Claim verification using tabular data

Cloze-style statements with structured statistical data

Savvas Pafitis[1]

Masters in Computational Statistics and Machine Learning 2020-2021

Sebastian Riedel, Pasquale Minervini

September 2021

[1]**Disclaimer:** This report is submitted as part requirement for the MSc in Computational Statistics and Machine Learning at UCL. It is substantially the result of my own work except where explicitly indicated in the text. The report may be freely copied and distributed provided the source is explicitly acknowledged.

Abstract

In this work, we construct a novel dataset of Cloze-style statements that require reasoning over structured data to answer. Our contribution, uses published statistical reports and the paired spreadsheet data from the UK Office of National Statistics. We assess a selection of current state-of-the-art language models in a zero-shot mask modelling scenario. We propose this probe as a new benchmark in understanding the ability of language models to reason over tables, in an open-domain setting. Furthermore, this benchmark can be used to qualify retrieval approaches in determining relevant context within a table.

To my grandfather.

Contents

1 Introduction **5**

1.1 Motivation . . . 5
1.2 Aims and Research Questions . . . 6
1.3 Summary . . . 7
1.4 Outline . . . 7

2 Background and related work **8**

2.1 Office of National Statistics . . . 8
2.2 Fact Checking . . . 9
2.3 Natural Language Processing . . . 11
2.4 Cloze-style questions . . . 12
2.5 OpenQA and MRC . . . 13
2.6 Answer generation, named entities and natural questions . . . 13
2.7 Adding context to language modeling . . . 14
2.8 Structured data as context . . . 15
2.8.1 Context: . . . 16
2.8.2 Table Linearisation: . . . 16
2.8.3 Extended encodings for structured data: . . . 16
2.9 Related Datasets . . . 17
2.9.1 Fact-Checking Corpus . . . 17
2.9.2 SQuAD . . . 19
2.9.3 FEVER . . . 20
2.9.4 LAMA . . . 21
2.9.5 TabFact . . . 21
2.9.6 PolitiHop . . . 22

3 Cloze-style dataset from ONS **24**

3.1 Statistical Bulletins . . . 24
3.2 Bulletin Structure . . . 26
3.2.1 Main Points . . . 26
3.2.2 Data used in the bulletin . . . 27
3.2.3 Analysis Section . . . 27
3.2.4 Related Links, Additional Information and Contact Details . . . 30

3.3 Application Programming Interface (API) . . . 31
3.3.1 No access to bulletins . . . 31
3.3.2 Inconsistent naming of datasets . . . 31
3.3.3 No way to link bulletins to datasets . . . 31
3.3.4 Incomplete collection of data . . . 32

4 Processing, filtering and analysis of data 33
4.1 Bulletin and dataset collection . . . 33
4.1.1 Finding bulletins . . . 33
4.1.2 Processing and storing bulletin content . . . 34
4.2 Structured data processing . . . 35
4.2.1 Preliminary pre-processing . . . 37
4.2.2 Identifying where a table starts . . . 37
4.2.3 Managing column headers . . . 38
4.2.4 Managing sub-tables . . . 40
4.2.5 Ongoing issues . . . 41
4.3 Filtering for relevant content . . . 42
4.3.1 Identifying relevant columns . . . 42
4.3.2 Identifying relevant rows . . . 43
4.3.3 Lemmatisation . . . 43
4.3.4 Relevant worksheets within spreadsheet . . . 44
4.4 Exploratory analysis of the data . . . 44
4.4.1 Bulletins and datasets: . . . 46
4.4.2 Common answers and answer-types: . . . 46
4.4.3 Multi-token answers: . . . 47
4.4.4 Required operations: . . . 48
4.5 Evaluation . . . 49
4.5.1 NER performance . . . 49
4.5.2 Structured data processing hyper-parameters . . . 51

5 Experimental Results 53
5.1 Model implementation . . . 53
5.2 Multi-token mask modelling . . . 54
5.3 Baselines . . . 55
5.4 Investigating predictions . . . 56
5.5 Why context reduces performance . . . 56
5.5.1 How does poor context affect the predictions . . . 58

6 Conclusions and further work 60
6.1 Summary . . . 60
6.1.1 Further work . . . 60

A Appendix **66**
A.1 Generating counterfactual data for table entailment 66
A.2 Available datasets within the ONS API 66
A.3 Bulletin categories and subcategories 68
A.4 Sampled clozes and answers with varying length 68
A.5 Amphetamine spreadsheet example 71

Chapter 1

Introduction

1.1 Motivation

Factually incorrect statements can have a very negative impact on public discourse. Such malicious statements have affected presidential elections, public opinion and continue to manipulate society in uncontrollable ways. In turn, fact-checking bodies are trying to minimise the effect of such "fake-news" by verifying claims and prompting readers with their findings. Manually doing so however can be slow and expensive. With the widespread usage of the internet and social media platforms, people are producing and consuming an ever increasing amount of information on a day-to-day basis and as such, manual verification fails as a scalable solution against misinformation.

Vast collections of (semi-)structured data, found as HTML-based tables or lists in websites such as Wikipedia or as downloadable spreadsheets such as the ones provided by national and international statistics services are widely accessible and contain factual information that can be useful in determining the credibility of a claim. As a response, researchers are looking into methods of utilising structured data for the automatic verification of the veracity of a claim.

Due to the open-domain nature of the problem, the relevancy of the source of information needs to be investigated using the appropriate retrieval techniques. Accordingly, natural language models should be able to reason over these tables, in a multi-hop context as most of the claims require complicated reasoning that requires more than a single inferential step.

Currently, there are no probes available that are comprised of real-life sociopolitical statements that require structured statistical data to verify. Most work provides artificial statements, usually generated from Wikipedia articles. Additionally, the few probes that provide structured data for table reasoning do so in a very limited way. They consist of artificial claims and provide tables of small size and rigid structure. To our knowledge no probe combines statements made by pundits and the tabular data that can answer them.

With this in mind, we propose a unique probe that pairs cloze-style claims with the statistical data required to answer them. The contents of the probe originate from statements made in statistical reports by the Office of National Statistics (ONS) on various subject areas about the United Kingdom. The statements cover a plethora of subject areas such as economics, politics, employment and community. Each claim is paired with a small collection of (semi-)structured spreadsheets that contain data related to the claim. These tables although structured, vary sig-

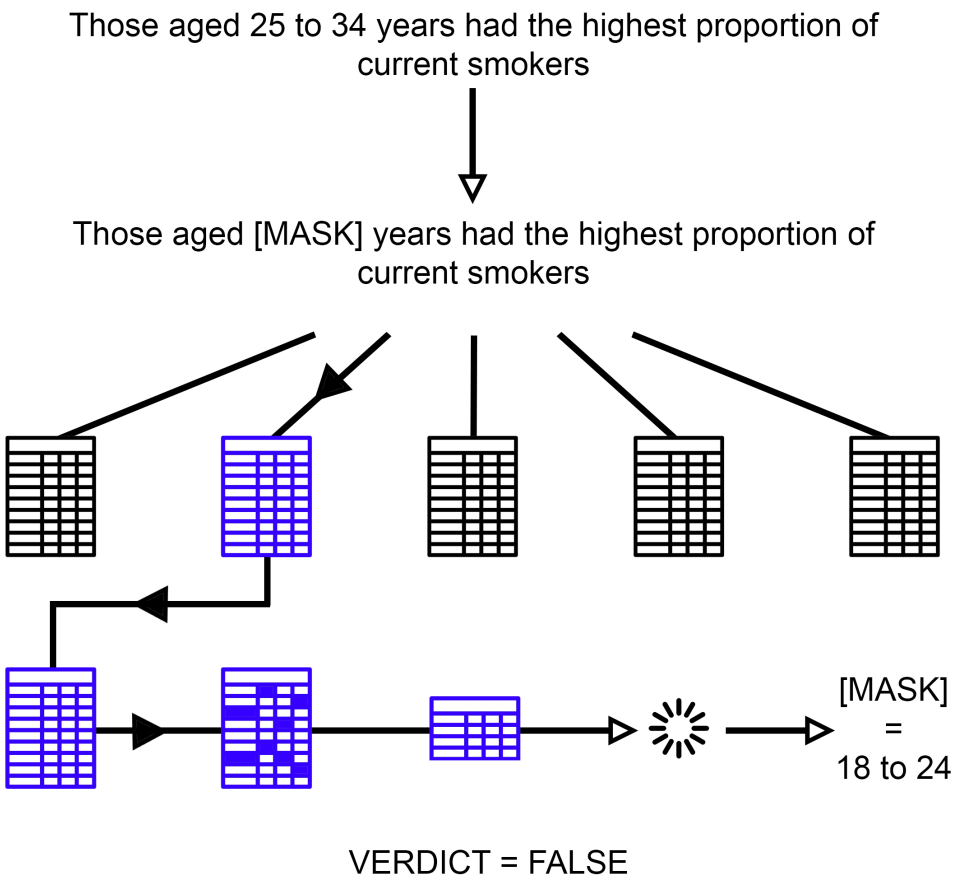


Figure 1.1: Motivation behind probe, what it aims to achieve an end-to-end showcase. 1) Receive a statement to be verified; 2) Convert statement into MLM task; 3) Find gold source from a collection of relevant datasets (open-domain); 4) Process data to find relevant contents to be used as context; 5) Reduce to small table for token capacity restrictions; 6) Reason (basic arithmetic, counting, comparing); 7) Generate prediction.

nificantly between them and require additional processing which is not required in other similar probes.

This novel probe is an new ambitious benchmark that requires the robust pre-processing of structured data, identification of relevant information within and the prediction/ generation of an answer using complex multi-hop reasoning.

Implementation details and code can be found here.[1]

1.2 Aims and Research Questions

- Collect published statistical reports and the structured data required,
- Generate Cloze-style statements from these reports,
- Build a processing pipeline that enables parsing of the statistical data,

[1] Anonymised repository: `https://anonymous.4open.science/r/thesis_pf996/`

- Assess and benchmark the current performance of state-of-the-art language models

Our contribution aims to provide the necessary statements and tabular data required to benchmark the predictive ability of language models in the claim verification scenario. Figure 1.1 showcases an end-to-end example. We receive a claim, in this case relating to the smoking habits of the United Kingdom. We convert the statement into a cloze-style statement and look for the gold source of information from an open-domain collection of spreadsheets. We then process this source, identifying the relevant content. We reduce the context into smaller chunks that current state-of-the-art models can process and determine the validity of the statement by performing reasoning over these tables. The reasoning we require falls in categories such as basic arithmetic, where we add or subtract two (or more) cells; or comparisons between cell values and similar.

1.3 Summary

Consider the prompts:

— In the UK, 15.9% of men smoked compared with 12.5% of women.

— Those aged 25 to 34 years had the highest proportion of current smokers (19.0%).

The motivation behind our dissertation is to automatically validate such a sentence against statistical data. Generally, this problem can be framed as a supervised machine learning problem, where the input claim is mapped to a true or false value. However, we generally do not have sufficient training data for this task. We aim to create a new probe that will allow language models to be used to answer prompts such as the above.

1.4 Outline

The dissertation is structured in the following way: we first describe the motivations and general setting in Chapter 1, the literature and related work in Chapter 2, a thorough discussion on data collection and pre-processing in Chapter 3, the retrieval and filtering techniques in Chapter 4, the experimental results in Chapter 5 and final conclusions and further work in Chapter 6.

Chapter 2

Background and related work

In this chapter we provide information on our main data source, discuss the motivations and background methods around fact verification as well as the associated work in natural language modelling. Discussions on the Office of National Statistics is found in Section 2.1, fact checking in Section 2.2, natural language processing in Section 2.3, and background and related methods in Sections 2.4-2.9.

2.1 Office of National Statistics

> *The Office for National Statistics is the UK's largest independent producer of official statistics and the recognised national statistical institute of the UK. It is responsible for collecting and publishing statistics related to the economy, population and society at national, regional and local levels. It plays a leading role in national and international good practice in the production of official statistics.*[1]

The Office of National Statistics (ONS) is the governmental body responsible for the collection and publication of statistical data related to the economy, population and society of the United Kingdom. It functions as the primary source of information for the usage in social and economic policy-making as well as a central resource[2] in debates about the determination of priorities, allocation of resources and decision making on interest rates and borrowing.[3]

The ONS publishes new content in the form of statistical reports on a weekly basis. These reports which ONS refers to as "bulletins", provide a summarisation of the accompanying datasets, written by in-house statisticians. Their main function is to provide documentation for the public that describes and explains the data collection process as well as the key findings of a particular study.

The ONS provides bulletins and corresponding datasets that cover the following areas:

- Agriculture and Environment

[1] `https://www.gov.uk/government/organisations/office-for-national-statistics`

[2] Official Statistics, Office of National Statistics, `https://www.ons.gov.uk/aboutus/whatwedo/statistics/statisticsweproduce`

[3] Office of National Statistics, Wikipedia, `https://en.wikipedia.org/wiki/Office_for_National_Statistics`

- Business and Energy
- Children, Education and Skills
- Crime and Justice
- Economy (ESCoE)
- Government
- Health and Social Care
- Labour Market
- People and Places
- Population
- Travel and Transport

For a more detailed description of the types of statistical reports and datasets available through ONS, please refer to Chapter 3.

2.2 Fact Checking

Fact checking is the task of determining the veracity of claims made by public figures such as politicians, pundits, celebrities and others. Commonly, this task is performed by journalists employed by news organisations in the process of news article generation (Vlachos et al. 2014). Recently, initiatives such as Snopes [4], FactCheck.org [5], PolitiFact[6] and Full Fact[7] have been introduced, dedicated in reducing misinformation and promoting transparency within public discourse. Example 2.1 shows two claims and corresponding journalist-derived verdicts.

[4] https://www.snopes.com/fact-check/
[5] https://www.factcheck.org
[6] https://www.politifact.com
[7] https://fullfact.org

EXAMPLE 2.1[a]
TAKEN FROM (VLACHOS ET AL. 2014)

Claim: (by Minister Shailesh Vara)
"The average criminal bar barrister working fulltime is earning some £84,000."

