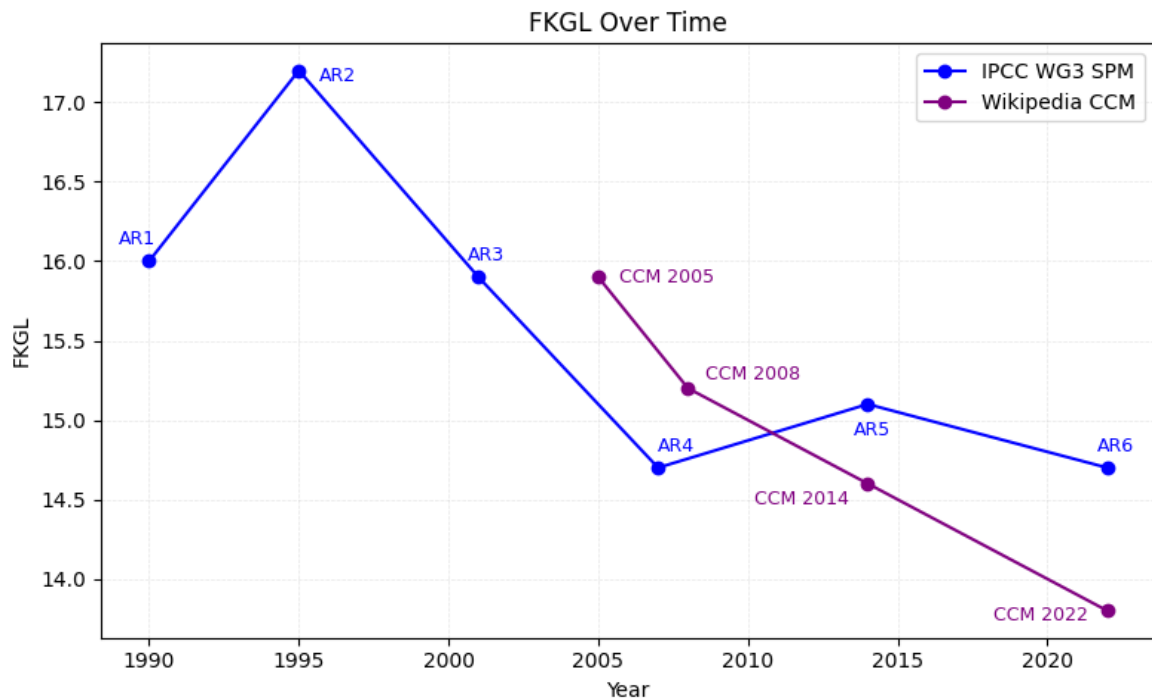


Figure 10: Flesch-Kincaid Grade Level Over Time

According to textstat results in Table 13, the SPMs and Wikipedia's CCM are rated as college-level texts, which seems appropriate.

As indicated by Figure 9 and Figure 10, readability has improved over time. The FRE score increased for both Wikipedia and the IPCC; however, the IPCC's score has remained stable since AR4. A similar trend is observed for the FKGL, although it is inversely related, as the lower the score, the easier the text is to read.

Finally, Wikipedia has shown a clear trend of increasing readability over time. This may be intentional, as Wikipedia makes a conscious effort to enhance readability by offering editorial advice to its contributors. (Wikipedia, 2024c)

4.1.8 POS Tags Distribution

We did not cover all the possible tags applied by spaCy to analyse the distribution of POS tags, as we believe some are irrelevant or occur with very low frequency. We highlighted the most important ones for our analysis in Table 14.

Document	NOUN	VERB	AUX	ADJ	ADV	PRON	ADP	CCONJ	SCONJ
AR1_WG3_SPM	32.0	10.3	4.4	11.3	2.8	2.2	13.3	5.4	1.3
AR2_WG3_SPM	32.5	10.4	5.2	12.7	3.2	2.1	13.8	4.8	1.4
AR3_WG3_SPM	32.7	9.0	3.9	11.4	2.5	1.8	13.5	5.3	1.1
AR4_WG3_SPM	32.2	8.4	3.8	10.9	2.4	1.6	12.0	4.5	0.9
AR5_WG3_SPM	31.7	8.3	3.7	11.7	2.7	1.4	14.0	4.4	0.9
AR6_WG3_SPM	32.7	10.6	3.3	12.6	2.1	1.5	13.6	5.9	0.7
Wiki_CCM_2005-06-26	26.4	11.0	3.7	7.2	2.3	2.8	12.7	2.6	1.6
Wiki_CCM_2008-05-07	27.5	10.3	4.3	8.9	2.9	2.2	12.6	3.3	1.2
Wiki_CCM_2014-09-02	26.5	9.9	4.5	9.1	2.8	1.9	12.8	3.2	1.4
Wiki_CCM_2022-06-13	30.5	10.9	5.1	10.0	3.1	2.0	13.3	3.8	1.4

Table 14: Distribution of Relevant POS Tags (percentages)

Nouns (NOUN): A higher proportion of nouns is found in SPMs. This is consistent with the general understanding that a higher noun count suggests a text is more informational or descriptive, aligning with the SPMs' primary purpose. Wikipedia's 2022 article showed a similar trend, confirming earlier observations that Wikipedia is increasingly aligning with the IPCC style.

Verbs (VERB): Wikipedia tends to use more verbs overall. This suggests a greater emphasis on actions and events within Wikipedia articles, aligning with Korte et al.'s (2023) observation that Wikipedia frames CC primarily through events.

Auxiliary Verbs (AUX): No significant difference or variation in the use of auxiliary verbs was noticeable between the two sources. While a high proportion of auxiliary verbs can indicate the use of complex verb phrases, the presence of questions, or the use of non-present tenses, these features did not emerge as distinguishing factors in this analysis.

Adjectives (ADJ): The SPMs clearly contains a higher proportion of adjectives. This finding supports earlier observations that they aim for precision, as adjectives are characteristic of descriptive writing, providing additional detail about attributes. Interestingly, Wikipedia has gradually increased its use of adjectives over the years, indicating a move towards a similar style found in technical documents aiming for precision.

Adverbs (ADV): The variation in adverb usage is less than 1% for both SPMs and Wikipedia, making any meaningful interpretation difficult. While adverbs can contribute to precision by modifying verbs and adjectives to provide more detail, as a higher frequency might be expected in precise texts, they were not considered a strong distinguishing feature in this comparison due to the minimal variation.

Pronouns (PRON): Wikipedia consistently uses pronouns at a rate of 2%. In contrast, the IPCC appears to have actively reduced pronoun usage in its SPMs over time, nearly halving it from their first SPM in 1990 to the AR6 SPM in 2022. This reduction aligns with the expectation for technical texts to use fewer pronouns to ensure precision and maintain a more formal style, as a higher use of pronouns can lead to ambiguity.

Adpositions (ADP): No significant difference or variation in the use of adpositions (which include prepositions and postpositions) was noticeable. While a high frequency of adpositions is often associated with complex noun phrases and may indicate syntactic complexity, this was not a distinguishing factor between the analysed texts.

Coordinating Conjunctions (CCONJ): It might seem a reasonable hypothesis that a higher proportion of CCONJ in a document reflects a higher proportion of longer sentences in that document. However, in our corpus, an increase or decrease in CCONJ from one document to the other does not always match an increase or decrease in sentence length (compared to results in Section 4.1.6). Accordingly, the results suggest a more nuanced relationship between CCONJ and sentence length in these documents.

Subordinating Conjunctions (SCONJ): The IPCC appears to have deliberately reduced its use of SCONJ over time in its SPMs, halving their frequency compared to its initial AR. Since SCONJ indicate the presence of subordinate clauses and can be used to assess syntactic complexity, this reduction seems deliberate and suggests a move towards less syntactically complex sentence structures in later IPCC SPMs. On the other hand, Wikipedia showed no particular trend and remained at a level similar to that of the first AR.

4.1.9 TF-IDF scores

As mentioned in Section 3.3.1, for TF-IDF, documents were lemmatized and the vocabulary was limited to 2,000 terms, while the TF-IDF model was fit on the entire corpus. We discuss

here the results based on the 10 words with the highest TF-IDF per document; see Appendix G for the complete results.

While many terms shared between documents appeared in the top 10 TF-IDF results, we focused on the unique terms to gain the most insight into each document.

In **AR1_WG3_SPM**, we noticed the words “greenhouse”, “gas” and “emission”, which refer to greenhouse gas emissions, an important factor in climate change. There are also the terms “develop”, “country”, “energy” and “resource”. This could be a reference to countries using their resources to produce energy, which in turn generates greenhouse gases. It seems the first WG3 focuses on identifying the issues and their link to energy.

In **AR2_WG3_SPM**, the words “cost”, “economic”, “policy”, “damage” and “estimate” indicate a shift towards a more economic and social perspective on climate change. Rather than identifying the issues, it seems that AR2’s SPM focuses on quantifying the damage.

In **AR3_WG3_SPM**, we noticed the appearance of the term “mitigation”. This suggests that the focus is now on direct action, with terms such as “scenario”, “carbon”, “cost” and “reduction”.

AR4_WG3_SPM appeared to continue AR3 SPM’s focus on mitigation. Carbon is now referred to as “CO₂”, and “greenhouse gas” is abbreviated as “GHG”. The term “eq” refers to equivalent, which is often paired with “CO₂” as “CO₂-eq”,²⁰ which is a standard unit for measuring carbon footprint.

In **AR5_WG3_SPM**, the terms “evidence”, “medium” and “high” emerged. This reflects a focus on the certainty of findings and may suggest an attempt to address previous criticism regarding uncertainty about climate change or climate change denial.

The words “confidence” and “high” are particularly prominent in **AR6_WG3_SPM**. It follows AR5’s approach, which emphasises certainty, while also introducing “pathways” as mitigation strategies.

While the earliest version of **Wikipedia’s CCM article from 2005** uses terminology similar to that of the earlier ARs, such as “global”, “warming”, “carbon” and “emission”, it clearly places emphasis on events, with terms like “Kyoto”, “Protocol” and “2005”.

Wikipedia’s 2008 CCM article has a shift towards “energy” and “power” sources and the need to “reduce” them.

Wikipedia’s 2014 CCM article continued to focus on “energy”, but the emergence of the term “nuclear” is notable, as this technology was likely considered an emerging mitigation strategy at the time. While “2011” could refer to various events, it is difficult not to associate it with the Fukushima disaster of that year, especially when “nuclear” is also a key term. The article likely discusses the dangers of nuclear technology, a common topic in public debate.

Finally, **Wikipedia’s 2022 CCM article** features a classic list of words that reflects contemporary discourse on CCM. The appearance of the word “mitigation” in the list indicates a convergence with the language of the IPCC, suggesting that Wikipedia does, at least in part, reflect the scientific discourse.

According to these observations, while the IPCC adjusts its focus over the years, likely in response to evolving expectations, it consistently maintains a highly technical tone, using precise metrics and terminology. In contrast, Wikipedia, although it incorporates key IPCC terms, also reflects public discourse and contemporary events. However, the most recent

²⁰ In the corpus, these terms were preprocessed as 'co2', 'eq', and sometimes 'co2eq'. The additional formatting in this section is intended solely to support understanding of the terms.

iteration of the CCM article from 2022 aligns more closely than ever with the IPCC's language and framing, which is what we sometimes also observed in previous sections.

4.2 Modality Analysis

4.2.1 Verb and Adverbs

To reveal the trends in our corpus, we calculated the normalized frequency, calculated as the number of occurrences of modal verbs and adverbs per 1,000 words, excluding punctuation and stop words.

Frequencies of modal verbs and adverbs are available in Table 15. They were processed using Python code available in Appendix H.

Document	Total Modal Verb Freq.	Total Modal Adverb Freq.
AR1_WG3_SPM	27.093	0.774
AR2_WG3_SPM	31.153	1.038
AR3_WG3_SPM	24.520	0.570
AR4_WG3_SPM	16.819	0.247
AR5_WG3_SPM	12.132	1.415
AR6_WG3_SPM	15.983	0.349
Wiki_CCM_2005-06-26	11.704	None
Wiki_CCM_2008-05-07	21.900	0.448
Wiki_CCM_2014-09-02	18.519	0.497
Wiki_CCM_2022-06-13	25.544	0.643

Table 15: Total Modal Verb/Adverb Freq. per Text (normalized per 1000, rounded, 3 d.p.)

In Table 15, we observed that modal verbs are much more frequent than modal adverbs, but this is an expected result, as modal verbs are generally used more often than adverbs.

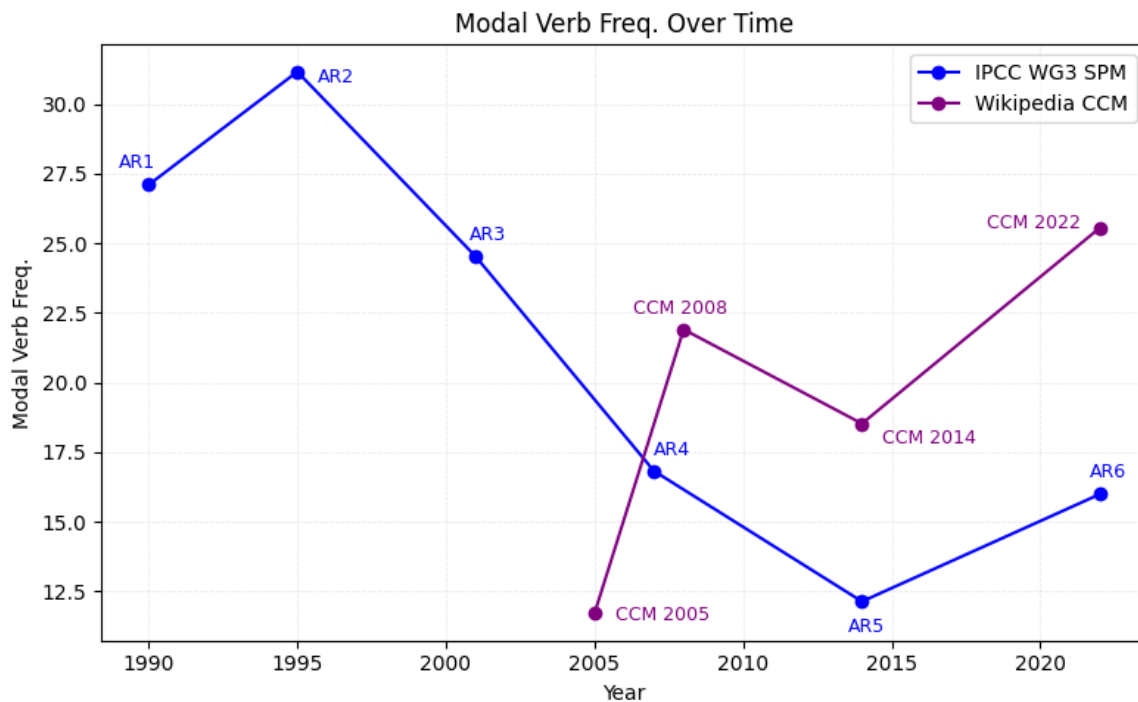


Figure 11: Total Modal Verb Frequency (per 1000) Over Time

In Figure 11, we noticed a decrease in the IPCC's usage of modal verbs over time. This decline could be attributed to several factors:

a) The earliest reports focused on predicting scenarios using modal verbs expressing possibility such as *may* or *could*. The TF-IDF results from Section 4.1.9 also support this claim, particularly the appearance of the word “estimate” in AR2.

b) The most recent reports are establishing facts that do not require modal verbs. The emergence of the word *evidence* also in Section 4.1.9 further supports this observation.

c) Specific guidelines that rely less on modal verbs have been established to guide writing teams. This is further confirmed by likelihood and confidence expressions defined in AR4 WG1's work (Le Treut & Somerville, 2007), which provides advice on how to address uncertainty in writing.

Regarding Wikipedia, except for 2005's article that is hard to assess, we notice that Figure 11's curve follows the IPCC's variation in modal verb usage over the three last reports, reflecting the IPCC's content. However, Wikipedia still generally uses more modal verbs. This could be linked to Wikipedia's focus on events and named entities, which makes it more likely to contain predictions related to them.

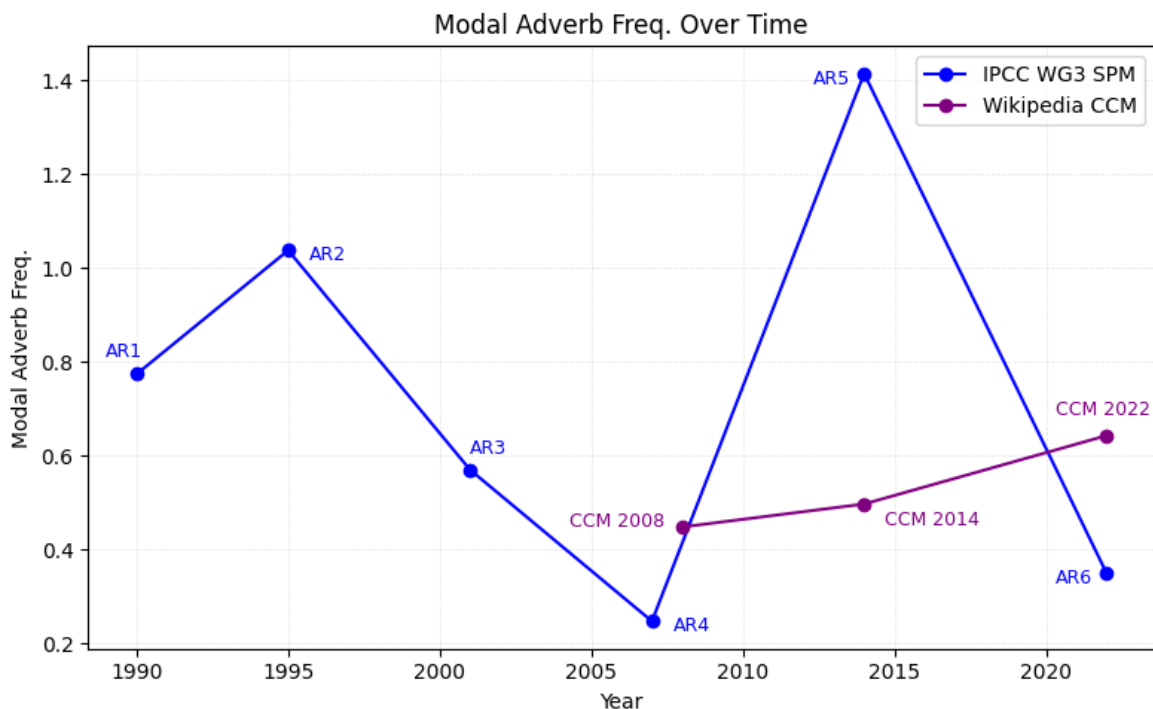


Figure 12: Total Modal Adverb Frequency (per 1000) Over Time

In Table 15, Wikipedia's 2005 CCM article contains no adverbs, likely due to its short length, and as such, we couldn't draw any conclusions from it. For the other Wikipedia articles, it was difficult to discern a trend with only three documents in Figure 12. However, the 2008 and 2022 articles show an adverb frequency that is closer to its SPM counterpart, which aligns with our earlier observation that the latest IPCC WG3 SPM content is closely reflected by Wikipedia's 2022 CCM.

We also observed a sharp spike in the 2014 SPM, which is almost certainly the result of a deliberate change in writing policy or a shift in focus towards climate change policy and controversies. We evaluated the individual usage of adverbs for this specific SPM, an

unusually frequent use of the adverb “likely” confirms a potential change in the writing policy. In Table 16, we focused on the frequency of the term “likely” exclusively.

Document	Term	Normalized Freq. 1000
AR1_WG3_SPM	likely	0.111
AR2_WG3_SPM	likely	0.173
AR3_WG3_SPM	likely	None
AR4_WG3_SPM	likely	None
AR5_WG3_SPM	likely	1.011
AR6_WG3_SPM	likely	0.279

Table 16: "Likely" Adverb Normalized Frequency per Text (per 1000, rounded, 3 d.p.)

This frequent use of the adverb “likely” was probably associated with the introduction of likelihood expressions within the IPCC framework, which will be examined in the next section below (4.2.2).

However, adverb frequency is relatively low, at around 1 or 2 occurrences per 1,000 words. Modal verbs are 10 to 20 times more frequent. Taken together, the impact of adverbs is minimal, either way, we still observe a downward trend in the use of modality consistent with the standardisation of IPCC reports and a shift towards more factual reports instead of speculation. However, the SPM from 2022 shows a slight rebound, possibly reflecting yet a new shift in communication strategy.

Ultimately, modality is best analysed in context, as a qualitative approach can capture not only the frequency but also the precise function of modality. Some modal expressions also appear near negation structures, which can reverse their meaning. It would have been valuable to examine this interaction within the corpus, but given the time constraints, we adopted a purely quantitative approach, aiming to provide an initial overview of the distribution of modal verbs and adverbs across the texts as a basis for comparison.

4.2.2 Likelihood and Confidence

Likelihood and confidence expressions are a specific category of standardized language introduced by the IPCC to convey degrees of certainty through quantified modalities. The original tables listing the expressions of likelihood and confidence are available in Section 3.3.2.

Document	Total Likelihood Freq.	Total Confidence Freq.
AR1_WG3_SPM	1.106	None
AR2_WG3_SPM	2.423	None
AR3_WG3_SPM	0.950	None
AR4_WG3_SPM	0.495	None
AR5_WG3_SPM	5.358	4.145
AR6_WG3_SPM	0.419	19.542
Wiki_CCM_2005-06-26	2.601	None
Wiki_CCM_2008-05-07	1.120	None
Wiki_CCM_2014-09-02	1.243	0.124
Wiki_CCM_2022-06-13	0.735	None

Table 17: Total Likelihood and Confidence Frequencies per Text (rounded, 3 d.p.)

While this type of expression was first introduced in AR4 (Le Treut & Somerville, 2007), its usage became more prominent in AR5 for expressing likelihood and in AR6 for expressing confidence, according to results in Table 17. These highly codified expressions, each with a precise and standardized meaning, are not reflected in Wikipedia articles, which is understandable, as such expressions in the IPCC context are supported by formal definitions, tables, and quantified values for each level of modality, as seen previously in the methodology section (3.3.2).

Over time, modal verbs and adverbs have been largely replaced by standardized expressions of modality in the SPMs, offering clearly defined meanings. In contrast, Wikipedia maintains a more conventional approach to modality. While the IPCC system offers a high level of precision appropriate for its role in informing policymakers, such precision is less relevant for Wikipedia, which serves a broader general audience.

4.3 Semantic Similarity Comparison

As mentioned in Section 3.3.3, we did not process all documents for this analysis, ignoring AR1 and AR2 as we needed to process pairs, and there is no Wikipedia counterpart to AR1 and AR2’s content in our setup.

Semantic similarity was computed using Python code available in Appendix I. Recall that, in essence, for each SPM sentence, the maximum cosine similarity with any sentence in its corresponding Wikipedia document was computed, based on Sentence BERT embeddings (see 3.3.3 for details). Average similarity is reported to Table 18.

Document A	Document B	Doc. A Sentences Count	Document B Sentences Count	Mean Similarity	Median Similarity
AR3_WG3_SPM	Wiki_CCM_2005-06-26	343	48	0.424	0.428
AR4_WG3_SPM	Wiki_CCM_2008-05-07	558	263	0.473	0.483
AR5_WG3_SPM	Wiki_CCM_2014-09-02	580	583	0.547	0.561
AR6_WG3_SPM	Wiki_CCM_2022-06-13	896	786	0.553	0.572

Table 18: Average and Median Document Similarity Scores (rounded, 3 d.p.)

In Table 18, the analysis returned values that show a consistent increase in both the mean and median sentence-level similarity scores over time. The mean rose from 0.424 in the earliest document to 0.553 in the latest. A similar trend is observable for the median, which increased from 0.482 to 0.572. This indicates an improving degree of similarity, suggesting that Wikipedia CCM articles are increasingly reflecting the IPCC WG3 SPMs. The closeness of the mean and median suggests there are few outliers or off-topic content in Wikipedia.

While these results are consistent with findings from previous sections, which show that the 2022 Wikipedia CCM article aligns well with its IPCC WG3 SPM counterpart, the increase in similarity observed in this analysis remains modest, indicating that some degree of difference with the IPCC reports persists.

4.4 Topic Modelling

Topic modelling results are available in Table 19. Our implementation is described in Section 3.3.4 and the related Python code demonstrating the use of BERTopic is available in Appendix J.

Document	Topic	Name	Representation
AR1_WG3_SPM	-1	-1_emission_gas_change_greenhouse	['emission', 'gas', 'change', 'greenhouse', 'country', 'climate', 'develop', 'energy', 'use', 'resource']
AR2_WG3_SPM	1	1_emission_high_confidence_mitigation	['emission', 'high', 'confidence', 'mitigation', 'cost', 'ghg', 'global', 'change', 'energy', 'climate']
AR3_WG3_SPM	0	0_emission_energy_carbon_climate	['emission', 'energy', 'carbon', 'climate', 'change', 'global', 'gas', 'mitigation', 'cost', 'reduce']
AR4_WG3_SPM	0	0_emission_energy_carbon_climate	['emission', 'energy', 'carbon', 'climate', 'change', 'global', 'gas', 'mitigation', 'cost', 'reduce']
AR5_WG3_SPM	1	1_emission_high_confidence_mitigation	['emission', 'high', 'confidence', 'mitigation', 'cost', 'ghg', 'global', 'change', 'energy', 'climate']
AR6_WG3_SPM	1	1_emission_high_confidence_mitigation	['emission', 'high', 'confidence', 'mitigation', 'cost', 'ghg', 'global', 'change', 'energy', 'climate']
Wiki_CCM_2005-06-26	0	0_emission_energy_carbon_climate	['emission', 'energy', 'carbon', 'climate', 'change', 'global', 'gas', 'mitigation', 'cost', 'reduce']
Wiki_CCM_2008-05-07	0	0_emission_energy_carbon_climate	['emission', 'energy', 'carbon', 'climate', 'change', 'global', 'gas', 'mitigation', 'cost', 'reduce']
Wiki_CCM_2014-09-02	0	0_emission_energy_carbon_climate	['emission', 'energy', 'carbon', 'climate', 'change', 'global', 'gas', 'mitigation', 'cost', 'reduce']
Wiki_CCM_2022-06-13	0	0_emission_energy_carbon_climate	['emission', 'energy', 'carbon', 'climate', 'change', 'global', 'gas', 'mitigation', 'cost', 'reduce']

Table 19: Topic Distribution for Each Text According to BERTopic

Due to the small and thematically focused nature of the corpus, BERTopic identified only 2 topics, the third one (-1) being the outlier, a category for off-topic documents that couldn't be classified.

In Table 19, we noticed an important keyword overlap between topic 0 and 1, this means that topics 0 and 1 are not distinct topics but only variations of CCM discussion.

Topic 0, “emission, energy, carbon, climate” is the largest topic and contains all Wikipedia CCM articles as well as 2 SPM. This may suggest that Wikipedia articles and some IPCC SPMs share a common approach to CCM. While Wikipedia may have differed from the IPCC in its framing of events, it was never entirely off-topic.

Topic 1, “emission, high, confidence, mitigation”, contains three SPMs. This topic is likely formed around the very specific way in which SPMs 5 and 6 assert the certainty of findings, as previously discussed in Sections 3.3.2 and 4.2.2. Specific words such as “high” and “confidence” reflect the language used in likelihood and confidence expressions which were predominantly employed in SPMs 5 and 6. The classification of SPM 2 under Topic 1 is

surprising. However, it correlates with the fact that it contains the highest number of modal verbs and adverbs, as noted in Section 4.1.2. Although modal verbs do not appear in the current topic signature on Table 19, they are most likely ranked lower.

Topic -1, the outlier “emission, gas, change, greenhouse”, includes only the earliest of the reports. This is interesting, as the first report has a foundational nature, aiming to establish an initial understanding of greenhouse gases and their link to development resources and energy, while the next reports focused on mitigation strategies.

Finally, while topic modelling did identify trends within the corpus, it did not reveal anything that was not previously noticed. Given the small size of the corpus, this outcome was expected, and any unexpected findings would have been difficult to interpret due to the margin of error implied by such a limited corpus.

4.5 Sentiment and Emotions Analysis

As described in 3.3.5, we annotated sentiment using both VADER (a rule and lexicon-based tool) and a transformer-based model, and emotion using only a transformer-based model. A single Python script, available in Appendix K, runs all workflows simultaneously. Results are available in Table 20 for VADER sentiment analysis, Table 21 for transformer-based sentiment analysis, and Table 22 for emotion detection.