Verdict:**FALSE** (by Channel 4 Fact Check)
The figures the Ministry of Justice have stressed this week seem decidedly dodgy. Even if you do want to use the figures, once you take away the many overheads self-employed advocates have to pay you are left with a middling sum of money.

Claim: (by U.S. Rep. Mike Rogers)
"Crimea was part of Russia until 1954, when it was given to the Soviet Republic of the Ukraine."

Verdict:**TRUE** (by PolitiFact)
Rogers said Crimea belonged to Russia until 1954, when Khrushchev gave the land to Ukraine, then a Soviet republic.

Claim: (by President Barack Obama)
"For the first time in over a decade, business leaders around the world have declared that China is no longer the world's No. 1 place to invest; America is."

Verdict:**MOSTLY TRUE** (by PolitiFact)
The president is accurate by citing one particular study, and that study did ask business leaders what they thought about investing in the United States. A broader look at other rankings doesn't make the United States seem like such a powerhouse, even if it does still best China in some lists.

Assessing the truthfulness of a claim usually requires external information on the subject matter. Collecting and processing such information can be expensive and time-consuming for most fact-checkers. As per the Example 2.1, to answer the claim on the "full-time earnings" of a UK criminal-law barrister, the fact-checker would be required to gather wage data, expense reports and other financial statements to deduce the appropriate average as stated in the claim.

Misinformation, although commonly found in textual claims, can take multiple formats. With the computing power of new personal computers and the commercialisation of machine learning frameworks, users now have the ability to alter text (fake-news), images and videos (deep-fakes) with great ease. Here, we focus our work on misinformation in text-based format, as it remains largely the focus of most professional fact-checkers.

Work in the field of natural language processing (NLP) (Vlachos et al. 2014; Schlichtkrull et al. 2020), information retrieval (IR) (Yoneda et al. 2018; Hanselowski et al. 2019; Malon 2019),

databases and their intersection with computational journalism (Cohen et al. 2011; Flew et al. 2012), have fueled the discussion surrounding automated systems as a fact-checking service (Li et al. 2015; Lazer et al. 2018).

2.3 Natural Language Processing

Natural Language Processing (NLP) is the area of computer science that focuses on teaching machines to process, comprehend and generate human language. With the insurgence of available text data, NLP models have advanced sufficiently to be integrated in our day to day life; with prominent examples being the virtual assistants on commercial smartphones and services that offer grammar correction and machine translation. Such services are valuable in demonstrating the ability of language models (LM) to understand and reason.

Recent state-of-the-art models are pre-trained on very large collections of unlabelled text corpora. These are web-crawled data from the open web, books, transcripts from political debate such as conversations from the European Parliament and other sources. The goal of this pre-training is for the model to establish a satisfactory representations of language.

Language models, uni- or bi-directional, utilise the conditional probability of a word given the rest within a sentence to determine the likelihood of observing it. Uni-directional models commonly use the previous words within a sentence to produce a probability distribution of the immediate word following. Bi-directional models operate differently, now treating each word as a function of all other words within the sentence. This change, allows them to form contextual[8] representations of words, that are different based on the nature of the sentence they are found in.

Now, work by (Herzig, Nowak, et al. 2020) has showed that structured data such as spreadsheets or other table-like data, can also have a form of contextual embeddings. BERT-like models can be extended to understand how each cell within a table interacts with their parent column or row and how each cell interacts with others by encoding positional information in these embeddings. These models then can not only understand how language is used within a context but have also shown that they can reason over tables. This is investigated further in Section 2.8.

Additionally, such models have architectures that were specifically designed to allow them to focus. They can (shift) focus to relevant parts of the corpus and generate more accurate/ representative predictions. Specifically, two factors have attributed to this success: 1) Attention mechanisms, (Bahdanau et al. 2016), which enable the LM models to focus on relevant and important sub-parts of the context/ corpus, aiding its understanding; 2) Multi-hop architectures, (Bauer et al. 2019; De Cao et al. 2019), that enable the reading of corpora in multiple passes.

Both of these properties enable the model to continuously focus between the query and context, similarly to how a human would read the question and data over and over again until deducing what the answer should be. These enable the models to maintain the contextual information necessary in answering successfully such queries.

Finally, as recently shown by (Petroni, Rocktäschel, et al. 2019), the billions of parameters usually found in these LMs capture implicit information that was available to them during training.

[8] A common example is how the usage of the word "running" differs when used in sentences like "They are running for president" and "They are running a marathon". Bi-directional models, such as BERT (Devlin et al. 2019) allow for the distinction of these two occurrences of the same word.

With purposeful querying a user can prompt these models to generate answers similarly to how a Knowledge Base (KB)[9] operates. In conclusion, NLP techniques paired with the relevant factual information are a natural solution to the problem of fact verification at scale.

2.4 Cloze-style questions

Defined as short prompts of text with a part masked out (Taylor 1953), cloze-style questions have been increasingly investigated by the NLP community. Although first developed in psychology as a mechanism to measure effectiveness of communication, where a person was to fill-in the blank by reading and understanding a relevant document – cloze-style questions are now being used to train and test language models when tasked with reading comprehension and question answering (QA). Cloze claims and QA sentences, although grammatically different, can be converted to one another (Petroni, Rocktäschel, et al. 2019). For this reason, the development and availability of large cloze-style datasets has enabled the training of deep neural networks, with considerable success in the mask language modelling (MLM) and QA task.

An example within the context of our work is presented in Example 2.2. We present two clozes from the same source[10] sentence as more than one cloze question can be generated by masking different entities each time.

Example 2.2: Sample cloze statements from statistical bulletin[a]
Adult smoking habits in the UK: 2019

Cloze: The proportion of current smokers in the UK has fallen significantly from 14.7% in 2018 to 14.1% in [MASK]
Answer: 2019

Cloze: The proportion of current smokers in the UK has fallen significantly from 14.7% in [MASK] to 14.1% in 2019
Answer: 2018

Cloze-style sentences are directly used during the pre-training of language models specifically in the masked language model (MLM) task. This is the task BERT optimises for and since then has been a popular choice. The MLMs are tasked with predicting the missing token(s) within an input sentence. For instance as per Example 2.2 we could provide the MLM with the appropriate bulletin, in this case *"Adult Smoking Habits in the UK: 2019"* and provide the cloze sentence *"The proportion of current smokers in the UK has [MASK] significantly from 14.7% in 2018 to 14.1% in [MASK]"*. The model in this case would have to encode syntactic and semantic information (e.g. to predict *"fallen"*) as well as domain-specific knowledge and reasoning over structured data (e.g. to predict *"2019"*).

[9] Section 2.9.4 explores this further.

[10] Example taken from bulletin: `https://www.ons.gov.uk/peoplepopulationandcommunity/healthandsocialcare/healthandlifeexpectancies/bulletins/adultsmokinghabitsingreatbritain/2019`; Different models expect a different mask token. BERT uses [MASK] whereas RoBERTa uses <mask>. Cloze questions do not change and this arbitrarily changes their formulation.

2.5 OpenQA and MRC

Measuring the ability of models to store and use domain expertise, requires a downstream task where this knowledge is of crucial importance. Our work lies between open-domain question answering (OpenQA) and machine-reading comprehension (MRC).

OpenQA is a common and perhaps one of the most knowledge-intensive tasks in NLP as discussed by (Sun et al. 2018; Guu et al. 2020). It requires answering a question prompt without a pre-specified context. In comparison, MRC works with a given context, allowing machines to read and comprehend it to answer the prompt. We operate somewhere in the middle. Given a prompt, we do not know exactly which dataset contains the necessary information nor where to look within it. Likewise, we do not operate in a completely open-domain setting as we do not blindly search through all datasets of ONS.

For example, a statement about the smoking habits of people of the UK, cannot be answered by data about the financial economy. That said, we do not know which dataset contains the necessary information. Would it be a health dataset that describes yearly deaths (and possibly those caused by smoking) or would it be a dataset that collects information on the types of cigarettes people smoke?

Naturally, OpenQA is a harder task compared to MRC as it requires a retrieval component that identifies the gold source through a large body of datasets. Our work similarly requires a retrieval component, but has to identify the gold source through a smaller collection but one that contains highly relevant documents. Given that this smaller collection now contains relevant documents, it can be difficult to determine exactly which source of information is responsible for answering the claim in question.

Recent work, such as the one in (Yang et al. 2019), pair BERT-based models with an open-source information retrieval toolkit. Their work shows how, instead of operating over a small amount of input text, their system integrates IR techniques and BERT-readers to identify answers from large corpora of Wikipedia articles.

Although similar to what we aim to achieve, we point out that these readers are tuned for paragraph-like text whereas our work utilises (semi-)structured data in spreadsheet-like formats.

2.6 Answer generation, named entities and natural questions

As we have discussed in the previous sections, training these language models requires cloze questions to accompany the MLM task and some form of natural questions as in Open-QA to assess the quality of these models during inference time. Generating this data in an automatic way requires smart processing of our text data. Say we collected a large amount of corpora from various ONS reports. How do we convert this text to cloze questions? How do we identify what is relevant to mask out and how do we determine if the answer is expected to be numeric, a person or a country? How do we then generate natural sounding questions from these masked cloze sentences?

Work by (Lewis et al. 2019) investigates and provides potential solutions to these questions. They operate in the unsupervised extractive question answering (EQA) setting and study how to generate synthetic training data automatically. EQA is the task of answering questions given

a context document under the assumption that answers are spans of tokens within the given document. The unsupervised counterpart is when no aligned questions, contexts or answer data is available.

They generate synthetic data by first sampling random context paragraphs from a large corpus of documents. Then, random noun-phrases or named-entity (NE) mentions are chosen as answers. Named entities are words/ tokens that correspond to real-world objects[11] – for example, a person, a country, a product, a number. These sentences are then converted to cloze-style "fill-in-the-blank" sentences with the answer being the noun-phrase or named-entity. Finally, the cloze sentences are converted into natural questions using a *seq2seq* model that translates between cloze and natural questions. A complete list of these NEs can be found in Table 4.4 as we use them for our work too.

2.7 Adding context to language modeling

Pretraining language models on large collections of data has shown that a surprising amount of factual world knowledge is captured. This is crucial for multiple NLP tasks such as QA and specifically fact verification.

However, recent work (Guu et al. 2020; Petroni, Lewis, et al. 2020) suggests that the finite number of parameters in language models is a limiting factor for language models when storing and retrieving the factual knowledge they were exposed to during training time. Naively tackling this requires designing ever-larger networks, as knowledge is implicitly stored in the model parameters. Not only this is a nonsolution, it gate-keeps NLP research to those with access to high-powered computing units as most independent practitioners would be unable to train models of said size.

Instead, language models have been increasingly leveraging contexts when making predictions to alleviate this issue. Conditioning on: surrounding words (Mikolov et al. 2013), on whole sentences (Kiros et al. 2015), paragraphs (Radford et al. 2018; Devlin et al. 2019) or more relevantly structured data (Eisenschlos et al. 2020; Herzig, Nowak, et al. 2020; H. Zhang et al. 2020; Herzig, Müller, et al. 2021), has shown to improve the quality of the answers generated by enforcing language models to produce a response that is relevant to the conditional in question, subject to the quality of said conditioning. This inclusion of context, forces the language model to generate a more focused prediction as it restricts the domain of relevant answers but also raises the question of how one finds a relevant sentence, paragraph or table to condition on (Schlichtkrull et al. 2020).

Work that utilised an information retrieval component and a machine reader as in DrQA (D. Chen et al. 2017) investigated the task of machine reading at scale, by using Wikipedia as the unique knowledge source for open-domain QA. Here, the retriever is fixed and only the reading component is trained. REALM (Guu et al. 2020) extends previous work to train such a knowledge retriever using the mask language modelling task as training signal, by backpropagating through a retrieval step that considers millions of documents. They showed how knowledge retrievers can be trained in an unsupervised manner. (Petroni, Lewis, et al. 2020) takes this further by removing any need of supervision by using pre-trained language models and off-the-shelf information retrieval systems. Their work demonstrates that such retrieved context drastically improves BERT and RoBERTa on the LAMA probe, demonstrating the unsupervised machine reading capabilities of

[11] `https://spacy.io/usage/spacy-101#annotations-ner`

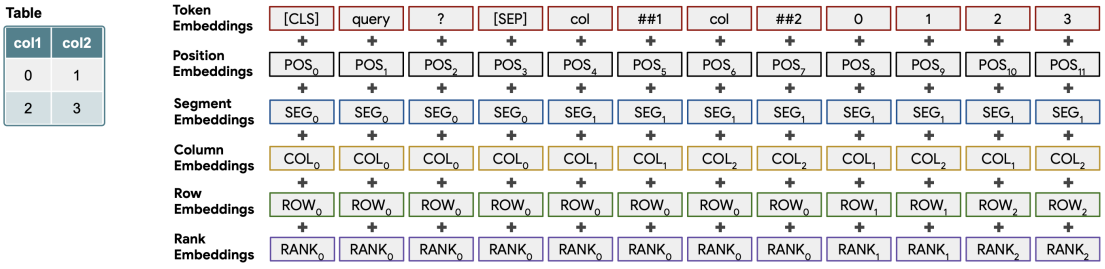


Figure 2.1: TaPas encodings. Taken directly from original paper (Herzig, Nowak, et al. 2020). Tokenization performed with the BERT-tokenizer. Showcases different encoding dimensions that are in charge of encoding table structure and cell interaction. Input follows standard practices of concatenating query with context separated by a [SEP] token.

LMs. Additionally, they found that off-the-shelf information retrieval systems are sufficient for BERT to match the performance of a supervised counterpart.

2.8 Structured data as context

Compared to fact verification over textual evidence, verification on (semi-)structured data requires additional extensions. We require our model to encode and understand structural information of said tables as well as be able to perform operations such as counting, comparing or basic arithmetic. Even with the largely dominating performance of pre-trained language models such as BERT, they cannot be directly used to encode structured data as they are pre-trained on unstructured natural language.

Traditionally, tasks that involved structured data were considered a semantic parsing problem. However the work in TaPas (Herzig, Nowak, et al. 2020) investigates how one can answer questions over tables without the generation of logical forms, instead utilising a weakly supervised approach. TaPas extends BERT's architecture to encode table inputs, train on text and tables and predict the required denotation by selecting appropriate table cells. TaPas also manages to handle more complicated denotations that require operations over multiple cells, such as Sum, Count, Average.