4.5.1 Sentiment Analysis with VADER

Table 20 reports the sentiment analysis results from VADER. VADER’s compound score is a weighted metric that reflects the intensity and polarity of sentiment, ranging from -1 (negative) to +1 (positive). Values around 0 mean neutral sentiment.

Document	Total Paragraphs	Avg. VADER Compound	VADER Positive (%)	VADER Neutral (%)	VADER Negative (%)
AR1_WG3_SPM	79	0.681	87.3	0.0	12.7
AR2_WG3_SPM	205	0.171	54.6	17.1	28.3
AR3_WG3_SPM	72	0.366	70.8	9.7	19.4
AR4_WG3_SPM	167	0.299	58.1	19.2	22.8
AR5_WG3_SPM	112	0.386	74.1	1.8	24.1
AR6_WG3_SPM	108	0.534	80.6	1.9	17.6
Wiki_CCM_2005-06-26	12	0.381	83.3	0.0	16.7
Wiki_CCM_2008-05-07	92	0.427	76.1	7.6	16.3
Wiki_CCM_2014-09-02	118	0.304	54.2	33.9	11.9
Wiki_CCM_2022-06-13	126	0.405	65.9	20.6	13.5

Table 20: VADER Sentiment Analysis Results (normalized 3 d.p. for Compound, 1 d.p. for %)

From Table 20, we observed that the sentiment across all documents is predominantly positive, with no clear temporal trend.

Given that a neutral tone is typically expected in scientific writing, those results were somewhat unexpected. To confirm this trend, we needed to compare them against a transformer-based sentiment analysis for a more robust evaluation.

4.5.2 Transformer-based Sentiment Analysis

In contrast to VADER, RoBERTa results in Table 21 indicate that neutrality dominates across the corpus, with a rising trend of positivity in the SPM. Although the two methods differ in their assessments of positivity and neutrality, they align in their evaluation of negative sentiment.

Document	RoBERTa Positive (%)	RoBERTa Neutral (%)	RoBERTa Negative (%)
AR1_WG3_SPM.txt	29.1	65.8	5.1
AR2_WG3_SPM.txt	24.4	67.3	8.3
AR3_WG3_SPM.txt	47.2	51.4	1.4
AR4_WG3_SPM.txt	36.5	58.7	4.8
AR5_WG3_SPM.txt	50.0	39.3	10.7
AR6_WG3_SPM.txt	57.4	35.2	7.4
Wiki_CCM_2005-06-26.txt	8.3	75.0	16.7
Wiki_CCM_2008-05-07.txt	33.7	57.6	8.7
Wiki_CCM_2014-09-02.txt	18.6	76.3	5.1
Wiki_CCM_2022-06-13.txt	28.6	63.5	7.9

Table 21: RoBERTa Sentiment Analysis Results (rounded, 1 d.p.)

In summary, in terms of sentiment analysis, the IPCC SPM and Wikipedia articles are very similar.

4.5.3 Emotion Detection

Document	Anger (%)	Disgust (%)	Fear (%)	Joy (%)	Neutral (%)	Sadness (%)	Surprise (%)
AR1_WG3_SPM	1.3	0.0	5.1	1.3	92.4	0.0	0.0
AR2_WG3_SPM	0.0	0.5	2.4	0.5	96.1	0.5	0.0
AR3_WG3_SPM	0.0	0.0	0.0	0.0	100.0	0.0	0.0
AR4_WG3_SPM	0.0	2.4	1.8	0.0	95.8	0.0	0.0
AR5_WG3_SPM	0.0	0.0	0.9	0.0	99.1	0.0	0.0
AR6_WG3_SPM	0.0	0.0	1.9	0.0	98.1	0.0	0.0
Wiki_CCM_2005-06-26	0.0	8.3	16.7	0.0	75.0	0.0	0.0
Wiki_CCM_2008-05-07	4.3	0.0	10.9	2.2	81.5	1.1	0.0
Wiki_CCM_2014-09-02	1.7	0.8	8.5	0.8	86.4	1.7	0.0
Wiki_CCM_2022-06-13	2.4	3.2	4.8	0.0	89.7	0.0	0.0

Table 22: RoBERTa Emotion Detection Results (rounded, 1 d.p.)

Emotion detection results in Table 22 indicate a predominance of the neutral emotion, consistent with previous sentiment analyses. AR3 stands out for having all paragraphs classified as neutral, Wikipedia CCM articles appear slightly less neutral and exhibit a greater degree of fear.

Interestingly, this fear aligns with Korte et al.'s (2023) argument that general media often frames climate change as a “crisis”. However, we do not observe any notably pessimistic tone in the Wikipedia CCM articles that would support the findings of Barkemeyer et al. (2016).

Overall, these findings support the hypothesis that the IPCC maintains an objectively neutral tone in its reports, and that Wikipedia largely succeeds in reflecting this tone.

4.6 Named Entity Recognition Analysis

As described in 3.3.6, both spaCy NER and a CC specific NER were used. A single Python script was run to count spaCy entities, which were already processed and stored in spaCy Doc objects (see 3.2.2), and to execute the transformer-based CC NER. The script is available in Appendix L.

In this section, we evaluate NER results in the context of climate change mitigation, while evaluating Korte et al.'s (2023) hypothesis that Wikipedia frames climate change more in terms of events and personalities than the IPCC does.

Accordingly, some categories deemed irrelevant or supported by limited data have been excluded from Table 23 and Table 24. However, complete tables containing all the results are available in Appendix M for spaCy's NER and Appendix N for CC NER.

This section is divided into three subsections. The first subsection (4.6.1) is an assessment of spaCy's NER results. The second subsection (4.6.2) assess results from a dedicated CC NER transformer-based model, and the third subsection (4.6.3) synthesises the results.

4.6.1 SpaCy's NER Results

For spaCy, we retrieved dates or periods (DATE), named events (EVENT), geopolitical entities (GPE), nationalities and religious/political groups (NORP), organizations (ORG) and persons or fictional characters (PERSON), as shown in Table 23.

SpaCy NER categories and definitions are available in Section 2.6.3 (Table 1, p.22).

Document	DATE	EVENT	GPE	NORP	ORG	PERSON
AR1_WG3_SPM	7.409	0.442	1.106	0.221	15.371	0.663
AR2_WG3_SPM	4.500	0.173	0.173	0.346	5.365	0.000
AR3_WG3_SPM	11.785	0.190	1.901	0.950	17.107	1.711
AR4_WG3_SPM	14.222	0.247	1.113	0.495	22.755	2.473
AR5_WG3_SPM	24.669	0.404	1.921	0.607	19.412	2.831
AR6_WG3_SPM	17.797	0.140	0.628	1.047	22.334	0.768
Wiki_CCM_2005-06-26	33.810	2.601	42.913	2.601	45.514	5.202
Wiki_CCM_2008-05-07	20.376	1.120	16.346	1.567	31.572	11.196
Wiki_CCM_2014-09-02	30.698	0.746	13.547	2.237	41.511	15.287
Wiki_CCM_2022-06-13	20.583	0.643	11.578	3.124	22.972	5.881

Table 23: spaCy Relevant NER Categories (frequencies normalized per 1000, rounded, 3 d.p.)

Results in Table 23 indicated that dates occur more frequently in Wikipedia articles, although the IPCC also relies on them considerably. This difference, however, is not significant enough to draw firm conclusions, even if a higher frequency of dates combined with other categories such as events could indicate a tendency to frame content around specific moments in time.

Regarding events, there is also a higher count in the earlier Wikipedia articles, but the 2014 and 2022 articles reduced this by half, bringing them closer to the IPCC's usage, though still notably different. Both statistics support Korte et al.'s (2023) claim.

In Wikipedia, the results showed a very high count of geopolitical entities, up to 42 times higher for the 2005 article compared to the first 1990 SPM. However, that article is very short and should not be heavily relied upon. The difference remains substantial in other cases,

such as the 2022 Wikipedia article which contains more than 18 times the number of geopolitical entity occurrences compared to its SPM counterpart.

The remaining categories, which include nationalities or religious and political groups, organisations, and persons, also appeared with much higher frequency in Wikipedia. This provided strong evidence supporting Korte et al.'s (2023) claims, confirming that Wikipedia does indeed frame climate change differently from the IPCC in the documents analysed. That said, we observed a downward trend over time, with the most recent Wikipedia article showing frequencies closer to its latest SPM counterpart, although significant differences remain in some categories. However, the frequency of organisations is nearly equal in the 2022 article and its SPM counterpart

4.6.2 Climate Change NER Results

While comparing named entities across the different documents was informative, examining them through the lens of climate change proved to be more insightful.

CC NER categories and definitions are available in Section 3.3.6 (Table 7, p.38-39), while complete CC NER results are available in Appendix N.

Note: Original class names were prefixed with 'climate-'; this prefix has been omitted in the tables for conciseness.

Document	Impacts	Mitigations	Greenhouse Gases	Organizations	Problem Origins	Hazards
AR1_WG3_SPM	1.106	40.142	12.275	19.020	42.243	10.727
AR2_WG3_SPM	10.211	31.499	3.461	7.442	16.961	8.827
AR3_WG3_SPM	1.331	50.561	13.496	11.595	37.635	4.372
AR4_WG3_SPM	0.989	57.878	22.384	10.017	49.963	4.081
AR5_WG3_SPM	0.809	46.608	20.928	9.099	33.970	3.235
AR6_WG3_SPM	0.907	57.300	23.032	6.281	38.945	2.931
Wiki_CCM_2005-06-26	2.601	71.521	6.502	55.917	9.103	29.909
Wiki_CCM_2008-05-07	2.463	74.116	20.376	55.083	38.065	11.420
Wiki_CCM_2014-09-02	1.740	71.837	21.999	47.974	39.150	6.587
Wiki_CCM_2022-06-13	1.838	77.001	7.535	20.674	43.646	11.853

Table 24: Relevant CC NER Categories (per 1000, rounded, 3 d.p.)

The results in Table 24 show that, for **climate-impacts** (effects of hazards to humans), AR2 is a notable outlier: It contains up to ten times more climate-impacts entities than the other reports. Consequently, AR2 is much lower on other categories than other ARs. This aligns with the theme of its own report, which focuses on exploring how climate change could affect society. Wikipedia tends to mention these impacts slightly more often than all AR but AR2, which suggests that it places greater emphasis on the consequences of inaction and the urgency of action, thereby framing climate change as a crisis.

For **climate-mitigations** (activities to reduce the impact of CC), which is the focus of our corpus, both sources dedicate a portion of their content to the topic. However, Wikipedia showed a higher frequency. This may be because Wikipedia presents information in a more direct manner, whereas the IPCC includes studies and focuses on feasibility.

Except for AR2, **climate-greenhouse-gases** (mention of gases that impacts the atmosphere) are consistently mentioned in the SPMs, but their presence is more variable on Wikipedia.

This was difficult to assess, but it is possible that the most recent Wikipedia article on climate change mitigation has become less technical and no longer refers explicitly to the names of specific gases, to attract a more general audience.

Climate-organization (organizations with CC interests) are mentioned far more frequently in Wikipedia articles than in the SPMs. The SPMs are more moderate and show a decreasing trend in this category over time, a pattern that Wikipedia also follows while still significantly higher than its counterpart, the 2022 Wikipedia article shows a significant reduction in frequency, indicating a closer alignment with the original IPCC material.

Both the IPCC and Wikipedia show strong interest in the **climate-problem-origins** category (Issues that results in CC), only the 2005 document shows a significantly lower frequency, which may be due to the short length of the article. In this aspect, both sources are equal.

Climate-hazards are a major focus of the 2005 article, indicating an emphasis on the dangers themselves rather than on their underlying causes, as discussed previously. Wikipedia shows a greater tendency to frame climate change through the lens of hazards, which once again aligns with the “crisis” narrative often found in public media, as noted by Korte et al. (2023). In contrast, the SPMs have adopted a more measured tone over time, particularly in the most recent report.

Overall, the climate change NER analysis shows that Wikipedia CCM articles successfully cover the key thematic areas of climate change mitigation, although the emphasis, level of technical detail, and framing can differ. In contrast, the IPCC Working Group III SPM maintains a scientifically assessed presentation, as is expected of a Summary for Policymakers.

4.6.3 Results Synthesis

Combined NER analyses revealed distinct styles between the sources. Wikipedia’s CCM articles showed a high frequency of PERSON, EVENT, GPE, and NORP entities, supporting Korte et al.’s (2023) theory of an event-driven framing. This tendency is also reflected in the emphasis on climate hazards identified by the CC NER model. In contrast, the IPCC WG3 SPMs maintain an institutional tone with a consistently high level of technical detail, particularly in reference to greenhouse gases. While the two sources differed in framing, both managed to address the core themes of CCM. Notably, Wikipedia has shown increasing alignment with the IPCC SPMs over time. This is consistent with our earlier analyses, which also indicated that the most recent Wikipedia article is becoming more closely aligned with the latest SPM.

4.7 Section Summary: Results and Discussion

To summarize Section 4, the presented analyses revealed a nuanced relationship between the IPCC WG3 SPMs and Wikipedia’s CCM articles. While stylistic, lexicometric, and readability metrics initially highlighted distinct approaches tailored to different audiences, a clear trend of convergence emerged over time. Semantic similarity increased, and Wikipedia’s thematic content and framing, particularly in its most recent article revision, have become more closely aligned with those of the IPCC. Modality and NER analysis further highlighted contrasting communicative strategies: the IPCC tends to employ formal, codified language, while Wikipedia reflects a broader, more event-driven public discourse. However, these differences suggest variations in emphasis and simplification for a general audience, rather than a bias or distortion of the core scientific message presented by the IPCC.

Conclusion

In this study, we conducted a comparative analysis of the Intergovernmental Panel on Climate Change (IPCC) Working Group III (WG3) Summaries for Policymakers (SPM), which focus on climate change mitigation, and thematically and temporally corresponding versions of Wikipedia's "Climate Change Mitigation" (CCM) article. The primary objective was to employ Natural Language Processing (NLP) techniques and tools to identify potential differences or biases in how Wikipedia presents CCM compared to the IPCC, as this is a key scientific topic often targeted by disinformation.

Using an array of NLP techniques including lexicometry, stylistic analysis, readability assessments, modality analysis, semantic similarity, topic modelling, sentiment and emotion detection, and both general and climate-specific Named Entity Recognition (NER), the study conducted a detailed comparison between six IPCC WG3 SPM (from AR1 to AR6) and four temporally aligned Wikipedia CCM articles from 2005, 2008, 2014, and 2022. This revealed a complex and evolving relationship between the two sources.

The study began with **lexicometric, stylistic, and readability analyses**. These techniques highlighted fundamental differences, likely influenced by the intended audiences of each source. The IPCC SPMs consistently demonstrated a higher level of technicality, greater lexical density, and college-level readability, all of which are characteristic of scientific reports aimed at policymakers. In contrast, Wikipedia was initially simpler but over time exhibited a clear progression towards greater technical complexity, increased lexical richness, and improved readability. These developments suggest a gradual stylistic convergence with the IPCC. A TF-IDF analysis illustrated the IPCC's shifting focus from problem identification to economic impacts, mitigation strategies, and controlled expressions of certainty. Wikipedia, while adopting much of the IPCC's terminology, also reflected elements of public discourse and references to contemporary events.

Modality analysis revealed a clear shift in the IPCC's language, from the use of general modal verbs in earlier reports to the adoption of highly standardised expressions of likelihood and confidence in later assessments, particularly in AR5 and AR6. This transition reflects a deliberate effort to communicate uncertainty with greater precision. Wikipedia did not adopt these codified forms and instead showed an increase in the use of modal verbs and adverbs over time, consistent with its aim to inform a more general audience.

Semantic similarity scores, calculated using Sentence-BERT and cosine similarity, revealed a slow yet gradual and consistent convergence between the IPCC SPMs and their corresponding Wikipedia articles. The average similarity score rose from 0.424 for the AR3 and 2005 pairing to 0.553 for the AR6 and 2022 pairing. This suggests that Wikipedia is increasingly reflecting the content of the IPCC's findings on CCM, although notable differences remain due to the distinct purposes of each source.

Although the corpus was limited in size, **topic modelling** successfully grouped some of the IPCC SPMs and Wikipedia articles together based on shared mitigation themes. In contrast, the later IPCC reports formed distinct clusters, characterised by their modal expressions of likelihood and confidence. The first Working Group III SPM was identified as an outlier, due to its distinct focus on identifying climate change (CC) issues rather than presenting mitigation strategies.

Sentiment and emotion analyses indicated that both sources generally maintained a predominantly neutral tone, as expected in scientific writing. While VADER initially suggested a positive sentiment, transformer-based models offered a more nuanced interpretation,

revealing overall neutrality with an upward trend in positivity within the IPCC SPMs. Wikipedia articles showed a slightly higher occurrence of “fear”, which aligns with Korte et al. (2023) findings on media framing CC as a “crisis”.

Named Entity Recognition (NER) provided evidence of contrasting framing strategies between the two sources. A first NER analysis using spaCy showed that Wikipedia’s articles more frequently included entities related to events, persons, and geopolitical locations. This supports Korte et al.’s (2023) hypothesis that Wikipedia adopts a more event and politically driven narrative style. A second NER analysis was then conducted using a specialised CC NER model. This analysis revealed that, while both sources addressed core CCM themes, Wikipedia tended to place greater emphasis on hazards, impacts, and organisations. The IPCC maintained a more consistent and technical focus, particularly regarding greenhouse gases. Finally, both NER analyses indicated a clear trend towards convergence, with the most recent Wikipedia CCM article from 2022 showing entity frequencies in several categories that are increasingly aligned with those in the IPCC 2022 SPM.

To answer the **research question** posed in the introduction, specifically whether there are differences or biases in Wikipedia’s portrayal of the IPCC reports, the findings from the analysis of Wikipedia’s CCM articles and the WG3 SPMs indicate that Wikipedia does not exhibit bias in the form of deliberate distortion or omission. Instead, it shows differences in framing, emphasis, and the simplification of technical content, which are consistent with its function as a publicly accessible resource intended for a general audience. While Wikipedia is influenced by public discourse, events, and an inclination towards a “crisis” narrative, it demonstrates a clear effort to align with and accurately convey the IPCC’s core scientific findings on CCM.

While it aimed to be comprehensive, this study has some significant **limitations**. The size of the corpus restricted the extent of some Natural Language Processing analyses, particularly topic modelling. The focus on WG3 SPMs and multiple revisions of a single Wikipedia article means that the findings may not be generalisable to all IPCC content or to the broader scope of Wikipedia’s climate change portal. Finally, the dynamic nature of Wikipedia means that the 2022 snapshot analysed in this study is already outdated.

Considering these limitations, **future research** could expand the corpus to include all IPCC working groups as well as a broader selection of related Wikipedia articles. While this study focused on English due to its prominence on Wikipedia, a cross-linguistic analysis would offer valuable insights into how CCM is represented across different languages. Although this research examined changes over time, a more extensive longitudinal study capturing all article revisions would provide a deeper understanding of content evolution. Additionally, the application of other NLP methods, such as Aspect-Based Sentiment Analysis or the use of cross-encoders for semantic similarity, could enrich the findings. A greater emphasis on qualitative approaches could also complement the quantitative analyses, offering further insight.

In conclusion, this study demonstrates that although the IPCC and Wikipedia serve different audiences, Wikipedia’s coverage of climate change mitigation is not marked by significant bias against IPCC findings. It rather reflects a sustained effort to make complex scientific information accessible to the public, with growing alignment to IPCC assessments and an increasingly important role in supporting the public understanding of one of the most critical challenges for humankind.

References

- Aarsen, T. (2024, January 5). *Sentence-transformers/all-MiniLM-L6-v2* · Hugging Face. <https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2>
- Adobe®. (2006). *PDF Reference sixth edition*.
https://web.archive.org/web/20081001170454/https://www.adobe.com/devnet/acrobat/pdfs/pdf_reference_1-7.pdf
- Anthony, L. (2005). AntConc: Design and development of a freeware corpus analysis toolkit for the technical writing classroom. *IPCC 2005. Proceedings. International Professional Communication Conference, 2005.*, 729–737.
<https://doi.org/10.1109/IPCC.2005.1494244>
- Auer, C., Lysak, M., Nassar, A., Dolfi, M., Livathinos, N., Vagenas, P., Ramis, C. B., Omenetti, M., Lindlbauer, F., Dinkla, K., Mishra, L., Kim, Y., Gupta, S., Lima, R. T. de, Weber, V., Morin, L., Meijer, I., Kuropiatnyk, V., & Staar, P. W. J. (2024). *Docling Technical Report* (No. arXiv:2408.09869). arXiv.
<https://doi.org/10.48550/arXiv.2408.09869>
- Barbieri, F., Camacho-Collados, J., Espinosa, A., Luis, & Neves, L. (2024). *Cardiffnlp/twitter-roberta-base-sentiment* · Hugging Face.
<https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment>
- Barkemeyer, R., Dessai, S., Monge-Sanz, B., Renzi, B. G., & Napolitano, G. (2016). Linguistic analysis of IPCC summaries for policymakers and associated coverage. *Nature Climate Change*, 6(3), 311–316.
<https://doi.org/10.1038/nclimate2824>
- Beutler, W. (2019). Paid With Interest: COI Editing and its Discontents. *Wikipedia @ 20*. <https://wikipedia20.mitpress.mit.edu/pub/kmwtdhw/release/4>
- Bhattacharjee, B., Trivedi, A., Muraoka, M., Ramasubramanian, M., Udagawa, T., Gurung, I., Pantha, N., Zhang, R., Dandala, B., Ramachandran, R., Maskey, M., Bugbee, K., Little, M., Fancher, E., Gerasimov, I., Mehrabian, A., Sanders, L., Costes, S., Blanco-Cuaresma, S., ... Lee, T. (2024). *INDUS: Effective and Efficient Language Models for Scientific Applications* (No. arXiv:2405.10725). arXiv. <https://doi.org/10.48550/arXiv.2405.10725>
- Biros, C., & Peynaud, C. (2019). *Disseminating climate change knowledge. Representation of the International Panel on Climate Change in three types of specialized discourse*. <https://doi.org/10.1285/i22390359V29P179>
- Biros, C., Rossi, C., & Talbot, A. (2021). *Corpus GIEC* [Corpus]. ORTOLANG.
<https://hdl.handle.net/11403/corpus-giec/v1>
- Bounegru, L., De Pryck, K., Venturini, T., & Mauri, M. (2020). “We only have 12 years”: YouTube and the IPCC report on global warming of 1.5°C. *First Monday*. <https://doi.org/10.5210/fm.v25i2.10112>

- Canonical Ltd. (2024). *Ubuntu – Package Search Results*.
<https://packages.ubuntu.com/search?lang=en&suite=all&searchon=names&keywords=curl>
- Ceylan, C. (2022). *Application of Natural Language Processing to Unstructured Data: A Case Study of Climate Change* [Thesis, Massachusetts Institute of Technology]. <https://dspace.mit.edu/handle/1721.1/144647>
- Cohen, N. (2009, August 24). Wikipedia to Limit Changes to Articles on People. *The New York Times*.
<https://www.nytimes.com/2009/08/25/technology/internet/25wikipedia.html>
- De Pryck, K., & Hulme, M. (Eds.). (2022). *A Critical Assessment of the Intergovernmental Panel on Climate Change*. Cambridge University Press.
<https://doi.org/10.1017/9781009082099>
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* (No. arXiv:1810.04805). arXiv. <https://doi.org/10.48550/arXiv.1810.04805>
- Duran, N. (2024, May 21). *Nicolauduran45/specter-climate-change-NER · Hugging Face*. <https://huggingface.co/nicolauduran45/specter-climate-change-NER>
- Explosion. (2025). *spaCy Models Documentation*. English. <https://spacy.io/models/en>
- Fenniak, M., & Martin, T. (2024). *pypdf: A pure-python PDF library capable of splitting, merging, cropping, and transforming PDF files* (Version 5.1.0) [Python; OS Independent].
- Flesch, R. (1948). A new readability yardstick. *Journal of Applied Psychology*, 32(3), 221–233. <https://doi.org/10.1037/h0057532>
- Fleury, S., & Zimina, M. (2014). Trameur: A Framework for Annotated Text Corpora Exploration. In L. Tounsi & R. Rak (Eds.), *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations* (pp. 57–61). Dublin City University and Association for Computational Linguistics. <https://aclanthology.org/C14-2013/>
- Gallo, I., Binaghi, E., Carullo, M., & Lamberti, N. (2008). Named Entity Recognition by Neural Sliding Window. *2008 The Eighth IAPR International Workshop on Document Analysis Systems*, 567–573. <https://doi.org/10.1109/DAS.2008.13>
- Grootendorst, M. (2022). *BERTopic: Neural topic modeling with a class-based TF-IDF procedure* (No. arXiv:2203.05794). arXiv.
<https://doi.org/10.48550/arXiv.2203.05794>
- Halfaker, A., Geiger, R. S., Morgan, J. T., & Riedl, J. (2013). The Rise and Decline of an Open Collaboration System: How Wikipedia's Reaction to Popularity Is Causing Its Decline. *American Behavioral Scientist*, 57(5), 664–688.
<https://doi.org/10.1177/0002764212469365>
- Halliday, M. A. K. (1989). *Spoken and written language*. Oxford University Press.

- Han, Y., Ceross, A., & Bergmann, J. H. M. (2024). *The Use of Readability Metrics in Legal Text: A Systematic Literature Review* (No. arXiv:2411.09497). arXiv. <https://doi.org/10.48550/arXiv.2411.09497>
- Härdle, W., & Chen, C. (2016). *Probabilistic Topic Models in Natural Language Processing*.
- Hart, R. P. (1984). Systematic analysis of political discourse: The development of DICTION. *Political Communication Yearbook*, 1, 97–134.
- Hartmann, J. (2022). *J-hartmann/emotion-english-distilroberta-base · Hugging Face*. <https://huggingface.co/j-hartmann/emotion-english-distilroberta-base>
- Hearst, M. A. (1997). Text Tiling: Segmenting Text into Multi-paragraph Subtopic Passages. *Computational Linguistics*, 23(1), 33–64.
- Heiden, S., Magué, J.-P., & Pincemin, B. (2010). *TXM: Une plateforme logicielle open-source pour la textométrie - conception et développement*. 13.
- Herrando-Pérez, S., Bradshaw, C. J. A., Lewandowsky, S., & Vieites, D. R. (2019). Statistical Language Backs Conservatism in Climate-Change Assessments. *BioScience*, 69(3), 209–219. <https://doi.org/10.1093/biosci/biz004>
- Hickson, I., & Hyatt, D. (2008). *HTML 5*. <https://www.w3.org/TR/2008/WD-html5-20080122/#references>
- Huddleston, R., & Pullum, G. K. (2002). *The Cambridge Grammar of the English Language*. Cambridge University Press.
- Hutto, C., & Gilbert, E. (2014). VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. *Proceedings of the International AAAI Conference on Web and Social Media*, 8(1), Article 1. <https://doi.org/10.1609/icwsm.v8i1.14550>
- Hyperbase. (2024). *Hyperbase.unice.fr*. <https://hyperbase.unice.fr/>
- IBM-research/Climate-Change-NER · Datasets at Hugging Face*. (2024, October 11). <https://huggingface.co/datasets/ibm-research/Climate-Change-NER>
- Iezzi, D. F., Mayaffre, D., & Misuraca, M. (Eds.). (2020). *Text Analytics: Advances and Challenges*. Springer International Publishing. <https://doi.org/10.1007/978-3-030-52680-1>
- IPCC. (2018). *Special Report: Global Warming of 1.5 °C (SR15)*. <https://www.ipcc.ch/sr15/>
- IPCC. (2024a). Procedures—IPCC. *ipcc.Ch*. <https://www.ipcc.ch/documentation/procedures/>
- IPCC. (2024b). *TFI — IPCC*. <https://www.ipcc.ch/working-group/tfi/>