The way this is achieved is by modifying the architecture and in turn the encoding mechanism to construct embeddings that now are structure-aware. Instead of only storing information via token, sentence and transformer-positional embeddings, TaPas now introduces embeddings that capture and describe the structural form in tabular data. In the TaPas model implementation the embeddings now use token embeddings and 6 additional dimensions. However, TableBERT takes a different approach (W. Chen et al. 2020) where it instead seeks to generate a natural sounding sentence to describe the table.

To better understand how these embeddings work, we show how context is passed and how tables are linearised. We then describe the TaPas embeddings in detail.

2.8.1 Context:

Context is passed by appending it to the end of the input prompt, separated by a "[SEP]" token. This would look like:

[CLS] this is the input [SEP] this is the context

This concatenated sentence is treated as your input sentence. The embeddings are in charge of providing information on what is the question/ input prompt and whats the additional context provided. Pre-trained language models will encode the statements and linearised tables into continuous vectors for verification.

2.8.2 Table Linearisation:

Table linearisation is the process of taking structured data, usually in table format, and turning it into a long sequence. Usually, the standard BERT tokenizer is used to break the words in both statements and tables into subwords and join the two sequences with a [SEP] token in between. TaPas first explored this and does this by reading the table from left to right. Each cell value is tokenised and written into a sequence of tokens. A new row is indicated by the "[SEP]" token. TableBERT instead provides a more natural sounding approach, where the table is "vocalised", similarly to how one would read a table aloud. The difference here is that column names and cell values are paired together to create sentences like "Row 1: Column Name 1 is Cell Value 1; Column Name 2 is Cell Value 2; ...". See Table 2.1 for reference.

University	Name	Age
University College London	John	25
University College London	Jane	24

TaPas Linearisation: [SEP] University, Name, Age [SEP] University College London, John, 25 [SEP] University College London, Jane, 25

TableBERT Linearisation: [SEP] Row 1: University is University College London; Name is John; Age is 25 [SEP] Row 2: University is University College London; Name is Jane; Age is 24

Table 2.1: Table linearisation showcase

2.8.3 Extended encodings for structured data:

The 5 additional embedding dimensions, as in Figure 2.1, that TaPas uses are as follows:

- **Position ID** is the index of the token in the flattened sentence (identical to BERT)
- **Segment ID** is either 0 or 1, whether its the question or the context respectively
- **Column ID and Row ID** is the index of the column/ row that this token appears in. It takes value 0 if this is part of the question

- **Rank ID** if column values can be parsed as floats or dates, these are sorted accordingly and assign an embedding based on their numeric rank (0 for non-comparable, 1 for the smallest, $i + 1$ for those with rank i.
- **Previous Answer** is used if we are in a conversational setup where the current prompt might refer to a prompt we saw before. Special embedding is used to show if the token was part of the answer to the previous question (1 if was answer, 0 otherwise)

Work by (H. Zhang et al. 2020) demonstrate that providing positional information without pre-training is not sufficient for Transformers to encode tables. They have shown that, using positional encodings, linearising a table or even appending the appropriate column name as in TableBERT (discussed in 2.8.3) does not suffice. They instead propose Structure-Aware-Transformers. This extends the architecture by injecting structural information into the mask of self attention layers. This recovers alignment information by masking signals of unimportant cells during self-attention. In short, the lower layers ensure that: 1) cells from same row describe the same entry; 2) column name clarifies the attribute; 3) table title provides global background; 4) encode verification statement. The upper layers ensure satisfactory cross-row attention amonst cells of the same column. They further extend the linearisation to include an artificial SummaryRow, where they identify and concatenate key cells from multiple rows.

2.9 Related Datasets

Upon investigation of the related datasets we discuss below, it is evident that no probe requires the model to execute similar verification to what human annotators perform. These related probes instead provide a very clean, short and structured collection of tabular data.

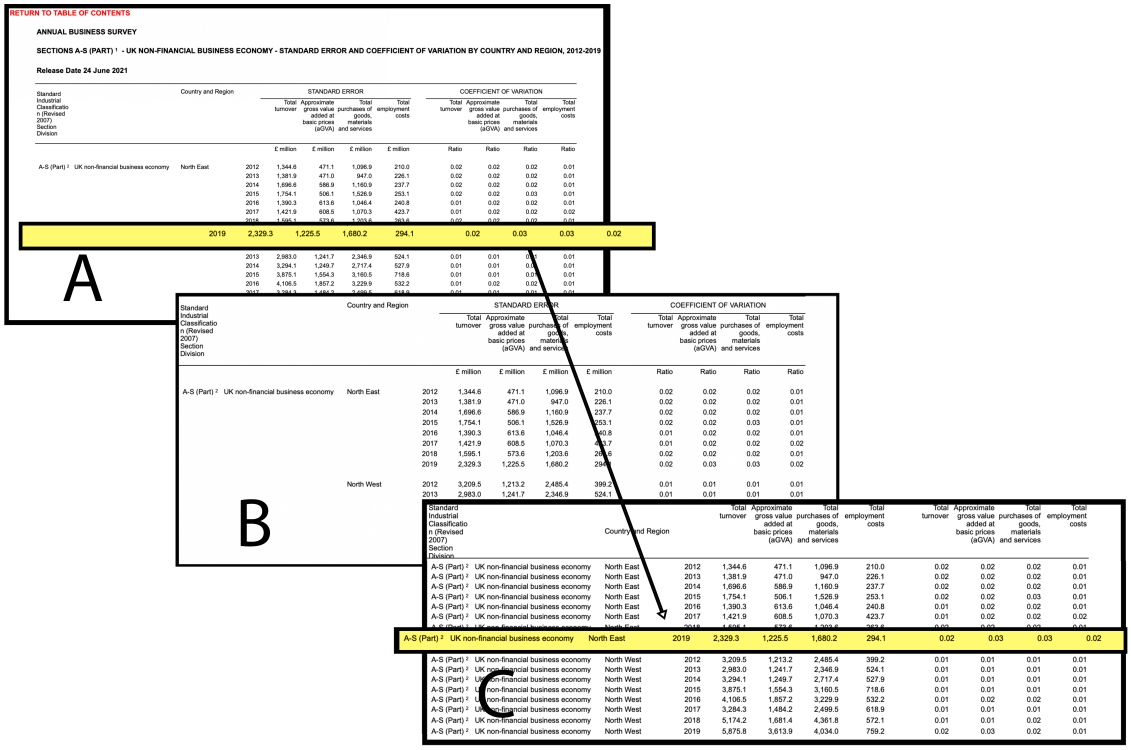
We instead pair our statements with the same tables that human annotators would have access to and require the same processing from our modelling. This entails identifying the relevant sub-context within which is often pre-labelled and given in other work. We do not do this as we operate in an unsupervised setting. We expect the solution to our probe to find the source of interest, identify relevant context and deduce/ reason the veracity of a claim.

To our knowledge, our work is the only one that provides an unlabelled collection of cloze-style statements originating from a single published source with paired structured/ tabular statistical data. These statements originate from publications that require statistical data for their verification and mimic the language and structure of what a reader would stumble upon. Below we discuss similar related datasets that were constructed around the task of automatic fact verification.

2.9.1 Fact-Checking Corpus

The work by Vlachos and Riedel in (Vlachos et al. 2014) yielded a dataset consisting of 221 fact-checked statements from PolitiFact[12] and Channel 4[13]. Included statements span a wide range of prevalent issues of U.K and U.S public life and are accompanied by detailed verdicts with granular labels such as True, MostlyTrue, HalfTrue and their negative counterparts.

[12] https://www.politifact.com/truth-o-meter/statements/
[13] https://blogs.channel4.com/factcheck/



Cloze:
In 2019, approximate gross value added at basic prices (aGVA) of the UK non-financial business economy was estimated to be [MASK] billion.

LINEARISED CONTEXT
Before processing:
Row 8 is: ...; - is - ; Country and Region is -; - is 2019; Total turnover is 2,329.3; Approximate gross value added at basic prices (aGVA) is 1,225.5; Total purchases...; ...
After processing:
Row 8 is: ...; - is UK-Non financial business economy; Country and Region is -; - is 2019; Total turnover is 2,329.3; Approximate gross value added at basic prices (aGVA) is 1,225.5; Total purchases...; ...

Figure 2.2: From raw data to linearised context. Figure presents how raw data from a bulletin is processed to achieve the required linearised context. Highlighted row is linearised to construct context that can answer target cloze. After processing, context includes necessary keywords (such as *"UK non-financial industry"*). (TABLE A) represents the data as-is from source. (TABLE B) represents table after unrelated rows/ columns have been omitted (these are usually meta-data, statistician's comments, links to other excel sheets or badly formatted spreadsheets). (TABLE C) represents the augmentation of cell values to complete missing entries originating from multi-level headers (*"North East"* and *"North West"* are forward-filled for the linearisation of context to work correctly).

Additional information such as speaker details, creation time, links to the complete analysis of verdicts, sources used and a suitability-comment on the quality of the statement for fact-checking is included for every entry in the dataset.[14] The authors note that it is common for sources to include statistical reports or tabular data however these are not always usable as the language used is indicative of the verdict.[15]

However, the justification for each verdict, including the sources used is not readily available in a machine-readable form. Although this information is usually available in analyses by journalists, this becomes a challenge when creating claim verification components. With this, the task defined by the dataset can be challenging for available NLP methods to solve(Thorne, Vlachos, Christodoulopoulos, et al. 2018).

Our work is different from the work described above in the following aspects: 1) we only use statements and data originating from a single source, ONS; 2) we pair each statement with the necessary statistical data for verification; 3) our datasets although structured require extensive and robust (pre-)processing on a case-by-case basis.

2.9.2 SQuAD

Stanford Question Answering Dataset (SQuAD), (Rajpurkar, J. Zhang, et al. 2016; Rajpurkar, Jia, et al. 2018), and more recently SQuAD2.0 is a reading comprehension probe, complete with more than 150000 questions posed by humans based off Wikipedia articles. The answer is a string or segment of text from the corresponding reading passage. The SQuAD probe is arguably one of the most important benchmarks within this section of NLP. In its inception, automatic verification of these questions was considerably hard (F1: 51.0% vs 86.8% for humans). Current state of the art solutions surpass human performance (F1: 93.2%).

SQuAD example 2.3: Sample article and statements

Article (snippet): In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravity. The main forms of pre- cipitation include drizzle, rain, sleet, snow, graupel and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals within a cloud. Short, in- tense periods of rain in scattered locations are called "showers".

1) What causes precipitation to fall? → *gravity*
2) What is another main form of precipitation besides drizzle, rain, snow, sleet and hail? → *graupel*
3) Where do water droplets collide with ice crystals to form precipitation? → *within a cloud*

SQuAD is considerably different from our work. Given that the questions are constructed directly from the article, the answer is expected to be an exact string or sequence of strings within.

[14] Statements that cannot be assessed objectively have been avoided. This includes: speculative statements, statements that include discussions of causal relations or are unrelated to factual information are deemed inappropriate.

[15] E.g. part of the reports corresponding to Claim 1 in Example 2.1 reads: "the full-time figure has the handy effect of stripping out the very lowest earners and bumping up the average". The answer can then be extracted trivially from this sentence.

In comparison, our work expects the solution to identify the source and reason over the required table to construct its prediction. Additionally, the nature of questions differs considerably as the publications we work on have a considerably narrower scope. Our work can mimic claims found in political and public debate whereas SQuAD focuses more on scientific or historical statements.

2.9.3 FEVER

The authors introduced FEVER: Fact Extraction and VERification shared task, (Thorne, Vlachos, Christodoulopoulos, et al. 2018; Thorne, Vlachos, Cocarascu, et al. 2019), in 2018 and subsequent extension in 2019. The goal of the task is to assess the validity of a claim based on a collection of supporting statements from Wikipedia. FEVER also inspired the creation of similar datasets in languages other than English such as (Khouja 2020).

The dataset consists of 185,445 claims manually verified against the introductory sections of Wikipedia entries and classified as SUPPORTED, REFUTED or NOTENOUGHINFO. Human annotators also mutated[16] (entity switching, negation of sentences, etc.) original claims to expand the collection of claims. Annotation pipelines were constructed to ensure consistency, returning a high inter-annotator agreement (Thorne, Vlachos, Christodoulopoulos, et al. 2018).

FEVER requires three sub-tasks: document/ information retrieval, evidence extraction and answer generation. Work such as (Yoneda et al. 2018) used a pipelined approach whereas (Nishida et al. 2019) treat evidence extraction and answer generation simultaneously by regarding the task an explainable multi-hop question answering (QA) task. Multi-hop reasoning is required for the task to be solved as the claims are often composed by multiple statements which at times can be contradictory (Hardalov et al. 2021). This negatively affects the interpretability as evidence used to reason is not necessarily located close to the answer, making it difficult for users to verify (Nishida et al. 2019).

In the following extension, FEVER 2 explores adversarial attacks on the original FEVER dataset. This included a "break-it" phase where a collection of adversarial instances were designed to induce classification errors in fact verification systems. These types of attacks highlight the limitations of systems when performing inductive reasoning and composition of knowledge (Thorne, Vlachos, Cocarascu, et al. 2019). This extension, as well as work by (Niewinski et al. 2019; Hidey et al. 2020), investigate classification errors from adversarial attacks and allow for higher interpretability and understanding of the results — a crucial step if widespread usage of such systems is expected (Atanasova et al. 2020; Ostrowski et al. 2021).

Our work is similar to FEVER in the sense that we require an information retrieval step, evidence extraction as well as an answer generation step. However, it differs from FEVER in the context that our claims require statistical data, presented in tabular format, to confirm their validity. FEVER only operates using context from Wikipedia articles. Additionally, FEVER sentences are considerably shorter, averaging at 8 words. This is considerably different from the rationales produced by ONS statements which require the synthesis of information. We also do not have annotated data[17] which makes this a much more difficult benchmark.