- IPCC & WMO (Eds.). (1992). *Climate change: The 1990 and 1992 IPCC assessments, IPCC first assessment report overview and policymaker summaries and 1992 IPCC supplement*. IPCC.
- Keegan, B., Gergle, D., & Contractor, N. (2012). Staying in the loop: Structure and dynamics of Wikipedia's breaking news collaborations. *Proceedings of the Eighth Annual International Symposium on Wikis and Open Collaboration*, 1–10. <https://doi.org/10.1145/2462932.2462934>
- Kilgarrieff, A., Baisa, V., Bušta, J., Jakubíček, M., Kovář, V., Michelfeit, J., Rychlý, P., & Suchomel, V. (2014). *The Sketch Engine: Ten years on*. 1(1). <https://www.muni.cz/en/research/publications/1193200>
- Kincaid, J. P., Fishburne Jr., R. P., Rogers, R. L., & Chissom, B. S. (1975). *Derivation of New Readability Formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel* (Research Branch No. ADA006655; p. 51). NAVAL TECHNICAL TRAINING COMMAND MILLINGTON TN RESEARCH BRANCH. <https://apps.dtic.mil/sti/citations/ADA006655>
- Kobaliani, L. (2023). *Stanley Kubrick: The life and work of the great filmmaker* [Online magazine]. Artdevivre.Com. <https://artdevivre.com/articles/stanley-kubrick-the-life-and-work-of-the-great-filmmaker/>
- Korte, J. W., Bartsch, S., Beckmann, R., El Baff, R., Hamm, A., & Hecking, T. (2023). From causes to consequences, from chat to crisis. The different climate changes of science and Wikipedia. *Environmental Science & Policy*, 148, 103553. <https://doi.org/10.1016/j.envsci.2023.103553>
- Le Treut, H., & Somerville, R. (2007). *Box 1.1 Treatment of Uncertainties in the Working Group I Assessments—AR4 WGI Chapter 1*. Archive.Ipcc.Ch. https://archive.ipcc.ch/publications_and_data/ar4/wg1/en/ch1s1-6.html
- Lebart, L., & Salem, A. (1994). *Statistique textuelle*.
- Lissón, P., & Ballier, N. (2018). Investigating Lexical Progression through Lexical Diversity Metrics in a Corpus of French L3. *Discours. Revue de Linguistique, Psycholinguistique et Informatique. A Journal of Linguistics, Psycholinguistics and Computational Linguistics*, 23, Article 23. <https://doi.org/10.4000/discours.9950>
- Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Springer International Publishing. <https://doi.org/10.1007/978-3-031-02145-9>
- Loria, S. (2024). *Sloria/TextBlob* [Python]. <https://github.com/slوريا/TextBlob> (Original work published 2013)
- Masson-Delmotte, V., Zhai, P., Pirani, A., Connors, S. L., Péan, C., Berger, S., Caud, N., Chen, Y., Goldfarb, L., Gomis, M. I., Huang, M., Leitzell, K., Lonnoy, E., Matthews, J. B. R., Maycock, T. K., Waterfield, T., Yelekçi, Ö., Yu, R., & Zhou, B. (Eds.). (2021). *Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the*

- Intergovernmental Panel on Climate Change*. Cambridge University Press.
<https://doi.org/10.1017/9781009157896>
- Merriam-Webster. (2024, November 6). *Definition of METADATA*.
<https://www.merriam-webster.com/dictionary/metadata>
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013, January 16). *Efficient Estimation of Word Representations in Vector Space*. International Conference on Learning Representations.
<https://www.semanticscholar.org/paper/Efficient-Estimation-of-Word-Representations-in-Mikolov-Chen/f6b51c8753a871dc94ff32152c00c01e94f90f09>
- Nagpal, A., & Gabrani, G. (2019). Python for Data Analytics, Scientific and Technical Applications. *2019 Amity International Conference on Artificial Intelligence (AICAI)*, 140–145. <https://doi.org/10.1109/AICAI.2019.8701341>
- Nandwani, P., & Verma, R. (2021). A review on sentiment analysis and emotion detection from text. *Social Network Analysis and Mining*, 11(1), 81.
<https://doi.org/10.1007/s13278-021-00776-6>
- Nugues, P. M. (2024). *Python for Natural Language Processing: Programming with NumPy, scikit-learn, Keras, and PyTorch*. Springer Nature Switzerland.
<https://doi.org/10.1007/978-3-031-57549-5>
- Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global Vectors for Word Representation. In A. Moschitti, B. Pang, & W. Daelemans (Eds.), *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1532–1543). Association for Computational Linguistics. <https://doi.org/10.3115/v1/D14-1162>
- Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep Contextualized Word Representations. In M. Walker, H. Ji, & A. Stent (Eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (pp. 2227–2237). Association for Computational Linguistics. <https://doi.org/10.18653/v1/N18-1202>
- Peters, T. (2024). *The Zen of Python*. Python Enhancement Proposals (PEPs).
<https://peps.python.org/pep-0020/>
- Pincemin, B. (2018). *Sept logiciels de textométrie*. halshs-01843695.
- Pincemin, B., & Heiden, S. (2008). *What is textometry? Introduction*. Textometry project website. <https://txm.gitpages.huma-num.fr/textometrie/en/Introduction/>
- Poortvliet, P. M., Niles, M. T., Veraart, J. A., Werners, S. E., Korpelaar, F. C., & Mulder, B. C. (2020). Communicating Climate Change Risk: A Content Analysis of IPCC's Summary for Policymakers. *Sustainability*, 12(12), Article 12. <https://doi.org/10.3390/su12124861>

- Python Software Foundation. (2024). *Programming FAQ*. Python Documentation. <https://docs.python.org/3/faq/programming.html>
- Řehůřek, R. (2024). *Gensim: Topic modelling for humans*. https://radimrehurek.com/gensim/auto_examples/index.html#documentation
- Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In K. Inui, J. Jiang, V. Ng, & X. Wan (Eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (pp. 3982–3992). Association for Computational Linguistics. <https://doi.org/10.18653/v1/D19-1410>
- Reinert, A. (1983). Une méthode de classification descendante hiérarchique: Application à l'analyse lexicale par contexte. *Cahiers de l'analyse des données*, 8(2), 187–198.
- Ribé, M. M., Kaltenbrunner, A., & Keefer, J. M. (2021). Bridging LGBT+ Content Gaps Across Wikipedia Language Editions. *The International Journal of Information, Diversity, & Inclusion*, 5(4), 90–131.
- Roeder, G. G. (2011). *Climate models in modal adverbials: Representational practice and deep uncertainty in the IPCC summary documents* [University of British Columbia]. <https://doi.org/10.14288/1.0072479>
- Romero, M. (2024, January 18). *Mrm8488/distilroberta-finetuned-financial-news-sentiment-analysis* · Hugging Face. <https://huggingface.co/mrm8488/distilroberta-finetuned-financial-news-sentiment-analysis>
- Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11), 613–620. <https://doi.org/10.1145/361219.361220>
- Shivam, B., & Chaitanya, A. (2014). *textstat: Calculate statistical features from text* (Version 0.7.5) [Python]. <https://github.com/textstat/textstat>
- Sidorov, G., Gelbukh, A., Gómez-Adorno, H., & Pinto, D. (2014). Soft Similarity and Soft Cosine Measure: Similarity of Features in Vector Space Model. *Computación y Sistemas*, 18(3). <https://doi.org/10.13053/cys-18-3-2043>
- Similarweb.com. (2025, May). *Top Websites Ranking—Most Visited Websites in May 2025*. Similarweb. <https://www.similarweb.com/top-websites/>
- Singh, A., D'Arcy, M., Cohan, A., Downey, D., & Feldman, S. (2022). *SciRepEval: A Multi-Format Benchmark for Scientific Document Representations*. <https://doi.org/10.48550/ARXIV.2211.13308>
- Spolsky, J. (2003, October 8). *The Absolute Minimum Every Software Developer Absolutely, Positively Must Know About Unicode and Character Sets (No Excuses!)*. Joel on Software. <https://www.joelonsoftware.com/2003/10/08/the->

absolute-minimum-every-software-developer-absolutely-positively-must-know-about-unicode-and-character-sets-no-excuses/

- Teh, Y. W., Jordan, M. I., Beal, M. J., & Blei, D. M. (2006). Hierarchical Dirichlet Processes. *Journal of the American Statistical Association*, 101(476), 1566–1581. <https://doi.org/10.1198/016214506000000302>
- Tokui, S. (2024). *CuPy*. <https://cupy.dev/>
- Tol, R. S. J. (2023). *The IPCC and the challenge of ex post policy evaluation* (No. arXiv:2207.14724). arXiv. <https://doi.org/10.48550/arXiv.2207.14724>
- Trope, Y., & Liberman, N. (2010). Construal-Level Theory of Psychological Distance. *Psychological Review*, 117(2), 440–463. <https://doi.org/10.1037/a0018963>
- Ure, J. (1971). Lexical density and register differentiation. In G. E. Perren & J. L. M. Trim (Eds.), *Applications of linguistics: Selected Papers of the Second International Congress of Applied Linguistics*. (pp. 443–452). Cambridge University Press.
- Vaghefi, S. A., Stambach, D., Muccione, V., Bingler, J., Ni, J., Kraus, M., Allen, S., Colesanti-Senni, C., Wekhof, T., Schimanski, T., Gostlow, G., Yu, T., Wang, Q., Webersinke, N., Huggel, C., & Leippold, M. (2023). ChatClimate: Grounding conversational AI in climate science. *Communications Earth & Environment*, 4(1), 1–13. <https://doi.org/10.1038/s43247-023-01084-x>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 6000–6010.
- Weischedel, Ralph, Palmer, Martha, Marcus, Mitchell, Hovy, Eduard, Pradhan, Sameer, Ramshaw, Lance, Xue, Nianwen, Taylor, Ann, Kaufman, Jeff, Franchini, Michelle, El-Bachouti, Mohammed, Belvin, Robert, & Houston, Ann. (2013). *OntoNotes Release 5.0* (p. 2806280 KB) [Dataset]. Linguistic Data Consortium. <https://doi.org/10.35111/XMHB-2B84>
- Wikimedia Foundation. (2025). *List of Wikipedias—Meta*. https://meta.wikimedia.org/wiki/List_of_Wikipedias
- Wikipedia. (2024a). *Wikipedia:Cleaning up vandalism*. https://en.wikipedia.org/w/index.php?title=Wikipedia:Cleaning_up_vandalism&oldid=1219987517
- Wikipedia. (2024b). *Wikipedia:Protection policy*. https://en.wikipedia.org/w/index.php?title=Wikipedia:Protection_policy&oldid=1255616523
- Wikipedia. (2024c). *Wikipedia:WikiProject Usability/Readability guidelines*. In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Wikipedia:WikiProject_Usability/Readability_guidelines&oldid=1227775453#Resources_on_readability

- Witte, K. (1992). Putting the fear back into fear appeals: The extended parallel process model. *Communication Monographs*, 59(4), 329–349.
<https://doi.org/10.1080/03637759209376276>
- Wöhner, T., & Peters, R. (2009). Assessing the quality of Wikipedia articles with lifecycle based metrics. *Proceedings of the 5th International Symposium on Wikis and Open Collaboration*, 1–10.
<https://doi.org/10.1145/1641309.1641333>
- Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, Ł., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., ... Dean, J. (2016). *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation* (No. arXiv:1609.08144). arXiv.
<https://doi.org/10.48550/arXiv.1609.08144>
- Yamada, I., Asai, A., Sakuma, J., Shindo, H., Takeda, H., Takefuji, Y., & Matsumoto, Y. (2020). *Wikipedia2Vec: An Efficient Toolkit for Learning and Visualizing the Embeddings of Words and Entities from Wikipedia* (No. arXiv:1812.06280). arXiv. <https://doi.org/10.48550/arXiv.1812.06280>
- Ziogas, A. N., Schneider, T., Nun, T. B., Calotoiu, A., Matteis, T. D., Licht, J. D. F., Lavarini, L., & Hoefler, T. (2021). Productivity, portability, performance: Data-centric python. *Proceedings of SC 2021: The International Conference for High Performance Computing, Networking, Storage and Analysis: Science and Beyond*. <https://doi.org/10.1145/3458817.3476176>

Appendices

Appendix A: Libraries and Software Versions.....	71
Appendix B: Documents, Links and GitHub.....	72
Appendix C: [Python code] Extracting Text from a PDF File Using pypdf.....	74
Appendix D: [Python code] Text Extraction from Wikipedia.....	75
Appendix E: [Python code] Preprocessing with spaCy.....	76
Appendix F: [Python code] Lexicometry, Stylistic and Readability Processing.....	77
Appendix G: [Table] Top 10 TF-IDF Scores per Document.....	80
Appendix H: [Python code] Expression of Modality Processing.....	81
Appendix I: [Python code] Semantic Similarity Processing.....	83
Appendix J: [Python code] Topic Modelling Processing.....	85
Appendix K: [Python code] Sentiment and Emotion Processing.....	87
Appendix L: [Python code] Named Entity Recognition Processing.....	90
Appendix M: [Table] Complete spaCy NER Results.....	93
Appendix N: [Table] Complete Climate Change NER Results.....	94
Appendix O: [Python code] Data Visualisation.....	95

Appendix A: Libraries and Software Versions

Softwares:

- Python version: 3.11.11
- TXM version 8.4

Python Libraries:

- BERTopic 0.17.0
- matplotlib 3.10.0
- mwparserfromhell 0.6.6
- NLTK 3.9.1
- pandas 2.2.2
- pypdf 5.4.0
- requests 2.32.3
- scikit-learn 1.6.1
- sentence-transformers 4.1.0
- spaCy 3.8.4
 - with “en_core_web_lg” 3.8
- textstat 0.7.5
- transformers 4.51.1
- PyTorch 2.6

Large Language Models:

- Gemini 2.5 Pro Preview 03-25 (For data cleaning, Section 3.2.1)
- Gemini 2.5 Pro Preview 05-06 (For semantic chunking, Section 3.3.5)

Appendix B: Documents, Links and GitHub

Online Repository:

- [GitHub Repository](#)

https://github.com/shael-nlp/cc_representation

Wikipedia Articles:

- [Climate Change Portal](#)

https://en.wikipedia.org/wiki/Portal:Climate_change

- [Climate Change Mitigation Article \(2005\)](#)

https://en.wikipedia.org/w/index.php?title=Climate_change_mitigation&oldid=17726917

- [Climate Change Mitigation Article \(2008\)](#)

https://en.wikipedia.org/w/index.php?title=Climate_change_mitigation&oldid=210773525

- [Climate Change Mitigation Article \(2014\)](#)

https://en.wikipedia.org/w/index.php?title=Climate_change_mitigation&oldid=623826179

- [Climate Change Mitigation Article \(2022\)](#)

https://en.wikipedia.org/w/index.php?title=Climate_change_mitigation&oldid=1092994316

- [Climate Change Mitigation Article \(up to date\)](#)

https://en.wikipedia.org/wiki/Climate_change_mitigation

IPCC Documents:

- [AR1 WG3 SPM](#)

https://www.ipcc.ch/site/assets/uploads/2018/03/ipcc_far_wg_III_spm.pdf

- [AR2 WG3 SPM](#)

<https://archive.ipcc.ch/pdf/climate-changes-1995/spm-economic-social-dimensions.pdf>

- [AR3 WG3 SPM](#)

<https://www.ipcc.ch/site/assets/uploads/2018/03/wg3spm.pdf>

- [AR4 WG3 SPM](#)

<https://www.ipcc.ch/site/assets/uploads/2018/03/ar4-wg3-spm.pdf>

- [AR5 WG3 SPM](#)

https://www.ipcc.ch/site/assets/uploads/2018/02/ipcc_wg3_ar5_summary-for-policymakers.pdf

- AR6 WG3 SPM

https://www.ipcc.ch/report/ar6/wg3/downloads/report/IPCC_AR6_WGIII_SummaryForPolicymakers.pdf

- Treatment of Uncertainties (Likelihood and Confidence Expressions)

https://archive.ipcc.ch/publications_and_data/ar4/wg1/en/ch1s1-6.html

HuggingFace Models:

- all-MiniLM-L6-v2

<https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2>

- distilroberta-finetuned-financial-news-sentiment-analysis

<https://huggingface.co/mrm8488/distilroberta-finetuned-financial-news-sentiment-analysis>

- emotion-english-distilroberta-base

<https://huggingface.co/j-hartmann/emotion-english-distilroberta-base>

- specter-climate-change-NER

<https://huggingface.co/nicolauduran45/specter-climate-change-NER>

- specter2_base

<https://huggingface.co/allenai/specter2>

Datasets:

- Climate-Change-NER

<https://huggingface.co/datasets/ibm-research/Climate-Change-NER>

Software:

- TXM 8.4

<https://txm.gitpages.huma-num.fr/textometrie/index.html>

- Gemini 2.5 (Google AI Studio)

<https://aistudio.google.com/>

Appendix C: [Python code] Extracting Text from a PDF File Using pypdf

```
from pypdf import PdfReader

reader = PdfReader("IPCC_AR6_WGIII_SummaryForPolicymakers.pdf")
number_of_pages = len(reader.pages)

all_text = ""
for page_num in range(number_of_pages):
    page = reader.pages[page_num]
    text = page.extract_text()
    if text:
        all_text += text + " "

with open("AR6_WG3_SPM.txt", "w", encoding="utf-8") as file:
    file.write(all_text)
```

Appendix D: [Python code] Text Extraction from Wikipedia

```
import requests
import mwparserfromhell

title = "Climate change mitigation"
endpoint = "https://en.wikipedia.org/w/api.php"
date_iso = "2014-09-02T07:44:59Z"

params = {
    "action": "query",
    "format": "json",
    "prop": "revisions",
    "titles": title,
    "rvlimit": 1,
    "rvprop": "ids|timestamp|content",
    "rvdir": "older",
    "rvstart": date_iso,
}

data = requests.get(endpoint, params=params).json()
page = next(iter(data["query"]["pages"].values()))

# Manages old and new Wikipedia revision format (pre or post 2020)
revision = page["revisions"][0]
wiki_markup = revision.get("*") or revision.get("slots", {}).get("main",
{}).get("*", "")

# Convert wiki markup to plain text
wikicode = mwparserfromhell.parse(wiki_markup)
plain_text = wikicode.strip_code()

with open("wikipedia_article_revision.txt", "w", encoding="utf-8") as f:
    f.write(plain_text)

print(f"Downloaded and cleaned revision from {revision['timestamp']}")
```

Appendix E: [Python code] Preprocessing with spaCy

```
import spacy
from spacy.tokens import DocBin
from pathlib import Path
import os

INPUT_DIR = Path("data")
OUTPUT_DIR = Path("processed_docs")
MODEL = "en_core_web_lg"

nlp = spacy.load(MODEL)

print(f"Processing .txt files from: {INPUT_DIR}")
text_files = list(INPUT_DIR.glob("*.txt"))

for text_file_path in sorted(text_files):
    print(f"Processing: {text_file_path.name}...")

    with open(text_file_path, "r", encoding="utf-8") as f:
        text_content = f.read()

    doc = nlp(text_content)
    doc_bin = DocBin(docs=[doc])
    output_file_path = OUTPUT_DIR / (text_file_path.stem + ".spacy")

    doc_bin.to_disk(output_file_path)
    print(f"Successfully saved: {output_file_path}")

print("Preprocessing complete.")
```

Appendix F: [Python code] Lexicometry, Stylistic and Readability Processing

```
import spacy
from spacy.tokens import DocBin
from sklearn.feature_extraction.text import TfidfVectorizer
import textstat
from collections import Counter
import math
import os
import pandas as pd

# Configuration
INPUT_DIR = "processed_docs/"
MODEL = "en_core_web_lg"
OUTPUT_CSV = "corpus_metrics.csv"

# Loading spaCy to use en_core_web_lg's stop words list
nlp = spacy.load(MODEL)
STOP_WORDS = nlp.Defaults.stop_words
print(f"Successfully loaded spaCy model and {len(STOP_WORDS)} stop words.")

all_doc_metrics_data = []
corpus_for_tfidf = []
doc_names = []

doc_files = [f for f in os.listdir(INPUT_DIR) if f.endswith(".spacy")]

for file_name in sorted(doc_files): # Sort for consistent order
    print(f"Now processing: {file_name} !")
    file_path = os.path.join(INPUT_DIR, file_name)

    doc_bin_loaded = DocBin().from_disk(file_path)
    loaded_docs_from_file = list(doc_bin_loaded.get_docs(nlp.vocab))

    # While we could have saved multiple documents per binary file
    # We decided to save only 1 document per file to keep document names
    doc = loaded_docs_from_file[0]

    doc_names.append(file_name)

    current_doc_metrics = {"document_name": file_name}

    raw_text = doc.text
    all_tokens = [token for token in doc if not token.is_space]
    words_no_punct = [token for token in all_tokens if not token.is_punct]
    words_no_punct_no_stop = [token for token in words_no_punct if
    token.text.lower() not in STOP_WORDS]
    lemmas_no_punct_no_stop = [token.lemma_.lower() for token in doc if not
    token.is_punct and not token.is_space and not token.is_stop]

    # Word Counts
    current_doc_metrics["total_tokens_incl_punct"] = len(all_tokens)
    current_doc_metrics["total_words_excl_punct"] = len(words_no_punct)
    current_doc_metrics["total_words_excl_punct_stop"] =
    len(words_no_punct_no_stop)

    # Lexical Diversity
    total_lemmas_ld = len(lemmas_no_punct_no_stop)
    unique_lemmas_ld = len(set(lemmas_no_punct_no_stop))
    current_doc_metrics["TTR"] = (unique_lemmas_ld / total_lemmas_ld)
    current_doc_metrics["herdan_c"] = (math.log(unique_lemmas_ld) /
    math.log(total_lemmas_ld))
```

```

    current_doc_metrics["guiraud_r"] = (unique_lemmas_ld /
math.sqrt(total_lemmas_ld))

    # Lexical Density
    content_pos = {'NOUN', 'VERB', 'ADJ', 'ADV', 'NUM', 'PROPN'}
    words_for_ld = words_no_punct
    content_word_tokens_for_ld = [token for token in words_for_ld if
token.pos_ in content_pos]
    current_doc_metrics["lexical_density"] =
(len(content_word_tokens_for_ld) / len(words_for_ld))

    # Readability
    current_doc_metrics["FRE"] = textstat.flesch_reading_ease(raw_text)
    current_doc_metrics["FKGL"] = textstat.flesch_kincaid_grade(raw_text)

    # Avg Sentence Length
    num_sentences = len(list(doc.sents))
    current_doc_metrics["num_sentences"] = num_sentences
    current_doc_metrics["avg_sentence_length"] =
(current_doc_metrics["total_words_excl_punct"] / num_sentences)

    # Avg Word Length
    total_chars_in_words = sum(len(token.text) for token in words_no_punct)
    current_doc_metrics["avg_word_length"] = (total_chars_in_words /
len(words_no_punct))

    # Relative Frequency of Function Words
    function_pos_categories =
['ADP', 'AUX', 'CCONJ', 'SCONJ', 'DET', 'PRON', 'PART', 'INTJ']
    function_word_count = sum(token.pos_ in function_pos_categories for
token in words_no_punct)

    current_doc_metrics["rel_freq_function_words"] = (function_word_count /
current_doc_metrics["total_words_excl_punct"]) * 100

    # POS Tags Distribution
    pos_tags_list = [token.pos_ for token in words_no_punct]
    pos_counts = Counter(pos_tags_list)
    total_words_for_pos = len(pos_tags_list)
    major_pos_cats = ['NOUN', 'VERB', 'AUX', 'ADJ', 'ADV', 'PRON', 'ADP',
'CCONJ', 'SCONJ']

    for pos_cat in major_pos_cats:
        current_doc_metrics[f"pos_{pos_cat}"] = (pos_counts.get(pos_cat, 0)
/ total_words_for_pos) * 100
        current_doc_metrics["pos_OTHER"] = sum(count for tag, count in
pos_counts.items() if tag not in major_pos_cats and tag != 'SPACE') /
total_words_for_pos * 100

    all_doc_metrics_data.append(current_doc_metrics)

    # Appending text for TF-IDF
    tfidf_text = " ".join([token.lemma_.lower() for token in doc if not
token.is_punct and not token.is_space and not token.is_stop])
    corpus_for_tfidf.append(tfidf_text)

    # Configuration TF-IDF
    vectorizer = TfidfVectorizer(max_features=2000)
    tfidf_matrix = vectorizer.fit_transform(corpus_for_tfidf)
    feature_names = vectorizer.get_feature_names_out()
    num_top_tfidf_terms = 10

```

```

for i, doc_name_from_order in enumerate(doc_names):
    metrics_dict_for_doc = all_doc_metrics_data[i]

    doc_tfidf_scores = tfidf_matrix[i].toarray().flatten()
    top_indices = doc_tfidf_scores.argsort() [-num_top_tfidf_terms:] [::-1]
    top_terms_scores = [(feature_names[j], doc_tfidf_scores[j]) for j in
top_indices if doc_tfidf_scores[j] > 0.0001]
    metrics_dict_for_doc[f"top_{num_top_tfidf_terms}_tfidf_terms"] = ";
".join([f"{term}:{score:.4f}" for term, score in top_terms_scores])

# Conversion to DF and saving as CSV for analysis
metrics_df = pd.DataFrame(all_doc_metrics_data)
metrics_df.to_csv(OUTPUT_CSV, index=False)
print(f"Metrics successfully saved to: {OUTPUT_CSV}")

```

Appendix G: [Table] Top 10 TF-IDF Scores per Document

IPCC SPMs	TF-IDF Terms	Wikipedia Articles	TF-IDF Terms
AR1_WG3_SPM	emission:0.3439 change:0.2305 gas:0.2026 energy:0.1934 climate:0.1934 country:0.1729 greenhouse:0.1655 use:0.1580 develop:0.1432 resource:0.1364	N/A	N/A
AR2_WG3_SPM	cost:0.3520 change:0.2900 climate:0.2848 country:0.2090 emission:0.2012 damage:0.1663 economic:0.1542 equity:0.1427 policy:0.1333 estimate:0.1259	N/A	N/A
AR3_WG3_SPM	cost:0.3231 emission:0.2692 mitigation:0.1977 gas:0.1862 change:0.1805 scenario:0.1569 reduction:0.1506 climate:0.1489 country:0.1461 carbon:0.1432	Wiki_CCM_2005-06-26	global:0.2379 2005:0.2301 warming:0.2259 energy:0.2221 change:0.2062 kyoto:0.1905 climate:0.1904 emission:0.1904 carbon:0.1745 protocol:0.1738
AR4_WG3_SPM	emission:0.3420 mitigation:0.2347 eq:0.2250 spm:0.1957 co2:0.1912 energy:0.1789 cost:0.1443 global:0.1356 ghg:0.1348 potential:0.1313	Wiki_CCM_2008-05-07	energy:0.3701 carbon:0.2628 emission:0.2295 climate:0.2073 global:0.2073 warming:0.1986 change:0.1851 gas:0.1591 reduce:0.1517 power:0.1517
AR5_WG3_SPM	emission:0.3030 mitigation:0.2447 scenario:0.2330 energy:0.1943 evidence:0.1911 high:0.1676 co2eq:0.1631 spm:0.1573 medium:0.1508 figure:0.1444	Wiki_CCM_2014-09-02	emission:0.3501 energy:0.3369 climate:0.2290 carbon:0.2092 change:0.1982 global:0.1850 gas:0.1652 power:0.1608 nuclear:0.1592 2011:0.1240
AR6_WG3_SPM	confidence:0.4142 emission:0.4038 high:0.2604 mitigation:0.2071 pathway:0.1906 ghg:0.1851 global:0.1670 spm:0.1445 warm:0.1135 2019:0.1118	Wiki_CCM_2022-06-13	emission:0.3276 climate:0.3189 energy:0.2506 change:0.2488 carbon:0.2488 global:0.1629 reduce:0.1489 gas:0.1489 mitigation:0.1324 2021:0.1090