[16] Mutations and alterations of original claims sometimes change the meaning of the original claim but are still valid claims that can be answered through the same data.

[17] We understand how annotation is crucial but is something we cannot provide with our resources. However, the scale of our probe can be continuously expanded as ONS publications are updated (bi-)weekly which makes annotation very difficult.

2.9.4 LAMA

LAMA: LAnguage Model Analysis probe was introduced in (Petroni, Rocktäschel, et al. 2019) as a dataset that helps answer whether pretrained *off-the-shelf* language models (BERT, ELMo, etc.) can be used as knowledge bases. The probe is motivated by the fact that traditional knowledge bases require complex NLP pipelines, involving entity extraction, coreference resolution, entity linking and relation extraction (Surdeanu et al. 2014; Petroni, Rocktäschel, et al. 2019), whereas language models possess attractive properties such as the lack of need for any schema engineering, human annotation and the support of open set of queries.

Within this work, a LM is defined to *know* an answer to a fact (Subject, Relation, Object) such as (Dante, born-in, Florence) if it can successfully predict masked objects in cloze-style sentences as "Dante was born in _____". It was designed to understand the amount of relational knowledge stored, the observable differences between types of knowledge (facts about entities, common sense, general QA) and how symbolic knowledge bases compare. As an end-goal, the work explores whether LMs can be used to design better unsupervised knowledge representations that could transfer factual and commonsense knowledge reliably to downstream tasks.

The knowledge set was constructed from four sources: Google-RE[18], ConceptNet (Speer et al. 2018), T-REx (Elsahar et al. 2018) and SQuAD (Rajpurkar, J. Zhang, et al. 2016). Google-RE and SQuAD have statements manually aligned with some Wikipedia text by design. T-REx facts are automatically aligned to Wikipedia and this alignment can be noisy (although (Elsahar et al. 2018) showed that the alignment technique is highly accurate at 97.8% over a test set). ConceptNet on the other hand has no guarantee about explicit alignment with Wikipedia corpora. Exact details on the origins of each corpus, how the cloze-style questions are generated and the extend these fact align with text found in Wikipedia are discussed in the original paper.

Recent work by (Poerner et al. 2020), show that BERT-like models rely heavily on entity names to generate plausible answers. Qualitative inspections of answers given from LAMA prompts suggest that this is a recurrent effect. As an example, when asked what is the nationality of an Italian sounding name, BERT predicts Italy. Although this is frequently correct, it is an indicator that such language models lack memorisation of facts (Poerner et al. 2020; Anonymous 2021). With this in mind, a modification is introduced LAMA-UHN (UnHelpful Names), a subset of the original LAMA dataset where helpful names have been removed.

Our work is distinct from LAMA since: 1) we do not manually align claims to data; 2) our claims can only be answered using statistical data which are found in a structured format; 3) this data is not readily available and hence was unseen during training time unlike the text corpora from Wikipedia; 4) the extracted claims mimic the language, structure and content of claims readers would find and need to verify themselves.

2.9.5 TabFact

TabFact, (W. Chen et al. 2020), introduces a large scale dataset which is made up by 118K manually annotated statements paired with 16K Wikipedia tables. The statements within the dataset are human annotated into Entailed and Refuted categories. The TabFact dataset uses

[18] `https://code.google.com/archive/p/relation-extraction-corpus/`

tables as the premise or source of information to classify the statements into the two categories. The probe is challenging as it requires linguistic as well as symbolic reasoning.

(W. Chen et al. 2020) also introduces Table-BERT and Latent-Program-Algorithm to solve these reasoning challenges. The original paper obtains less accurate results on their probe when compared to the TaPas architecture. We expect this to be the case due to the lack of table-specific pre-training. Linguistic reasoning requires linguistic inference or common sense, unlike past work where linguistic reasoning is dominated by paraphrasing. Symbolic reasoning requires symbolic execution over the table structure such as conditioning on specific rows/ columns as well as the execution of arithmetic operations.

They contrast their work with question answering as statements within TabFact might contain compound facts that need to be verified to predict the verdict. Our work is similar in the sense that the statements included require linguistic and symbolic reasoning. In contrast though, our work is paired with ONS spreasheets which are not known a-priori to contain the ground truth and require an extra retrieval step as well as a relevancy evaluation step.

The structure of our datasets varies significantly between each single entry and require case-specific processing. A solution should be able to work for all of the entries and thus requires a very robust architecture. TabFact has very limited tables. They are considerably shorter as they originate from the HTML tables from Wikipedia. These are rarely larger than 5 rows whereas our work consists of tables with mean size of 352 rows and 25 columns. Additionally, they never have missing column headers or cell properties as Wiki-tables follow very rigid formatting. This is not the case for the ONS data.

2.9.6 PolitiHop

PolitiHop, (Ostrowski et al. 2021), is the first political fact checking dataset with annotated evidence sentences. It requires multi-hop reasoning encompassing evidence retrieval and claim veracity prediction. The probe consists of 500 real-world claims with manual annotations, True, HalfTrue, False, of collections of interlinked evidence chunks from PolitiFact articles. These are required to predict the claims' labels.

The work addresses how multi-hop architectures compare versus single inference ones. They investigate whether adversarial cases, such as where NEs appear in both evidence and non-evidence sentences. They also study whether pre-training on in-domain datasets improves model performance. They observed that multi-hop architectures show better performance to the single inference counterparts. This is especially true during the retrieval step, where BERT can be easily fooled by named entities overlapping, between claim and evidence.

This work is closely related to our probe given that the types of claims present in both are similar in structure and content. Most probes are designed around artificial claims usually from databases such as Wikipedia. However, both PolitiHop and our probe extract sentences from published sources and cover socioeconomic and political claims. Our probe differs in content as we cover a larger selection of subject areas such as leisure, tourism, household characteristics and others.

A key difference is that PolitiHop takes claims politicians make and compares them with the sentences in published bulletins. The comparison then is done between text. We on the other hand

take sentences from published bulletins and compare them with statistical data. The comparison and usage of bulletins then is different.

We also note how PolitiHop only expects text as context. These are pre-labelled strings and have no inherent structure. PolitiHop does not utilise any tables nor does it expect table-reasoning. We work in a different domain and expect our claims to be answerable by reasoning over data in a structured format. With these observations then, we consider our contribution distinct from PolitiHop.

Chapter 3

Cloze-style dataset from ONS

In this chapter, we showcase the types of information available in the ONS website. We explore the contents of the statistical bulletins in Section 3.1 and their structure in Section 3.2. Finally we explain the limitations of the ONS API in the context of our probe in Section 3.3.

3.1 Statistical Bulletins

The ONS couples dataset releases with a collection of statistical bulletins that describe the findings of the analyses as well as recent additions and changes made. These bulletins are short, usually page-long analyses, that summarise key findings and provide a commentary on the data. For the reasons we discuss in the following sections, we believe ONS bulletins to be a good proxy to statistical claims found in articles, political debates and public discussions. We construct our probe based on content found from these bulletins.

According to the ONS website, bulletins should:[1]

- open up information to inquiring citizens
- provide easy links to data
- include only the most important information
- link to more detailed content for those who need it

Research[2] on user types and user expectations determines the bulletin structure. In-depth analysis of commonly found user types and data usage is available here. In short, the bulletins are designed to enable users complete[3] three key tasks:

- read analysis of the latest data, (accessibility)
- find the latest data, (reproducibility)
- understand how the data was collected and the methodology of the analysis, (transparency)

[1] https://style.ons.gov.uk/statistical-bulletin/bulletin-sections/
[2] https://style.ons.gov.uk/category/writing-for-the-web/personas/
[3] https://style.ons.gov.uk/category/statistical-bulletin/what-is-a-bulletin/

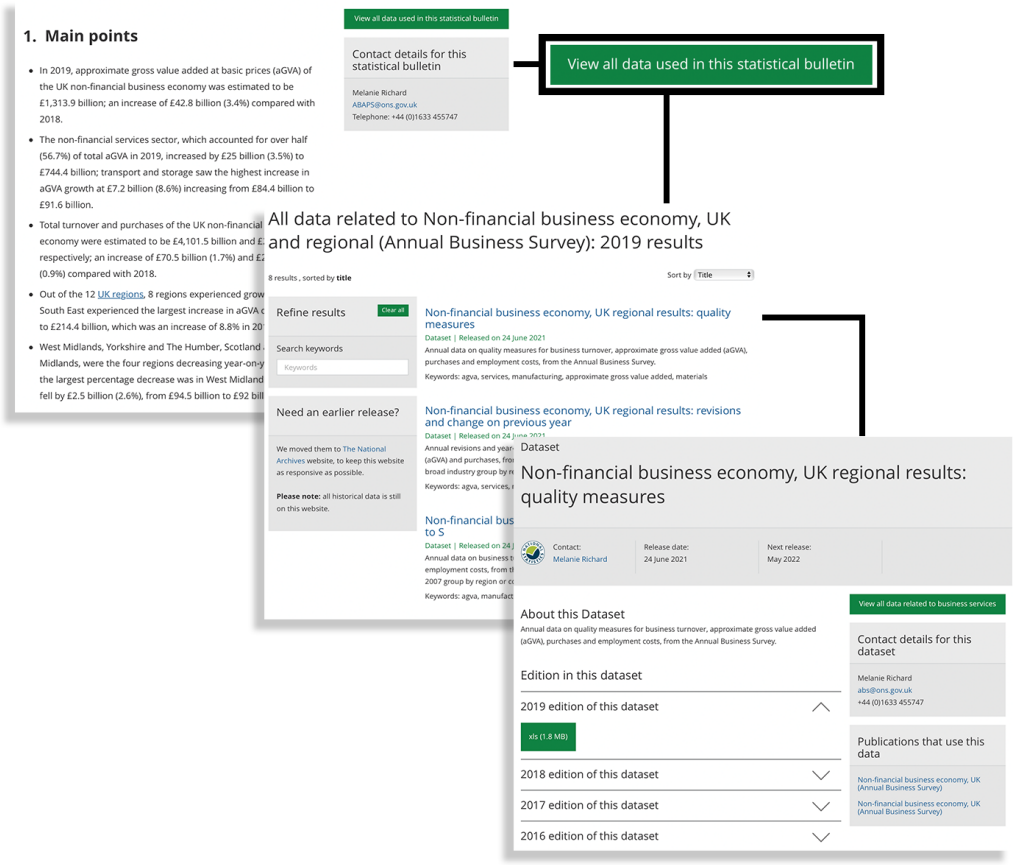


Figure 3.1: Snapshots from the ONS website. (TOP LEFT) Main points ranked by descending importance; (MIDDLE) related data accessible through the *"View all data ..."* prompt; (BOTTOM RIGHT) redirection page for sample dataset with editions (versions) available to download in Excel (`.xls`/ `.xlsx`) format. Bulletin available at: `https://www.ons.gov.uk/businessindustryandtrade/business/businessservices/bulletins/nonfinancialbusinesseconomyukandregionalannualbusinesssurvey/2019finalresults`

3.2 Bulletin Structure

The bulletins follow a consistent structure[1] which is designed to meet user needs and priorities. This rigid structure is an attractive property as it allows us to build a dependable retrieval system around it. We leverage this consistency to log the necessary details and meta-data required for the dataset construction. We discuss the structure of the main points in Section 3.2.1; how related datasets are portrayed in Section 3.2.2; the analysis section of bulletins in Section 3.2.3 and additional information in Section 3.2.4.

3.2.1 Main Points

The main points section highlight the most important and interesting findings from the bulletins at a glance. ONS found[1] that users only spend around 4 minutes when reading bulletins, hence these main points are designed as a stand-alone summary.

By construction, the main points section consists of up to six concisely written points, with each being:

- a single bullet point
- contain one message that is expanded on within the bulletin
- be a single sentence with what happened followed by the significance of this.

The guidelines also indicate that, although not a restriction, main points should be less than 75 words. This is an attractive property of the source data as language models have a maximum capacity of tokens they can process on each step. Usually this is capped[4] to 512 tokens (Herzig, Nowak, et al. 2020). Since we ideally want to pair the claim with a structured table, if we can design our dataset to consist of short claims then the LMs will have more capacity to focus on the structured data.

We present a running example below, Example 3.1, of how the main points are presented.[5]

[4]Longformers for example do not have this limitation, `https://arxiv.org/abs/2004.05150`

[5]Adult smoking habits in the UK: 2019, `https://www.ons.gov.uk/peoplepopulationandcommunity/healthandsocialcare/healthandlifeexpectancies/bulletins/adultsmokinghabitsingreatbritain/2019`

Example 3.1: Main Points Section[a]
Adult smoking habits in the UK: 2019

Main Points:

- In the UK, in 2019, 14.1% of people aged 18 years and above smoked cigarettes, which equates to around 6.9 million people in the population, based on our estimate from the Annual Population Survey (APS)
- The proportion of current smokers in the UK has fallen significantly from 14.7% in 2018 to 14.1% in 2019
- Of the constituent countries, 13.9% of adults in England smoked, 15.5% of adults in Wales, 15.4% of adults in Scotland and 15.6% of adults in Northern Ireland
- In the UK, 15.9% of men smoked compared with 12.5% of women
- Those aged 25 to 34 years had the highest proportion of current smokers (19.0%)

3.2.2 Data used in the bulletin

As presented in Figure 3.1, the button *"View all data ..."* redirects the reader to a new page that lists all datasets that were used to write the bulletin in question. These are usually referred to as "Related Data".

From this exhaustive list, we can navigate to the appropriate download link for each entry, via the HTML code of the page. Historical versions are available for each dataset, but we choose to focus on the most recent/ complete version available to us. The filepath for each dataset follows a consistent naming scheme, reliant on the dataset title. With this observation, automatic collection of the appropriate files was easy once the dataset titles were logged.

It is quickly apparent upon investigation that, the structure between datasets is highly variable. Although there are guidelines determining bulletin structure — there are none for the format of the spreadsheets. The spreadsheets are human readable but their current state makes machine-parsing a hard task.

We describe initial observations from a small sample but more thorough discussion can be found in Section 4.2. Particularly, all spreadsheets investigated are not ready for parsing. For example, we observed that no spreadsheet starts at the top left cell and contains notes in text form in the first few rows. This immediately breaks the detection of where the table data starts. We also observe that column headers are often missing. More importantly, column headers are often spread through multiple rows and cell values which would require some form of processing to infer the actual column name.

3.2.3 Analysis Section

The analysis section of bulletins provides more in-depth details that cannot be communicated in short bullet-points. The authoring statistician explores the data and provides supporting figures

that support the outcomes of the statistical analysis.

Although we attempted to leverage the contents of such sections, we found that the data quality was negatively affected and would diminish the value of our work. We attempted to use processing techniques to filter good quality content from these sections but found that even with aggressive filtering, the quality of claims was highly inconsistent.

We determined that for the sake of data quality, we would not use content from the Analysis Section and instead focus on the Main Points section. We provide a discussion below on how we reached this conclusion.

As stated in the ONS guidelines, the analysis section should:

- Provide a commentary, useful to the majority of readers, discussing in-detail noteworthy or important properties of the data
- Each topic should have their own sections and appropriate subsections
- Warn users of anything that can fundamentally affect the analysis

This section's guidelines are considerably looser[6] as they have to work around varying styles of analyses. During the investigation of bulletins and preliminary analysis of the content within, the analysis sections appear to be unusable without heavy supervision. This and the fact that some bulletins skip the analysis section completely, make structural inconsistencies very prominent within these sections.

The biggest issue we see is the variation of information types discussed within each section, ranging from clarifications about wording, graph descriptors, statistician comments and other general content. This makes automatic detection of what is useful and what is not a non-trivial task and for the large scale of bulletins available this is a significant barrier. Particularly, we see no straightforward way of classifying if each sentence extracted from such section can be used to mimic statistical claims.

To show this disparity between usability of sentences within the analysis sections, we give Example 3.2, a bulletin[7] discussing smoking prevalence in the UK. We encounter sentences that are statistical claims and some which are not:

[6] `https://style.ons.gov.uk/category/statistical-bulletin/main-points-and-analysis`

[7] `https://www.ons.gov.uk/peoplepopulationandcommunity/healthandsocialcare/healthandlifeexpectancies/bulletins/adultsmokinghabitsingreatbritain/2019`

Example 3.2: Analysis Section – Quotes of good and bad quality sentences[a]
Adult smoking habits in the UK: 2019

- **Directly usable:**
  - Country of birth: those who were born in Poland had the highest proportion of current smokers (24.5%), whereas those born in India had the lowest proportion of current smokers (5.3%)
  - Education: those with a degree had the lowest proportion of current smokers (7.3%), which is around a quarter of the proportion among those with no qualifications (29.1%)
  - The proportion of current smokers among Muslim men is 18.4%, whereas among Muslim women this is just 3.9%
- **Not directly usable:**
  - Smoking prevalence estimates by local authority area tend to fluctuate each year because of their small sample sizes producing more statistical uncertainty.
  - To show how smoking status tends to be associated with inequality, we focus on socioeconomic status, based on the National Statistics definition.
  - In Sections 5 and 6, we describe data from the Opinions and Lifestyle Survey (OPN), which cover Great Britain and include adults aged 16 years and above.

The statements deemed "Not directly usable", for the particular Example 3.2, discuss nuances in the data that are not related to statistical claims. We ideally want our statements to include a named entity, such as "*Males*", "*Poland*" or "*Muslim women*". We also want our statements to include a key fact, usually in a form of numerical value or percentage change such as "*24.5%*" or a commentary on the effect such as "*Highest proportion*", "*Increased*" or "*Decreased*". The content of this section do not possess this property throughout. If we cannot guarantee the existence of such key terms in the sentences we need to construct a mechanism that will filter the bad sentences out.

With the aim to provide the most expansive collection of statements available, we attempted to process and filter the content of these sections and see if we can include them in our dataset without affecting the data quality. This entailed taking advantage of language commonly used that was deemed inappropriate and ignoring sentences that included it. We explored pruning methods that removed sentences that: 1) commented on the methodology; 2) clarifications on how estimates were computed[8]; 3) contained hyperlinks; 4) HTML code ; 5) contained ONS-specific words such as statistician comments, contact details and other meta-data were amongst those removed. Details can be found below:

- Methodology:
  - Removed sentences that included words such as: "*statistical uncertainty*", "*our anal-*

[8]To our judgement, these clarifications did not contain language that would aid semantic parsing over tables. They did not contain any indication of what required operation was needed.

ysis", "*see table*", "*additional data*", "*statistically significant*", "*statistically insignificant*", "*socioeconomic status*", "*is defined as*", "*refers to*", "*restricted to*", "*available in*", "*reference table*", "*data collected*", "*data collection*", "*data on*", "*data are*", "*information on*", "*changes can be*".

- Estimates:
  - Removed sentences that included words such as: "*estimates were*", "*estimate was*", "*computed by*", "*estimated by*", "*used to estimate*".
- Hyperlinks:
  - Removed sentences that contained hyperlinks.
- HTML code:
  - We observed that nested paragraphs or lists of certain depth (> 3) usually referred to clarifications on the analysis and were unhelpful to our task. We removed '<p>' and '<li>' tags of this depth.
- ONS-specific keywords:
  - Removed sentences that included words such as: "*contact*", "*ONS*", "*office of national statistics*", "*email*", "*telephone*", "*statistician's comments*", "*note:*", "*clarification:*".

We also explored extracting sentences that contained numerical entities, such as percentages or numbers. This was performed on a small collection and manually inspected. Although this yielded more consistent results, we still observed significant noise in the extracted sentences. This also required the processing of the contents to label each entity which in turn increased the required processing time.

Although both approaches were unsuccessful[9], it helped create a robust indexing system and was beneficial for the general quality of prompts extracted. As a result, the analysis section of each bulletin **was not** used as source for our statistical claims. We focused on extracting cleaner claims through other means instead.

3.2.4 Related Links, Additional Information and Contact Details

Bulletins may end with a final section that includes additional information such as links to related publications or contact information for the authors. This section **was not used** as a source for statistical claims as we did not find its contents useful for our work.

This information could be useful as a mechanism to link publications together, however we did not find immediate use to this relation. Our work's scope did not include investigating how claims between two separate bulletins interact and for this reason we chose to not study this effect further. That said, we still provide a mechanism to link bulletins together by cross-referencing the datasets used between each bulletin. As ONS continuously updates datasets instead of creating new ones, the user can determine if two bulletins are related by checking whether they share dataset sources.

[9]Investigation of content that managed to go through this processing was still largely unusable. Hard-coding bad keywords was not a viable solution and aggressive pruning of content would not solve this issue. Identifying the numeric/ named-entities with spaCy was too time consuming to be viable.

3.3 Application Programming Interface (API)

Originally, our approach was to use the ONS API to get access and filter the data; available to readers here. As of August 2021, this is still in BETA and undergoes continuous improvements. It allows users to search, download and filter its contents by passing query-like commands – promising a fitting solution to the task of collecting all the necessary data we require to build our probe. Unfortunately, we found that the API at its current state, is missing critical functionality. The following subsections investigate how this affects our work. We would like to thank Rob Grant for their time and help on clarifying API functionality and limitations.

In summary, with the discussion that follows in Sections 3.3.1 and onwards, the API cannot be reliably used for our work. Unfortunately so, as if these issues were solved, the automatic collection and maintenance of up-to-date data would be very trouble-free.[10] Additionally, we could have used the API's functionality of retrieving subsets of the data directly, alleviating a big portion of the filtering step needed to identify the relevant context.

3.3.1 No access to bulletins

Currently, the API does not include any information on statistical bulletins. We cannot search for a target bulletin nor infer which one is used through other means (say through category/ subcategory investigation or other publications). Even if the API stores information on datasets, we cannot see what bulletins were written using these datasets. Ideally, we would like to see a clear tree-like structure of category, subcategory with the datasets and related statistical bulletins published; similarly to how its structured in the website. This is currently unavailable. Therefore, to process and collect the bulletins, one must approach the task through the ONS website. This requires a web-scrapper built around the HTML code of the webpage.

3.3.2 Inconsistent naming of datasets

Due to this API limitation, we must rely on the ONS webpage to extract information relating to bulletins. We explored in Section 3.2.2, how the website can be used to see which datasets were used in the bulletins. The critical issue here is that the API stores these datasets using a unique naming scheme that is inconsistent with that of the website. We cannot infer through any HTML element of the website (say the filepath of the data or the dataset title) a way to link the two. Even if[11] the filename expected by the API is related to the subject contents of the dataset, the exact filename string is not consistently constructed. We worked closely with ONS to confirm that there is no consistent way of knowing the required name a-priori. To both our knowledge, we do not see how this can be done without changing either naming scheme.

3.3.3 No way to link bulletins to datasets

With the observation that datasets follow two distinct naming schemes in the API and website; and the fact that no bulletin information is accessible through the API, we do not have any mechanism that allows to connect web bulletins to API data. Even if we use the webpage to construct a

[10] No need to redownload data and keep them up-to-date. Also, fixes could be pushed automatically

[11] This is not always the case and we saw datasets having names such as "*cpih01*"

database with related bulletins and datasets — we cannot then utilise any API functionality as we wouldn't know what to search for. For this to be solved, the two sources should use the same naming scheme or there should be a reference to the other for easy matching.

3.3.4 Incomplete collection of data

Finally, the available collection of datasets through the API is very limited. Even if all the previously discussed issues were solved, we would find that only 41 datasets are available through the API. Compared with the 957 we collected manually, this is a very small subset. It appears that making a dataset available through the API requires processing on the ONS side. This process is slow. During the four month period we queried the API, no changes in the collection of datasets was seen. Thus, we need to rely on extracting the remaining data through other means, effectively limiting the utility of the API. The complete list of datasets available through the API can be found in Appendix A.2.

Chapter 4

Processing, filtering and analysis of data

Discussions on how the statistical bulletins and corresponding data was collected can be found in Section 4.1, how the structured data was processed in Section 4.2, how relevant information was retrieved in Section 4.3. We explore the dataset and provide an analysis in Section 4.4 together with followed evaluation in Section 4.5.

4.1 Bulletin and dataset collection

As the API did not provide the functionality we sought and required to construct our dataset, we wrote a web-scrapper that crawled through the ONS website and logged the necessary information. We utilised the Python libraries *BeautifulSoup*[1] and *lxml*[2] for all the HTML processing. In the following subsections we discuss each step of the process.

In summary, we collect an indexing of all bulletins within a section and subsection. We process each bulletin and store the main points discussed. We identify appropriate named-entities and mask these out to generate our cloze-style questions. Given that the API has limited functionality, we collect the structured data through our web-scrapper, storing each spreadsheet locally together with a reference of bulletins that were written with it. We pair the questions together with the answer, answer type, bulletin and related datasets to formulate our complete lists of cloze questions.

4.1.1 Finding bulletins

To collect the bulletins, we took advantage of how the URLs are constructed and the search functionality of the ONS webpage. The complete list of sections and subsections (which can be found in Appendix A.3) was extracted from the drop-down menus. We then visit each subsection by querying the appropriate URL. Here we leverage the search functionality of the webpage by passing "variables" to our query, such as `sortBy=release_date` and `size=MAXSIZE`. For our work

[1] `https://www.crummy.com/software/BeautifulSoup/`
[2] `https://lxml.de`

we set[3] `MAXSIZE` at 1000 but the actual retrieved number is considerably less (382). We also pass an empty query string `query=` which returns all available bulletins.

```
https://www.ons.gov.uk/SECTION/SUBSECTION/
    publications?sortBy=release_date&query=&size=MAXSIZE
```

For example, if we wish to extract the bulletin we give in Example 2.2, we would look under the "*People, Population and Community*" section, and "*Health and Social Care/Health and Life Expectancies*' subsections.

Our query would be:

```
https://www.ons.gov.uk/peoplepopulationandcommunity/
    healthandsocialcare/healthandlifeexpectancies/
        publications?sortBy=release_date&query=&size=1000
```

This query would output a list of bulletins. We investigate the HTML content and see that each bulletin URL is stored under a unique `<a>` tag, with a specific attribute `data-gtm-uri:True`. This forces to look into elements that redirect with a link to another page. The bulletins are the only ones that have both properties. We process all four main sections, yielding a complete indexing, with the bulletin title, bulletin section/ subsection as well as their respective URLs.

4.1.2 Processing and storing bulletin content

With this exhaustive list of statistical bulletins in our indexer, we process each one singly to retrieve the relevant contents within. We look for specific combinations of `<div>` and `<p>` tags of certain nested depth. The appropriate elements also share specific attributes such as `id:"main-points"` and `class:"section__content--markdown"`. We extract all points that are nested within `<li>` tags. The consistency in the bulletin structure we discussed closely in Section 3.2.1 now plays a very important role, as we can be sure that we do not extract irrelevant content nor do we miss any statements of interest.

Additional processing of the extracted strings is needed, as we observed non-standard characters within the contents. We utilise regex to remove special[4] and unicode characters. Here, we process the contents similarly as in Section 3.2.3 to remove hyperlinks and bad key-words.

We follow the methodology established in (Lewis et al. 2019) to construct our named entity recogniser. Named entities fall into 20 distinct labels, which we aggregate into 5 more general labels. Table 4.4 contains the complete list of available labels together with the aggregate labels we use.

We understand that these labels are based on predictions of an underlying model and for this reason they might sometimes fail. We investigate the effect of this in Section 4.5.1

As the NE we identify in these sentences, are the ones that will be masked out — they are thus the answers to our cloze questions. We showcase the distribution of each NE type and in turn that of the answer types of our dataset in Figure 5.1. As the bulletins provided by ONS describe statistical data, the answers are predominantly *NUMERICMASK* and *TEMPORALMASK* types.