Appendix H: [Python code] Expression of Modality Processing

```
import spacy
from spacy.tokens import DocBin
from collections import Counter
import os
import re
import pandas as pd

# Configuration
INPUT_DIR = "processed_docs/"
OUTPUT_CSV = "modality_metrics.csv"
NORMALIZATION_FACTOR = 1000

MODAL_VERBS = [
    "can", "could", "may", "might", "must", "shall", "should", "will",
    "would"
]
MODAL_ADVERBS = ["apparently", "arguably", "assuredly", "certainly",
    "clearly",
    "conceivably", "definitely", "doubtless", "evidently", "hopefully",
    "indubitably", "ineluctably", "inescapably", "incontestably", "likely",
    "manifestly", "maybe", "necessarily", "obviously", "patently",
    "perhaps",
    "plainly", "possibly", "presumably", "probably", "seemingly", "surely",
    "truly", "unarguably", "unavoidably", "undeniably", "undoubtedly",
    "unquestionably"
]
CONFIDENCE_DICT = {
    "very high confidence": "90-100%", "high confidence": "80%",
    "medium confidence": "50%", "low confidence": "20%",
    "very low confidence": "0-10%"
}
LIKELIHOOD_DICT = {
    "virtually certain": "99-100%", "extremely likely": "95-100%",
    "very likely": "90-100%", "likely": "66-100%",
    "more likely than not": "50-100%", "about as likely as not": "33-66%",
    "unlikely": "0-33%", "very unlikely": "0-10%",
    "extremely unlikely": "0-5%", "exceptionally unlikely": "0-1%"
}

nlp = spacy.load("en_core_web_lg")
all_results_data = []

doc_files = [f for f in os.listdir(INPUT_DIR) if f.endswith(".spacy")]

for filename in sorted(doc_files):
    file_path = os.path.join(INPUT_DIR, filename)
    doc_name = os.path.basename(filename)

    print(f"Currently processing: {doc_name} !")

    doc_bin_loaded = DocBin().from_disk(file_path)

    loaded_docs_from_file = list(doc_bin_loaded.get_docs(nlp.vocab))
    doc = loaded_docs_from_file[0]

    current_likelihood_counts = Counter()
    current_confidence_counts = Counter()
    current_modal_verb_counts = Counter()
    current_modal_adverb_counts = Counter()
    current_negation_modal_count = 0
```



```

text_content_from_doc = doc.text
lower_text_content = text_content_from_doc.lower()

for phrase in LIKELIHOOD_DICT.keys():
    pattern = r'\b' + re.escape(phrase.lower()) + r'\b'
    matches = re.findall(pattern, lower_text_content)
    if matches:
        current_likelihoood_counts[phrase] += len(matches)

for phrase in CONFIDENCE_DICT.keys():
    pattern = r'\b' + re.escape(phrase.lower()) + r'\b'
    matches = re.findall(pattern, lower_text_content)
    if matches:
        current_confidence_counts[phrase] += len(matches)

for token in doc:
    if not token.is_punct and not token.is_stop:
        current_total_words += 1

    if token.pos_ == "AUX" and token.lemma_ in MODAL_VERBS:
        current_modal_verb_counts[token.lemma_] += 1
        # Negation check
        if token.i + 1 < len(doc) and doc[token.i + 1].lemma_ == "not":
            current_negation_modal_count += 1
        elif token.head.lemma_ == "not" and token.head.i == token.i - 1
:
            current_negation_modal_count += 1

    if token.pos_ == "ADV" and token.lemma_ in MODAL_ADVERBS:
        current_modal_adverb_counts[token.lemma_] += 1

for term, count in current_likelihoood_counts.items():
    norm_freq = (count / current_total_words) * NORMALIZATION_FACTOR
    all_results_data.append([doc_name, "Likelihood", term, count,
norm_freq, current_total_words])
for term, count in current_confidence_counts.items():
    norm_freq = (count / current_total_words) * NORMALIZATION_FACTOR
    all_results_data.append([doc_name, "Confidence", term, count,
norm_freq, current_total_words])
for term, count in current_modal_verb_counts.items():
    norm_freq = (count / current_total_words) * NORMALIZATION_FACTOR
    all_results_data.append([doc_name, "Modal Verbs", term, count,
norm_freq, current_total_words])
for term, count in current_modal_adverb_counts.items():
    norm_freq = (count / current_total_words) * NORMALIZATION_FACTOR
    all_results_data.append([doc_name, "Modal Adverbs", term, count,
norm_freq, current_total_words])
    norm_freq_neg = (current_negation_modal_count / current_total_words) *
NORMALIZATION_FACTOR
    all_results_data.append([doc_name, "Negation Near Modal", "count",
current_negation_modal_count, norm_freq_neg, current_total_words])

# --- Create Pandas DataFrame and Print/Save ---
metrics_df = pd.DataFrame(all_results_data, columns=[
    "Document", "Modal_Category", "Term", "Raw_Count",
    f"Normalized_Freq_{NORMALIZATION_FACTOR}", "Total_Words"
])

metrics_df.to_csv(OUTPUT_CSV, index=False)
print(f"Metrics successfully saved to: {OUTPUT_CSV}")

```

Appendix I: [Python code] Semantic Similarity Processing

```
import spacy
from spacy.tokens import DocBin
from sentence_transformers import SentenceTransformer, util
import os
import numpy as np
import csv
import torch

# Configuration
INPUT_DIR = "processed_docs/"
OUTPUT_CSV = "sentence_similarity_metrics.csv"

DOC_PAIRS = [
    (os.path.join(INPUT_DIR, "AR3_WG3_SPM.spacy"), os.path.join(INPUT_DIR,
"Wiki_CCM_2005-06-26.spacy")),
    (os.path.join(INPUT_DIR, "AR4_WG3_SPM.spacy"), os.path.join(INPUT_DIR,
"Wiki_CCM_2008-05-07.spacy")),
    (os.path.join(INPUT_DIR, "AR5_WG3_SPM.spacy"), os.path.join(INPUT_DIR,
"Wiki_CCM_2014-09-02.spacy")),
    (os.path.join(INPUT_DIR, "AR6_WG3_SPM.spacy"), os.path.join(INPUT_DIR,
"Wiki_CCM_2022-06-13.spacy")),
]

SBERT_MODEL_NAME = 'all-MiniLM-L6-v2'

nlp = spacy.load("en_core_web_lg")
sbert = SentenceTransformer(SBERT_MODEL_NAME)

sent_sim_results = []
header = [
    "IPCC_Document",
    "Wikipedia_Document",
    "IPCC_Sentences_Count",
    "Wikipedia_Sentences_Count",
    "Mean_Similarity",
    "Median_Similarity"
]
sent_sim_results.append(header)

for filepath_a, filepath_b in DOC_PAIRS:
    filename_a = os.path.basename(filepath_a)
    filename_b = os.path.basename(filepath_b)
    print(f"Comparing Sentences from: {filename_a} and {filename_b}")

    doc_bin_a = DocBin().from_disk(filepath_a)
    doc_a = list(doc_bin_a.get_docs(nlp.vocab))[0]

    doc_bin_b = DocBin().from_disk(filepath_b)
    doc_b = list(doc_bin_b.get_docs(nlp.vocab))[0]

    sentences_a = [sent.text.strip() for sent in doc_a.sents if
sent.text.strip()]
    sentences_b = [sent.text.strip() for sent in doc_b.sents if
sent.text.strip()]

    print(f"Found {len(sentences_a)} sentences in {filename_a},
{len(sentences_b)} in {filename_b}.\nStarting embeddings generation.")
```

```

embeddings_b = sbert.encode(sentences_b, convert_to_tensor=True,
show_progress_bar=False)

print("Embeddings done. Calculating cosine similarity...")
embeddings_b = embeddings_b.to(embeddings_a.device)
cosine_scores_ab = util.cos_sim(embeddings_a, embeddings_b)

scores = []
for i in range(len(sentences_a)):
    best_match = torch.max(cosine_scores_ab[i]).item()
    scores.append(best_match)

mean = f"{np.mean(scores):.3f}"
median = f"{np.median(scores):.3f}"
print(f"Mean: {mean}")
print(f"Median: {median}")

results_row = [
    filename_a,
    filename_b,
    len(sentences_a),
    len(sentences_b),
    mean,
    median
]
sent_sim_results.append(results_row)

with open(OUTPUT_CSV, mode='w', newline='', encoding='utf-8') as f:
    writer = csv.writer(f)
    writer.writerow(sent_sim_results)
print(f"Done. Results saved to {OUTPUT_CSV}")

```

Appendix J: [Python code] Topic Modelling Processing

```
import spacy
from spacy.tokens import DocBin
import glob
import os
from bertopic import BERTopic
import pandas as pd

# Configuration
INPUT_DIR = "processed_docs/"
OUTPUT_CSV = "topic_modeling_results.csv"

SPACY_MODEL = "en_core_web_lg"
EMBEDDING_MODEL = "all-MiniLM-L6-v2"

nlp = spacy.load(SPACY_MODEL)

corpus = []
doc_names = []

files = glob.glob(os.path.join(INPUT_DIR, "*.spacy"))

for filepath in files:
    doc_bin = DocBin().from_disk(filepath)
    docs = list(doc_bin.get_docs(nlp.vocab))

    doc = docs[0]

    processed_tokens = [
        token.lemma_.lower()
        for token in doc
        if not token.is_stop and not token.is_punct and not token.is_space
    ]

    corpus.append(" ".join(processed_tokens))

    doc_names.append(os.path.splitext(os.path.basename(filepath))[0])

# Topic modeling config
topic_model = BERTopic(
    embedding_model=EMBEDDING_MODEL,
    language="english",
    nr_topics="auto", # Topic number set to "auto"
    min_topic_size=2, # Keep to 2, higher values might affect results
    quality (due to corpus size)
    verbose=True
)

topics, probabilities = topic_model.fit_transform(corpus)

topic_info_df = topic_model.get_topic_info()

document_info_list = []
for i, doc_name in enumerate(doc_names):
    document_info_list.append({
        "Document_Name": doc_name,
        "Assigned_Topic_ID": topics[i],
    })

doc_topics = pd.DataFrame(document_info_list)
```

```
doc_topics,  
topic_info_df[['Topic', 'Name', 'Representation']],  
left_on='Assigned_Topic_ID',  
right_on='Topic',  
how='left'  
)  
  
results_df.to_csv(OUTPUT_CSV, index=False)  
  
print(f"Done. Results saved to: {OUTPUT_CSV}")
```

Appendix K: [Python code] Sentiment and Emotion Processing

```
import os
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from transformers import pipeline, AutoTokenizer,
AutoModelForSequenceClassification
import pandas as pd
import torch

# Configuration
INPUT_DIR = "split_docs/"
OUTPUT_CSV = "sentiment_emotion_metrics.csv"

analyzer = SentimentIntensityAnalyzer()

# Models
sentiment_tokenizer = AutoTokenizer.from_pretrained("mrm8488/distilroberta-
finetuned-financial-news-sentiment-analysis")
sentiment_model =
AutoModelForSequenceClassification.from_pretrained("mrm8488/distilroberta-
finetuned-financial-news-sentiment-analysis")
sentiment_pipeline = pipeline("sentiment-analysis", model=sentiment_model,
tokenizer=sentiment_tokenizer)

emotion_tokenizer = AutoTokenizer.from_pretrained("j-hartmann/emotion-
english-distilroberta-base")
emotion_model = AutoModelForSequenceClassification.from_pretrained("j-
hartmann/emotion-english-distilroberta-base")
emotion_pipeline = pipeline("text-classification", model=emotion_model,
tokenizer=emotion_tokenizer, return_all_scores=False)

# Initialization
all_document_metrics = []

all_filenames = [f for f in os.listdir(INPUT_DIR) if f.endswith(".txt")]

for filename in all_filenames:
    print(f"Processing {filename}...")
    filepath = os.path.join(INPUT_DIR, filename)

    with open(filepath, 'r', encoding='utf-8') as f:
        content = f.read()

    paragraphs = [p.strip() for p in content.split('\n\n') if p.strip()]
    num_paragraphs = len(paragraphs)

    document_data = {"document": filename, "total_paragraphs":
num_paragraphs}

# ----- Sentiment (VADER) -----
# -----

vader_compound = []
vader_pos = 0
vader_neg = 0
vader_neu = 0

for para in paragraphs:
    vs = analyzer.polarity_scores(para)
    vader_compound.append(vs['compound'])
    if vs['compound'] >= 0.05:
```

```

elif vs['compound'] <= -0.05:
    vader_neg += 1
else:
    vader_neu += 1

document_data["avg_VADER_compound"] = sum(vader_compound) /
num_paragraphs if num_paragraphs > 0 else 0
document_data["VADER_positive"] = (vader_pos / num_paragraphs) * 100 if
num_paragraphs > 0 else 0
document_data["VADER_negative"] = (vader_neg / num_paragraphs) * 100 if
num_paragraphs > 0 else 0
document_data["VADER_neutral"] = (vader_neu / num_paragraphs) * 100 if
num_paragraphs > 0 else 0

# ----- Sentiment (roBERTa) -----
# -----

roberta_sent_pos = 0
roberta_sent_neg = 0
roberta_sent_neu = 0

for para in paragraphs:
    result = sentiment_pipeline(para[:512])[0]
    label = result['label']
    if label == 'positive':
        roberta_sent_pos += 1
    elif label == 'negative':
        roberta_sent_neg += 1
    elif label == 'neutral':
        roberta_sent_neu += 1

document_data["roberta_positive"] = (roberta_sent_pos / num_paragraphs)
* 100 if num_paragraphs > 0 else 0
document_data["roberta_negative"] = (roberta_sent_neg / num_paragraphs)
* 100 if num_paragraphs > 0 else 0
document_data["roberta_neutral"] = (roberta_sent_neu / num_paragraphs)
* 100 if num_paragraphs > 0 else 0

# ----- Emotion (roBERTa) -----
# -----

emotion_labels = ["anger", "disgust", "fear", "joy", "neutral",
"sadness", "surprise"]
roberta_emotion_counts = {label: 0 for label in emotion_labels}

for para in paragraphs:
    result = emotion_pipeline(para[:512])[0] # Truncate
    label = result['label']
    if label in roberta_emotion_counts:
        roberta_emotion_counts[label] += 1

for label in emotion_labels:
    perc_emotion = (roberta_emotion_counts[label] / num_paragraphs) *
100 if num_paragraphs > 0 else 0
    document_data[f"emotion_{label}"] = perc_emotion

# DF conversion and saving as CSV

all_document_metrics.append(document_data)
print(f"Finished processing {filename}.")