[3]The website often crashes when setting this number arbitrarily large, say 9999

[4]We do not consider currency nor percentage symbols as special characters as they are key-drivers in our work

NER (spaCy)	Description	Aggregate Label
PERSON	People, including fictional	IDENTITYMASK
NORP	Nationalities, religious/ political groups	IDENTITYMASK
ORG	Companies, agencies, institutes ...	IDENTITYMASK
FAC	Buildings, airports, highways, ...	PLACEMASK
GPE	Countries, cities, states	PLACEMASK
LOC	Non-GPE locations, mountain ranges, bodies of water	PLACEMASK
PRODUCT	Objects, clothes, food, ...	THINGMASK
EVENT	Named hurricanes, battles, wars, sport events, ...	THINGMASK
WORK_OF_ART	Titles of books, songs, ...	THINGMASK
LAW	Named documents made into laws	THINGMASK
LANGUAGE	Any named language	THINGMASK
DATE	Absolute or relative dates or periods	TEMPORALMASK
TIME	Times smaller than a day	TEMPORALMASK
PERCENT	Percentages, including %	NUMERICMASK
MONEY	Monetary values, with units	NUMERICMASK
QUANTITY	Measurements	NUMERICMASK
ORDINAL	"first", "second", ...	NUMERICMASK
CARDINAL	Numerals that don't fall under other types	NUMERICMASK

Table 4.1: Named entity recognition from spaCy: original tags and aggregate labels

We understand that the generated data is largely revolving around numerical or date questions and is not balanced for all other types. We do not consider this an issue as, evidently, these are the types of questions usually found in statistical reports. Given that we provide the appropriate labelling, it is easy to balance the dataset if required by the user but we do not perform this ourselves. We believe that statements of different answer types are beneficial to benchmarking the quality of proposed solutions as they require slightly different reasoning to answer them.

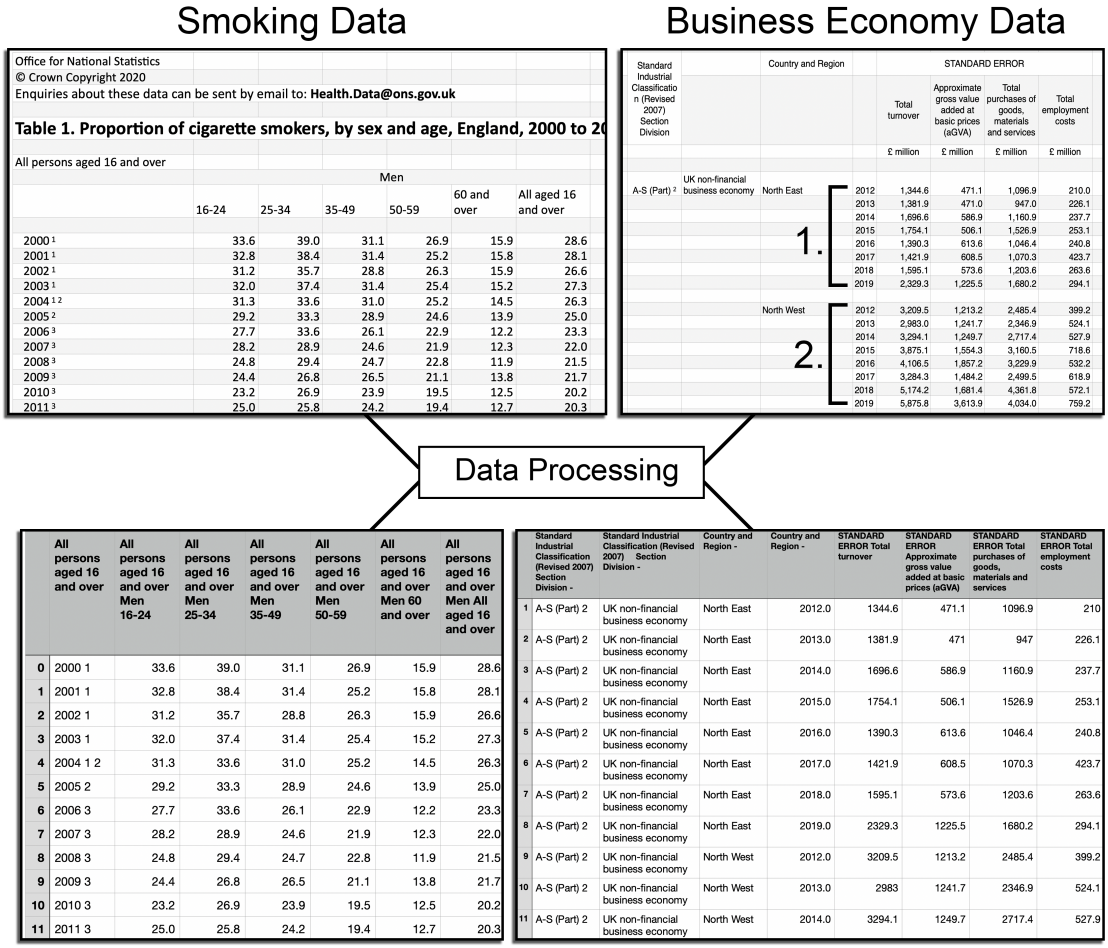
4.1.2.1 Storing related datasets

As we observed in Section 3.2.2, each bulletin is paired with the datasets that were used to generate its content via the "*View all related data*" button. For a complete list of all the related data, one appends `/relateddata` to the bulletin URL. We can then extract the dataset titles and URLs in the same way as we did for our bulletins. The website search functionality works largely the same, whether you are searching for bulletins or datasets. Once we have the dataset URLs we visit each one and search for a specific HTML tag, `data-gtm-uri:True`, that specifies that the dataset links to somewhere. This points to the Excel file we seek. We store each one locally with the naming scheme `datasets/section_subsection_title.xls`.

4.2 Structured data processing

Investigating the stored collection of spreadsheet data, quickly showcases the structural inconsistencies between tables. We touched shortly upon the structure of the ONS spreadsheets in Section 2.9 but we provide further details here. We separate these in subsections based on the processing required. We treat each sheet/ page within each data file (`.xls`) as its own entity.

Throughout this work, we were required to take initiative on how to handle filtering/ impu-



Office for National Statistics
© Crown Copyright 2020
Enquiries about these data can be sent by email to: **Health.Data@ons.gov.uk**

Table 1. Proportion of cigarette smokers, by sex and age, England, 2000 to 20

All persons aged 16 and over

	Men					
	16-24	25-34	35-49	50-59	60 and over	All aged 16 and over
2000 [1]	33.6	39.0	31.1	26.9	15.9	28.6
2001 [1]	32.8	38.4	31.4	25.2	15.8	28.1
2002 [1]	31.2	35.7	28.8	26.3	15.9	26.6
2003 [1]	32.0	37.4	31.4	25.4	15.2	27.3
2004 [1 2]	31.3	33.6	31.0	25.2	14.5	26.3
2005 [2]	29.2	33.3	28.9	24.6	13.9	25.0
2006 [3]	27.7	33.6	26.1	22.9	12.2	23.3
2007 [3]	28.2	28.9	24.6	21.9	12.3	22.0
2008 [3]	24.8	29.4	24.7	22.8	11.9	21.5
2009 [3]	24.4	26.8	26.5	21.1	13.8	21.7
2010 [3]	23.2	26.9	23.9	19.5	12.5	20.2
2011 [3]	25.0	25.8	24.2	19.4	12.7	20.3

Standard Industrial Classification (Revised 2007) Section Division		Country and Region		STANDARD ERROR			
				Total turnover	Approximate gross value added at basic prices (aGVA)	Total purchases of goods, materials and services	Total employment costs
				£ million	£ million	£ million	£ million
A-S (Part) [2]	UK non-financial business economy	North East	2012	1,344.6	471.1	1,096.9	210.0
			2013	1,381.9	471.0	947.0	226.1
			2014	1,696.6	586.9	1,160.9	237.7
			2015	1,754.1	506.1	1,526.9	253.1
			2016	1,390.3	613.6	1,046.4	240.8
			2017	1,421.9	608.5	1,070.3	423.7
			2018	1,595.1	573.6	1,203.6	263.6
			2019	2,329.3	1,225.5	1,680.2	294.1
		North West	2012	3,209.5	1,213.2	2,485.4	399.2
			2013	2,983.0	1,241.7	2,346.9	524.1
			2014	3,294.1	1,249.7	2,717.4	527.9
			2015	3,875.1	1,554.3	3,160.5	718.6
			2016	4,106.5	1,857.2	3,229.9	532.2
			2017	3,284.3	1,484.2	2,499.5	618.9
			2018	5,174.2	1,681.4	4,361.8	572.1
			2019	5,875.8	3,613.9	4,034.0	759.2

	All persons aged 16 and over	All persons aged 16 and over Men 16-24	All persons aged 16 and over Men 25-34	All persons aged 16 and over Men 35-49	All persons aged 16 and over Men 50-59	All persons aged 16 and over Men 60 and over	All persons aged 16 and over Men All aged 16 and over
0	2000 1	33.6	39.0	31.1	26.9	15.9	28.6
1	2001 1	32.8	38.4	31.4	25.2	15.8	28.1
2	2002 1	31.2	35.7	28.8	26.3	15.9	26.6
3	2003 1	32.0	37.4	31.4	25.4	15.2	27.3
4	2004 1 2	31.3	33.6	31.0	25.2	14.5	26.3
5	2005 2	29.2	33.3	28.9	24.6	13.9	25.0
6	2006 3	27.7	33.6	26.1	22.9	12.2	23.3
7	2007 3	28.2	28.9	24.6	21.9	12.3	22.0
8	2008 3	24.8	29.4	24.7	22.8	11.9	21.5
9	2009 3	24.4	26.8	26.5	21.1	13.8	21.7
10	2010 3	23.2	26.9	23.9	19.5	12.5	20.2
11	2011 3	25.0	25.8	24.2	19.4	12.7	20.3

	Standard Industrial Classification (Revised 2007) Section Division -	Standard Industrial Classification (Revised 2007) Section Division -	Country and Region -	Country and Region -	STANDARD ERROR Total turnover	STANDARD ERROR Approximate gross value added at basic prices (aGVA)	STANDARD ERROR Total purchases of goods, materials and services	STANDARD ERROR Total employment costs
1	A-S (Part) 2	UK non-financial business economy	North East	2012.0	1344.6	471.1	1096.9	210
2	A-S (Part) 2	UK non-financial business economy	North East	2013.0	1381.9	471	947	226.1
3	A-S (Part) 2	UK non-financial business economy	North East	2014.0	1696.6	586.9	1160.9	237.7
4	A-S (Part) 2	UK non-financial business economy	North East	2015.0	1754.1	506.1	1526.9	253.1
5	A-S (Part) 2	UK non-financial business economy	North East	2016.0	1390.3	613.6	1046.4	240.8
6	A-S (Part) 2	UK non-financial business economy	North East	2017.0	1421.9	608.5	1070.3	423.7
7	A-S (Part) 2	UK non-financial business economy	North East	2018.0	1595.1	573.6	1203.6	263.6
8	A-S (Part) 2	UK non-financial business economy	North East	2019.0	2329.3	1225.5	1680.2	294.1
9	A-S (Part) 2	UK non-financial business economy	North West	2012.0	3209.5	1213.2	2485.4	399.2
10	A-S (Part) 2	UK non-financial business economy	North West	2013.0	2983	1241.7	2346.9	524.1
11	A-S (Part) 2	UK non-financial business economy	North West	2014.0	3294.1	1249.7	2717.4	527.9

Figure 4.1: Processing of tabular data to reduce structural artefacts. LEFT side shows data from Example 3.1. RIGHT side shows data from Figure 3.1. LEFT side shows text notes in the first few rows which need to be removed. Also shows how column headers are spread through multiple rows and cell values and thus require some processing to deduce correct header. RIGHT side shows the two sub-tables within the spreadsheet and how forward fill works for cell values. Both sides show examples where empty rows and columns were removed.

tation methods. It is clear that there is no one-off solution to the trade-off between data quality and processing. We describe how we balance this trade-off and determine appropriate parameters in Section 4.5.2.

Although we aim to remove noise within our data, we understand how its raw form enhances the benchmarking difficulty. Given that it is less likely to stumble upon tidy and ready-for-parsing data, we performed processing only when we are confident of its outcome. We also wrote our code-base in a modular way such that any processing done can be easily swapped in or out. We provide the raw data but also provide the code that performs the following processing.

4.2.1 Preliminary pre-processing

We first drop any row or column that is completely empty. These are often there for aesthetic reasons such as separating columns when feature-types change or rows where multiple sub-tables are within the same spreadsheet. We then store a copy of the current state of the spreadsheet for future use. We refer to this copy as the "original table", even if these rows/ columns are removed.

Any spreadsheet that is made up by less than 3 columns or rows is deemed unusable. We do not perform any modification of cell values other than identifying missing values and replacing them with an empty string. We also observed spreadsheets where the content was not an actual table but rather an image of a table pasted within. We cannot process these files and are thus ignored. Further discussion on how these were derived is followed in Section 4.5.2.

4.2.2 Identifying where a table starts

Extensive investigation has shown that no entry in the studied collection starts in a clean format, with appropriate column headers and their respective values. Instead, we observed that most start with unrelated commentary or text notes. These might be hyperlinks to appropriate publications, buttons that redirect to other parts of the file or general comments about who collected and managed the data. We deem this information unrelated to the task.

This immediately reveals the need of a detection mechanism that identifies the start of the table. Without this, machine parsing the raw tables leads to an unusable dataset as the number of columns and rows, column headers and cell values are broken. Figure 4.1 shows the Smoking Data used in the Example 3.1 which demonstrates this issue.

In its unprocessed form, the first 5 rows need to be removed. The first 3 are trivial to remove as only a single cell is populated. We hypothesise that a row that only contains a single value is not critical to the data.[5] The fourth is removed automatically in the pre-processing step as it is completely empty. The fifth row, which acts as a spreadsheet title, has multiple cells populated as the title is split between cells. This is an uncommon writing style, however it occurs frequently within the ONS data. It is done arguably for legibility reasons but at the same time reduces the quality of the data drastically, as values are split between multiple cells.

To work around these artefacts, we take an approach of inferring these rows based on how many empty values each one has. Our hypothesis is that, the usable parts of the data are dense tables, with minimal empty values. The rows which are mostly empty but have few cells populated are usually there to support the data rather than be a critical part of it. To understand how many cell values a row should have to be considered unrelated, we utilise the table size to compare.

Particularly, we compare the number of columns with the median number of empty values in the table rows. We use the median when counting the number of empty cells due to its robustness to skewed distributions, as the mean frequently missed rows when tables had very few unusable rows. We count empty values in all rows and drop those which are 1.5 times more than the median number of empty values in a given row. In short, we consider rows that contain more than 1.5 times the median number of empty values to be noisy rows. A more detailed discussion on how this was derived is followed in Section 4.5.2. We provide a configurable processing pipeline complete with configs for the user to modify.

[5] We did not observe entries where the opposite holds

We did observe that sometimes the median value is zero, when most rows are non-empty. This would break our detection mechanism, as we multiply 1.5×0 to determine the threshold. Thus we need an additional catch. In this case, we default to using the number of columns to check for rows that have 70% missing values. If they do, we remove them.

With this processing we have removed rows which are not part of the main body of the data. It is now important to identify the first row that corresponds to the column headers. We sequentially process each row and utilise how column headers are specified when using Python and Pandas[6].

If a row does not have an assigned header, then Pandas automatically sets its name as "Unnamed: X". We use this to our advantage and progressively search for the first row that will not contain any "Unnamed" entries. Rows that partially contain column headers and empty cell values, like the multi-level headers in Business Economy Data, are assigned "NaN" instead. We can thus infer when we reached the first row that corresponds to column headers — and hence the start of the data.

4.2.3 Managing column headers

We have now identified where the data starts. We are left with a dense table, with no empty rows or columns. Unfortunately though, it is usually the case that with this processing we either have "NaN" values in some headers or missed some column headers that were split over multiple rows.

Imputing the "NaN" values is more straight-forward than identifying if we missed multi-level headers. Given that we sequentially process rows from top to bottom, we can look onward — using the value of the cell below the missing entry as a replacement. If the corresponding cell below is empty, then no imputation is performed. This works as expected, except cases where the column does not have a column value by design. We then wrongly impute a cell value as a column header. Discussion on this follows in Section 4.5.2.

Identifying if we missed column headers during our pre-processing requires more work. These missed multi-level headers, are made up by rows that are not populous enough to be considered part of the main body and are removed by our processing. Figure 4.1 again shows this effect, as the Smoking Data has multi-level column headers. Observe how the *Men* header has 6 subcategories such as *16-24*, *25-34* and so on.

We hypothesise that, if we missed the appropriate column headers, then they must be in the previous few rows we removed given we proceed from top to bottom. We thus use the "lost content" which is made up by looking at the rows before the start of the data. This "lost content" still requires processing as it would contain the same notes and extra meta-data we are trying to remove.

We take a similar approach but now calculate the median number of missing values only for this smaller collection of rows. Again we use this median value to compare, removing all rows with more empty values than the median. Here we do not use any thresholds. Rather, we hard set any row that is above the median in terms of number of empty values as unrelated and is removed. We also remove rows that only have a single cell populated, which are commonly the notes we previously described. This leaves us with only rows that correspond to the lost column headers.

We then need to aggregate these into our current column headers. To do this, we first forward

[6] `https://pandas.pydata.org/`

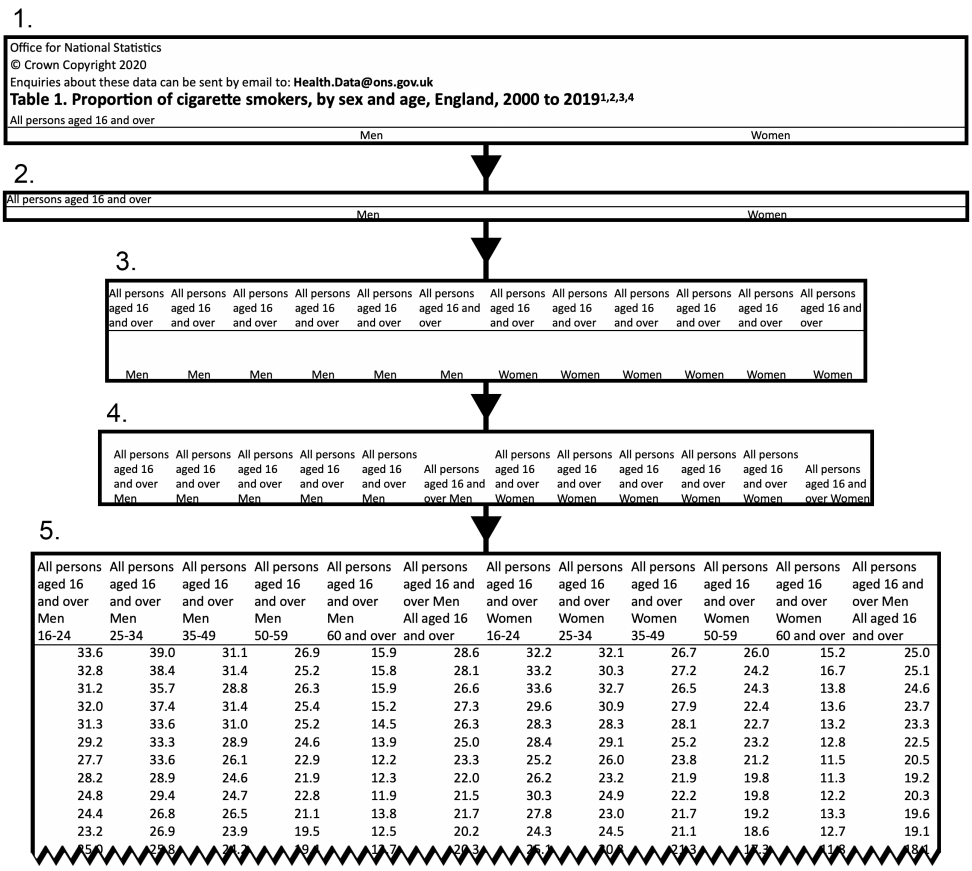


Figure 4.2: Steps to reconstruct lost column headers. STEP ONE shows the lost-content which is removed when establishing the start of the table. STEP TWO shows the two rows that are left after identifying and removing unrelated content such as notes and contact information. STEP THREE shows forward-filling the missing values row-wise. STEP FOUR shows aggregating the rows into a single row, top to bottom. STEP FIVE shows the merging of our inferred labels onto the current column headers.

fill all empty cells row-wise. This means repeatedly filling all empty values with the previous until a new value is seen. This can be clearly seen in STEP THREE of Figure 4.2, how the *Men* and *Women* column headers are turned from single cells to six. We then combine all rows together, merging the cells column-wise, from top to bottom.

We end up with a single row, as seen in STEP FOUR. These labels then are passed onto the processed table, aggregating them to the current column headers as in STEP FIVE. Whereas previously the headers would be *16-24* unidentifiable between *Men* and *Women* categories — we now have *All persons aged 16 and over Men 16-24* and *All persons aged 16 and over Women 16-24*.

4.2.4 Managing sub-tables

It is frequently the case that the ONS spreadsheets are comprised of smaller sub-tables. These are few rows long, representing data of particular groups with similar properties. For example, the Business Economy data in Figure 4.1 shows sub-tables for regions of *North East* and *North West* of the UK. These are smaller dense tables inside a bigger collection.

The issue here is not the presence of the sub-table per se, rather how some categorical features like the column "Country and Region" are constructed. Observing Figure 4.1, we can see that *North East* and *North West* values are only populated once on the first row of the sub-table. Ideally we would like each row to have features filled-in according to its properties. These aesthetic and writing choices are easy for humans to adapt to.

A reader can easily identify that, *North West* corresponds to the whole sub-table even if not explicitly stated. Unlike when parsed by a machine however, as table parsing works in a row-wise schedule, valuable information will be inevitably misrepresented. Specifically, rows that do not have "Country and Region" or similar columns filled, will be treated as if this information is missing — even if we, as human readers, can infer the implied properties. This is an issue, especially for our work, as we will be using these tables in their linearised form.

Without additional processing these types of rows will be unidentifiable. We need to adjust then our processing pipeline to work around these sub-tables and fill them accordingly. We take two approaches to solve this. We first observed that, tables have their most low-level features on the left-most side. Progressively, these turn to higher level features. For example, you would see the first column identifying the source of the data, the second being the type of industry, third country and fourth region. This hierarchy is somewhat natural we argue, given how this data originates from a country that largely reads from left to right. For this reason, we believe that the left-most columns are more likely to describe categorical features and likely to be more empty than required. If we can identify that this is the case, then our imputation can be more aggressive.

We also try to design a secondary method to catch columns that need to be filled but do not fall directly to the description above. To do this, we are inspired by how we handled the identification of the column headers. We want to identify which columns are populated enough to not be considered noise, but not dense enough to be within the main body of the data.

After we have cleaned the data by removing empty columns and rows, after determining the start of the main body and the appropriate headers — we count the number of missing values each column has. We then identify which columns are different from the others by comparing the corresponding number of missing values. We use the number of rows of the whole table as a regulariser, as we need to be able to identify these columns regardless of the original size of the table.

If we assume that most spreadsheets share the same stylistic approaches, then we can expect a single value for every sub-table present. We hypothesise that, the columns which we describe above, with categorical features that are not populated as expected, will be considerably less populated than the main body but at the same time contain more values than other noisy or empty columns.

Particularly, we first look at the total number of columns in the spreadsheet. If there are more than 10 columns, then we consider the first three as potentially in need of imputation. If not, then

imputation only follows if they are empty more than 85% of the total number of rows. Imputation is carried by forward filling the empty cells. That is, we continuously assign the empty cells as the previously known value, from top to bottom, until a new entry is seen. This would fill the first sub-table in Figure 4.1 with *North East* in the "Country and Region" column and *North West* for the second.

Identifying which columns fall under this category is something that requires a tailored solution and is unlikely to works out-of-the-box for all data. With this, we acknowledge how further tuning would be ideal, in a per-spreadsheet scenario. However, our approach does reliably identify columns that describe sub-table properties and imputes them consistently. We err on the side of caution though and try to be conservative with our imputation. Refer to Section 4.5.2 for how the hyper-parameter was derived.

4.2.5 Ongoing issues

While establishing what processing would be required in our data, we realised that there were few consistent issues we could not work around.

We first observed that, many columns with "obvious" entries frequently have empty column names. Specifically, in the Smoking Data of Figure 4.1, the first column which has entries "*2000*", "*2001*", ..., "*2019*" and so on can be easily inferred that they represent years or dates. Similarly, as seen in the Business Economy data, the second column has values "*UK non-financial business economy*", "*Agriculture, forestry and fishing*" and so on, hinting that the column represents types of industries. In both cases, the column names are omitted. It appears that the authors expect readers to infer these themselves as they are "obvious" to a certain extent, but non-trivial for machine approaches.

It is also apparent, in both processed cases of Figure 4.1, how unnamed columns are assigned false titles. The issue here is that when we forward fill, as in STEP THREE of Figure 4.2, the column will be treated as if it is part of a multi-level column, similarly to how *Men* and *Women* headers are handled. This is true when the unnamed column is preceded by a named one. If that is the case then we falsely impute the previous value as the column name. We did not figure a consistent way of detecting such columns as searching for an empty string does not suffice. Both columns with missing titles and those under a multi-level header would have empty strings as headers and hence cannot be distinguished this way.

Secondly, we have seen occurrences where single values were split between cells. This could have been easily rectified by enabling a text-wrapping functionality in the spreadsheet reader of choice, instead of hard-coding values in multiple cells. This is a very big issue we do not see any obvious way of rectifying automatically. Further in the Business Economy data (not shown in figure), we see entries that correspond to multiple industries aggregated together. This leads to cell values being very long texts, with entries such as "Agriculture, Fishing, Production, Construction, Distribution and Non-financial Services". The authors unfortunately split these into few cells to enable easier reading but at the same time completely change the implied cell meaning.

Thirdly, we have observed that some cell entries contain footnote references, such as the Smoking Data of Figure 4.1. Given that all cells are passed as strings, we cannot distinguish when the footnote reference number is an actual data point value and when it is a reference. There is no

distinguishing factor that separates an entry like "2019 1" and "2019[1]" after parsing.

In our judgement, these issues are natural if there are no strict guidelines on how each spreadsheet should be generated. It appears that multiple people are in charge of handling the data generation and follow mismatched style specifications.[7] We also believe that the fact that these datasets are so inconsistently structured hints that their primary use is not advanced analysis rather more one-off inspections by readers.

That said, it is an obstructive limitation to require all datasets to follow a very strict style and structure. As language is a mode used to transfer information, spreadsheets play a similar role — meant to communicate this statistical information to the reader. Given that we do not require language to be rigid in structure, we should be able to define and work around these issues.

We investigated historical entries for few key examples and observed that the structure frequently changes. This means that our solution should be general enough to work for all but at the same time specific enough to capture all the intricacies within. Without an automatic data generation method these issues will occur. We also like to point out that it appears that a motivating factor for the creation of the API was to enable more advanced analysis to be carried easily. Unfortunately as discuss previously, this is yet to be available.

4.3 Filtering for relevant content

Encoding all of the contents using a resource-heavy Transformer is both computationally intractable and likely not necessary (Yin et al. 2020). Given that language models have a finite limit on the number of tokens they can process, we need to make sure the context we provide is relevant and brief. If for example we provide a sentence of 50 tokens then we are left only with $512 - 1 - 50 - 1 = 460$ tokens allocated for the context.[8] Thus, even if the whole spreadsheet is passed, only the first 460 tokens will be put to use. Therefore the quality of the answers heavily relies on how good the filtering is. Our table filtering was inspired by (Schlichtkrull et al. 2020; Yin et al. 2020). We split this into two steps. We identify relevant columns and then identify relevant rows.

4.3.1 Identifying relevant columns

To identify the most relevant columns, we compute the TF-IDF representation of the cloze sentence as well as that of the column headings. We use bi- and tri-gram TF-IDF representations and average over their final similarity. Instead of computing the cosine similarity between the named entities and the table representation like (Schlichtkrull et al. 2020), we compute the length-normalised global similarity between cloze sentence and the table. This was because we did not have access to the named entity spans from (W. Chen et al. 2020). We rank the column similarities and return the top 5 most relevant. Additionally we default[9] to using the first three columns

[7] We investigated datasets from the same bulletins and saw very different inter-bulletin structure as well as multiple contact details for different subsheets.