```

```

df_combined = pd.DataFrame(all_document_metrics)

column_order = ["document", "total_paragraphs",
                "avg_VADER_compound", "VADER_positive", "VADER_negative",
                "VADER_neutral",
                "roberta_positive", "roberta_negative", "roberta_neutral"]
emotion_cols_ordered = [f"emotion_{label}" for label in ["anger",
                "disgust", "fear", "joy", "neutral", "sadness", "surprise"]]
column_order.extend(emotion_cols_ordered)

existing_columns_in_order = [col for col in column_order if col in
df_combined.columns]
df_combined = df_combined[existing_columns_in_order]

df_combined.to_csv(OUTPUT_CSV, index=False)
print(f"Done. Metrics saved to {OUTPUT_CSV}")

```


Appendix L: [Python code] Named Entity Recognition Processing

```
import os
import spacy
from spacy.tokens import DocBin
from transformers import AutoTokenizer, AutoModelForTokenClassification,
pipeline, logging as hf_logging
import pandas as pd
from collections import Counter
import torch

# Configuration
INPUT_DIR_SPACY = "processed_docs/"
SPLIT_TEXT_DIR = "split_texts/"
OUTPUT_CSV = "results_ner.csv"

MODEL_CC = "nicolauduran45/specter-climate-change-NER"

nlp = spacy.load("en_core_web_lg")

tokenizer_hf = AutoTokenizer.from_pretrained(MODEL_CC)
model_hf = AutoModelForTokenClassification.from_pretrained(MODEL_CC)

# Config for CC NER
cc_ner_pipeline = pipeline(
    "ner",
    model=model_hf,
    tokenizer=tokenizer_hf,
    aggregation_strategy="simple",
    device=0
)

all_results = []

doc_files = [f for f in os.listdir(INPUT_DIR_SPACY) if
f.endswith(".spacy")]

# Main loop
for filename in sorted(doc_files):
    filepath = os.path.join(INPUT_DIR_SPACY, filename)
    doc_name = os.path.basename(filename)

    # ----- Part 1 -----
    # -----

    print(f"Processing: {doc_name}")

    doc_bin = DocBin().from_disk(filepath)
    loaded_docs = list(doc_bin.get_docs(nlp.vocab))

    doc_obj = loaded_docs[0]

    # Total words from spaCy will be used for split_texts paragraphs as
    well
    # No need to recount words as texts are the same
    total_words = 0
    for token in doc_obj:
        if not token.is_punct and not token.is_stop:
            total_words += 1

    result_row = {'Document': doc_name, 'Total Words': total_words}
```

```

spacy_ents = [ent.label_ for ent in doc_obj.ents]
raw_ent_counts_spacy = Counter(spacy_ents)
normalized_ent_counts_spacy = Counter()

if total_words > 0:
    for ent_type, count in raw_ent_counts_spacy.items():
        normalized_ent_counts_spacy[ent_type] = (count / total_words) *
1000
else:
    for ent_type, count in raw_ent_counts_spacy.items():
        normalized_ent_counts_spacy[ent_type] = 0.0

for ent_type, count in raw_ent_counts_spacy.items():
    result_row[f'{ent_type}_spacy_raw'] = count
for ent_type, norm_count in normalized_ent_counts_spacy.items():
    result_row[f'{ent_type}_spacy_norm'] = norm_count

print(f"Found {len(doc_obj.ents)} entities with spaCy !")

# ----- Part 2 -----
# -----
split_text_filename = doc_name.replace(".spacy", ".txt")
split_text_path = os.path.join(SPLIT_TEXT_DIR, split_text_filename)

raw_ent_counts_cc = Counter()
normalized_ent_counts_cc = Counter()
total_cc_ent_doc = 0

if os.path.exists(split_text_path):
    with open(split_text_path, 'r', encoding='utf-8') as f:
        split_text = f.read()

    if split_text.strip():
        paragraphs = [p.strip() for p in split_text.split('\n\n') if
p.strip()]

        if not paragraphs:
            print(f"Error: No paragraphs found")
        else:
            all_ent_classes = []

            for i, paragraph_text in enumerate(paragraphs):

                # CC NER processing
                paragraph_results = cc_ner_pipeline(paragraph_text)

                for ent in paragraph_results:
                    all_ent_classes.append(ent['entity_group'])
                total_cc_ent_doc += len(paragraph_results)

            raw_ent_counts_cc = Counter(all_ent_classes)
            print(f"CC NER complete for {len(paragraphs)} paragraphs.
Found {total_cc_ent_doc} entities.")

            if total_words > 0:
                for ent_type, count in raw_ent_counts_cc.items():
                    normalized_ent_counts_cc[ent_type] = (count /
total_words) * 1000
            else:
                for ent_type, count in raw_ent_counts_cc.items():
                    normalized_ent_counts_cc[ent_type] = 0.0

```

```

        else:
            print(f"Cannot read: '{split_text_path}'. Check content or
encoding and try again")
        else:
            print(f"file not found: '{split_text_path}'. Check paths and try
again")

    for ent_type, count in raw_ent_counts_cc.items():
        result_row[f'{ent_type}_cc_raw'] = count
    for ent_type, norm_count in normalized_ent_counts_cc.items():
        result_row[f'{ent_type}_cc_norm'] = norm_count

    all_results.append(result_row)

# Convert to DF
results_df = pd.DataFrame(all_results)
results_df = results_df.fillna(0)

fixed_cols = ['Document', 'Total Words']

all_ent_prefixes = set()
for col_name in results_df.columns:
    if col_name not in fixed_cols:
        parts = col_name.split('_')
        if len(parts) > 1:
            prefix_candidate = "_".join(parts[:-1])
            all_ent_prefixes.add(prefix_candidate)

sorted_ent_prefixes = sorted(list(all_ent_prefixes))

ent_cols = []
for prefix in sorted_ent_prefixes:
    raw_col_name = f'{prefix}_raw'
    norm_col_name = f'{prefix}_norm'
    if raw_col_name in results_df.columns:
        ent_cols.append(raw_col_name)
    if norm_col_name in results_df.columns:
        ent_cols.append(norm_col_name)

final_columns = fixed_cols + ent_cols
results_df = results_df[[col for col in final_columns if col in
results_df.columns]]

results_df.to_csv(OUTPUT_CSV, index=False, encoding='utf-8')
print(f"Done. All metrics saved to: '{OUTPUT_CSV}'")

```

Appendix M: [Table] Complete spaCy NER Results

Frequencies normalized per 1000 words (punct. and stop-words excluded), rounded to 3 d.p.

Document	CARDINAL	DATE	EVENT	FAC	GPE	LANGUAGE
AR1_WG3_SPM	25.323	7.409	0.442	0.111	1.106	0.000
AR2_WG3_SPM	10.211	4.500	0.173	0.000	0.173	0.000
AR3_WG3_SPM	30.983	11.785	0.190	0.570	1.901	0.000
AR4_WG3_SPM	52.065	14.222	0.247	0.124	1.113	0.000
AR5_WG3_SPM	59.650	24.669	0.404	0.202	1.921	0.000
AR6_WG3_SPM	24.637	17.797	0.140	0.349	0.628	0.000
Wiki_CCM_2005-06-26	13.004	33.810	2.601	0.000	42.913	0.000
Wiki_CCM_2008-05-07	12.539	20.376	1.120	0.224	16.346	0.000
Wiki_CCM_2014-09-02	19.389	30.698	0.746	0.249	13.547	0.497
Wiki_CCM_2022-06-13	17.367	20.583	0.643	0.184	11.578	0.000
Document	LAW	LOC	MONEY	NORP	ORDINAL	ORG
AR1_WG3_SPM	1.769	1.659	0.885	0.221	1.216	15.371
AR2_WG3_SPM	1.558	0.173	1.211	0.346	1.211	5.365
AR3_WG3_SPM	3.802	0.570	3.421	0.950	0.190	17.107
AR4_WG3_SPM	1.731	1.360	2.350	0.495	2.350	22.755
AR5_WG3_SPM	1.618	0.101	1.921	0.607	0.708	19.412
AR6_WG3_SPM	1.256	0.349	0.209	1.047	0.698	22.334
Wiki_CCM_2005-06-26	7.802	3.901	1.300	2.601	1.300	45.514
Wiki_CCM_2008-05-07	2.463	2.239	2.687	1.567	2.463	31.572
Wiki_CCM_2014-09-02	3.107	1.119	2.610	2.237	1.740	41.511
Wiki_CCM_2022-06-13	1.930	2.205	2.113	3.124	0.368	22.972
Document	PERCENT	PERSON	PRODUCT	QUANTITY	TIME	WORK_OF_ART
AR1_WG3_SPM	5.529	0.663	0.995	2.543	0.111	0.442
AR2_WG3_SPM	2.942	0.000	0.000	0.000	0.000	0.173
AR3_WG3_SPM	2.661	1.711	5.132	1.140	0.190	0.190
AR4_WG3_SPM	7.915	2.473	5.813	2.226	1.360	0.371
AR5_WG3_SPM	10.312	2.831	2.730	2.629	0.101	0.000
AR6_WG3_SPM	16.052	0.768	5.793	0.489	0.070	0.209
Wiki_CCM_2005-06-26	6.502	5.202	0.000	0.000	0.000	0.000
Wiki_CCM_2008-05-07	6.941	11.196	3.135	4.254	0.896	1.120
Wiki_CCM_2014-09-02	5.841	15.287	2.859	2.610	0.249	0.621
Wiki_CCM_2022-06-13	12.037	5.881	1.562	3.032	0.368	0.459

Appendix N: [Table] Complete Climate Change NER Results

Frequencies normalized per 1000 words (punct. and stop-words excluded), rounded to 3 d.p.

Document	climate-assets	climate-datasets	climate-greenhouse-gases
AR1_WG3_SPM	20.126	6.303	12.275
AR2_WG3_SPM	12.288	1.385	3.461
AR3_WG3_SPM	7.603	25.470	13.496
AR4_WG3_SPM	14.222	19.664	22.384
AR5_WG3_SPM	20.119	31.847	20.928
AR6_WG3_SPM	20.310	21.078	23.032
Wiki_CCM_2005-06-26	18.205	2.601	6.502
Wiki_CCM_2008-05-07	17.689	0.224	20.376
Wiki_CCM_2014-09-02	12.304	2.983	21.999
Wiki_CCM_2022-06-13	22.604	1.930	7.535
Document	climate-impacts	climate-mitigations	climate-models
AR1_WG3_SPM	1.106	40.142	1.548
AR2_WG3_SPM	10.211	31.499	0.173
AR3_WG3_SPM	1.331	50.561	6.653
AR4_WG3_SPM	0.989	57.878	13.480
AR5_WG3_SPM	0.809	46.608	11.425
AR6_WG3_SPM	0.907	57.300	10.050
Wiki_CCM_2005-06-26	2.601	71.521	1.300
Wiki_CCM_2008-05-07	2.463	74.116	6.270
Wiki_CCM_2014-09-02	1.740	71.837	10.689
Wiki_CCM_2022-06-13	1.838	77.001	5.329
Document	climate-nature	climate-observations	climate-organisms
AR1_WG3_SPM	13.712	0.553	1.106
AR2_WG3_SPM	2.596	0.000	0.346
AR3_WG3_SPM	6.273	0.000	0.380
AR4_WG3_SPM	8.410	0.000	0.618
AR5_WG3_SPM	3.134	0.101	0.708
AR6_WG3_SPM	5.095	0.349	1.396
Wiki_CCM_2005-06-26	6.502	1.300	0.000
Wiki_CCM_2008-05-07	12.987	0.224	2.015
Wiki_CCM_2014-09-02	9.197	0.746	1.989
Wiki_CCM_2022-06-13	17.183	0.643	4.962
Document	climate-organizations	climate-problem-origins	climate-properties
AR1_WG3_SPM	19.020	42.243	7.851
AR2_WG3_SPM	7.442	16.961	5.365
AR3_WG3_SPM	11.595	37.635	7.983
AR4_WG3_SPM	10.017	49.963	23.621
AR5_WG3_SPM	9.099	33.970	22.040
AR6_WG3_SPM	6.281	38.945	13.819
Wiki_CCM_2005-06-26	55.917	9.103	3.901
Wiki_CCM_2008-05-07	55.083	38.065	8.285
Wiki_CCM_2014-09-02	47.974	39.150	14.666
Wiki_CCM_2022-06-13	20.674	43.646	10.567
Document	climate-hazards		
AR1_WG3_SPM	10.727		
AR2_WG3_SPM	8.827		
AR3_WG3_SPM	4.372		
AR4_WG3_SPM	4.081		
AR5_WG3_SPM	3.235		
AR6_WG3_SPM	2.931		
Wiki_CCM_2005-06-26	29.909		
Wiki_CCM_2008-05-07	11.420		
Wiki_CCM_2014-09-02	6.587		
Wiki_CCM_2022-06-13	11.853		

Appendix O: [Python code] Data Visualisation

Instead of using matplotlib in each script, we used a single script in which we manually changed the values for each figure. This appendix contains the original function, with generic values and all parameters set to default.

```
import matplotlib.pyplot as plt

# Document dates
years_ipcc = [1990, 1995, 2001, 2007, 2014, 2022]
years_ccm = [2005, 2008, 2014, 2022]

# Document names
ipcc_labels = ['AR1', 'AR2', 'AR3', 'AR4', 'AR5', 'AR6',]
ccm_labels = ['CCM 2005', 'CCM 2008', 'CCM 2014', 'CCM 2022']

# Values
ipcc_val = [1, 2, 3, 4, 5, 6]
ccm_val = [1, 2, 3, 4]

# Plot size
plt.figure(figsize=(8, 5))

# Style
plt.plot(years_ipcc, ipcc_val, marker='o', color='blue', label='IPCC WG3 SPM')
plt.plot(years_ccm, ccm_val, marker='o', color='purple', label='Wikipedia CCM')

# Text label positions are set individually by hand otherwise it overlaps
# IPCC
plt.annotate(ipcc_labels[0], (years_ipcc[0], ipcc_val[0]), xytext=(0, 0),
textcoords='offset points', fontsize=9, color='blue')
plt.annotate(ipcc_labels[1], (years_ipcc[1], ipcc_val[1]), xytext=(0, 0),
textcoords='offset points', fontsize=9, color='blue')
plt.annotate(ipcc_labels[2], (years_ipcc[2], ipcc_val[2]), xytext=(0, 0),
textcoords='offset points', fontsize=9, color='blue')
plt.annotate(ipcc_labels[3], (years_ipcc[3], ipcc_val[3]), xytext=(0, 0),
textcoords='offset points', fontsize=9, color='blue')
plt.annotate(ipcc_labels[4], (years_ipcc[4], ipcc_val[4]), xytext=(0, 0),
textcoords='offset points', fontsize=9, color='blue')
plt.annotate(ipcc_labels[5], (years_ipcc[5], ipcc_val[5]), xytext=(0, 0),
textcoords='offset points', fontsize=9, color='blue')
# Wiki
plt.annotate(ccm_labels[0], (years_ccm[0], ccm_val[0]), xytext=(0, 0),
textcoords='offset points', fontsize=9, color='purple')
plt.annotate(ccm_labels[1], (years_ccm[1], ccm_val[1]), xytext=(0, 0),
textcoords='offset points', fontsize=9, color='purple')
plt.annotate(ccm_labels[2], (years_ccm[2], ccm_val[2]), xytext=(0, 0),
textcoords='offset points', fontsize=9, color='purple')
plt.annotate(ccm_labels[3], (years_ccm[3], ccm_val[3]), xytext=(0, 0),
textcoords='offset points', fontsize=9, color='purple')

# Title
plt.xlabel('Year')
plt.ylabel('Current Metric')
plt.title('Title of Metric + Over Time')
plt.legend()
plt.grid(True, linestyle='--', linewidth=0.5, alpha=0.3)
plt.tight_layout()
plt.show()
```