[8] Breakdown: 512 token capacity, 1 [CLS] sentence-level classification token, 1 [SEP] token to separate sentence from context and 50 tokens making up the sentence.

[9] This hints back to the low-level property argument of Section 4.2.4; it is common that the cosine similarity of the first few column is zero while they are still relevant; we thus default to always including them.

regardless of their similarities.[10] We then are left with at most 8 columns.

We observed that throughout the ONS data, column headers are often abbreviated. Dictionaries that link abbreviations to their expanded form are usually available either in the original bulletin or within the first data sheet. We did not investigate this path further as it is not clear where each abbreviation is stored — making automatic replacement non-trivial. Implementation was performed with `sklearn.feature_extraction.text.TfidfVectorizer` — details can be found in the repository.

4.3.2 Identifying relevant rows

We are motivated to reduce the table further as the contents provide more detail about the semantics of a column than just its header — which is often ambiguous. We only consider the contents of the columns we filter in the previous step. We approach this as before, utilising bi- and tri-gram TF-IDF representations, to construct the similarity metric between row and cloze sentence. We do not perform any length normalisation here as we consider rows to be of equal length. We take the average between the bi-gram and tri-gram similarity as the final representation of similarity. (Yin et al. 2020) also investigate the avenue of generating "synthetic" rows, where they select cell values from multiple rows. While they claim that relevant content can appear in multiple rows, we find this inappropriate as it can be trivial to generate context that supports the cloze sentence even if factually incorrect. We do see how synthetic rows can be useful, however we argue that they require independent encoding that informs the model of their artificial nature. We do not consider this for our work.

4.3.3 Lemmatisation

We observed that exact or approximate string matching between the cloze sentence and the table contents is unlikely. We attempt to lemmatise both the table and the cloze — alleviating the effect of different conjugation forms. We understand that the quality of lemmatisation depends on the correct identification of the part-of-speech, however we only considered the `en-core-web-trf` model due to its reported highest performance.

Carrying out the table content lemmatisation during inference time requires considerable computation time. It is possible to pre-lemmatise and store the processed versions of the table data once and operate on them exclusively. However, we should be able to design a solution that can do this on the fly. The aim of performing lemmatisation is to improve the table filtering thus cannot be performed on the smaller tables.

We have observed that even after lemmatisation the change in extracted content is often negligible or even absent. We attribute this to the fact that the majority of table contents are either numerical data (NUMERICMASK as in Figure 5.1) or unseen tokens related to ONS (such as acronyms like "*aGVA*" or domain-specific keywords "*A-S (Part)*" as in Figure 4.1) which cannot be easily lemmatised. This hints towards a bigger issue: given that the table contents are frequently made up by numbers or domain-specific keywords, how can our retrieval and language models reason over them if an exact match isn't available?

[10] These are excluded during the TF-IDF calculation.

Lemmatisation then is largely unsuccessful, increasing computation time drastically while not benefiting the retrieved content. Continuing further, we do not perform lemmatisation on the dataset when running our models.

4.3.4 Relevant worksheets within spreadsheet

Each spreadsheet we collect is made up by multiple worksheets. Each one contains different data with different use cases. It is important to identify the most relevant one. Supplying the models with context that is irrelevant can be detrimental to its performance since they operate under the assumption that the context contains the ground truth (D. Chen et al. 2017).

We observe that worksheets are frequently named with a numerical scheme. They do not usually contain text strings that are descriptive of their contents. To explain the worksheet names, ONS usually provides a contents worksheet that contains a dictionary with descriptions for each one. However, it is not immediately clear which worksheet this will be.

Although commonly found in the first two worksheets, content worksheets can appear at the end. Additionally, some datasets see continuous changes after being published. Each change is followed by a clarification note that records the modifications. These are usually appended to the start of the spreadsheet, essentially moving the content worksheet further down.

If the content worksheet is named, we can look out for it and try to extract the naming scheme dictionary. Otherwise, we need to process each one singly and identify if it contains this information. To add to this, sometimes the ground truth is found in worksheets that are not named appropriately. This happens usually when the cloze statement is about a general observation and the answer lies in a summary worksheet. It is thus not clear how to handle these cases.

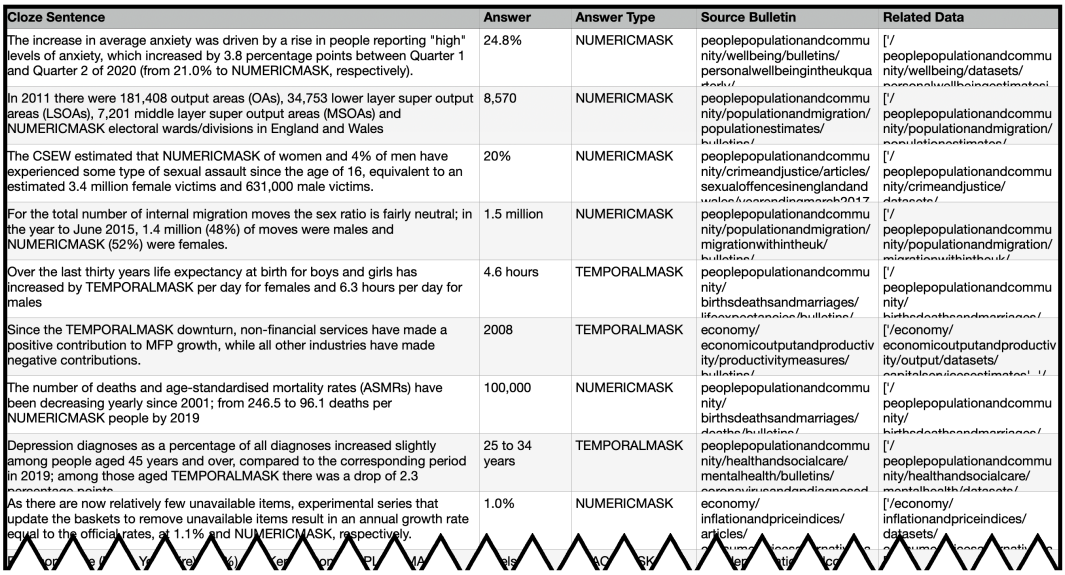
Inspired also by (Holtzman et al. 2021), it can be inadequate to assume that the worksheet with the highest relevancy metric/ probability will be the one of interest, due to surface competition. In the case of "Smoking Habits", we have two worksheets that both describe the usage of e-cigarettes in the United Kingdom. One provides monthly numbers and the other provides percentages. If we are tasked with answering a cloze statement about the global population and we are unaware if we should provide a count or a percentage, both seem equally relevant. Understanding which one contains the ground truth then is non-trivial.

For this reason, we do not take this path. Rather, we process each worksheet independently and generate predictions from all. We then consider our prediction correct if any worksheet produced the correct answer.

4.4 Exploratory analysis of the data

In this section we explore the data we collect and discuss noteworthy properties. We provide accompanying graphics that showcase the data and can help the reader understand the nature of the content.

In total, we processed a collection of 382 statistical bulletins which align with 967 unique datasets. Each bulletin is paired with their corresponding sub-collection of relevant datasets. Exact details can be found in Table 4.2.

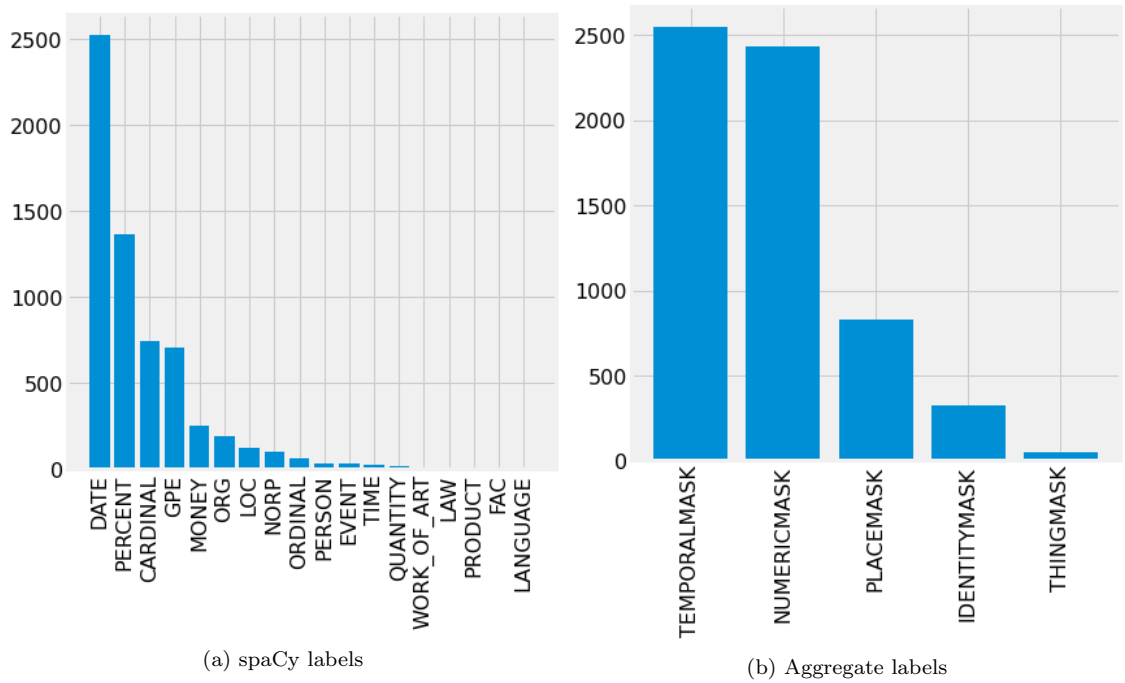


Cloze Sentence	Answer	Answer Type	Source Bulletin	Related Data
The increase in average anxiety was driven by a rise in people reporting "high" levels of anxiety, which increased by 3.8 percentage points between Quarter 1 and Quarter 2 of 2020 (from 21.0% to NUMERICMASK, respectively).	24.8%	NUMERICMASK	peoplepopulationandcommunity/wellbeing/bulletins/personalwellbeingintheukquarterly/	['/peoplepopulationandcommunity/wellbeing/datasets/personalwellbeingestimatesi
In 2011 there were 181,408 output areas (OAs), 34,753 lower layer super output areas (LSOAs), 7,201 middle layer super output areas (MSOAs) and NUMERICMASK electoral wards/divisions in England and Wales	8,570	NUMERICMASK	peoplepopulationandcommunity/populationandmigration/populationestimates/bulletins/	['/peoplepopulationandcommunity/populationandmigration/populationestimates/
The CSEW estimated that NUMERICMASK of women and 4% of men have experienced some type of sexual assault since the age of 16, equivalent to an estimated 3.4 million female victims and 631,000 male victims.	20%	NUMERICMASK	peoplepopulationandcommunity/crimeandjustice/articles/sexualoffencesinenglandandwales/yearendingmarch2017	['/peoplepopulationandcommunity/crimeandjustice/datasets/
For the total number of internal migration moves the sex ratio is fairly neutral; in the year to June 2015, 1.4 million (48%) of moves were males and NUMERICMASK (52%) were females.	1.5 million	NUMERICMASK	peoplepopulationandcommunity/populationandmigration/migrationwithintheuk/bulletins/	['/peoplepopulationandcommunity/populationandmigration/migrationwithintheuk/
Over the last thirty years life expectancy at birth for boys and girls has increased by TEMPORALMASK per day for females and 6.3 hours per day for males	4.6 hours	TEMPORALMASK	peoplepopulationandcommunity/birthsdeathsandmarriages/lifeexpectancies/bulletins/	['/peoplepopulationandcommunity/birthsdeathsandmarriages/
Since the TEMPORALMASK downturn, non-financial services have made a positive contribution to MFP growth, while all other industries have made negative contributions.	2008	TEMPORALMASK	economy/economicoutputandproductivity/productivitymeasures/bulletins/	['/economy/economicoutputandproductivity/output/datasets/capitalservicesestimates', '/
The number of deaths and age-standardised mortality rates (ASMRs) have been decreasing yearly since 2001; from 246.5 to 96.1 deaths per NUMERICMASK people by 2019	100,000	NUMERICMASK	peoplepopulationandcommunity/birthsdeathsandmarriages/deaths/bulletins/	['/peoplepopulationandcommunity/birthsdeathsandmarriages/
Depression diagnoses as a percentage of all diagnoses increased slightly among people aged 45 years and over, compared to the corresponding period in 2019; among those aged TEMPORALMASK there was a drop of 2.3 percentage points	25 to 34 years	TEMPORALMASK	peoplepopulationandcommunity/healthandsocialcare/mentalhealth/bulletins/coronavirusanddiagnosedmentalhealthandwellbeing/	['/peoplepopulationandcommunity/healthandsocialcare/mentalhealth/datasets/
As there are now relatively few unavailable items, experimental series that update the baskets to remove unavailable items result in an annual growth rate equal to the official rates, at 1.1% and NUMERICMASK, respectively.	1.0%	NUMERICMASK	economy/inflationandpriceindices/articles/	['/economy/inflationandpriceindices/datasets/
P[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Figure 4.3: Probe snapshot

Statistic	frequency
Num. of bulletins	382
Num. of datasets	957
Num. of clozes	6154
	mean (std)
Average Num. of datasets per bulletin	3.07 (2.98)
Average Num. of bulletins per dataset	1.20 (0.54)
Average Num. of extracted sentences per bulletin	4.74 (3.42)
Average Num. of generated clozes per sentence	3.41 (1.94)
Average Num. of generated clozes per bulletin	16.19 (12.74)
Average Num. of words per cloze	28.08 (7.53)
Average Row Size	352.98 (4145.87)
Average Column Size	25.61 (215.88)

Table 4.2: Dataset statistics



(a) spaCy labels

(b) Aggregate labels

Figure 4.4: Distribution of types of answers in our cloze-style questions

4.4.1 Bulletins and datasets:

On average, each bulletin is paired with $3.07 \pm (2.98)$ datasets. Likewise, each dataset is responsible for $1.20 \pm (0.54)$ bulletins. We observe that bulletins do not commonly share datasets between them. From each bulletin, we extract on average $4.74 \pm (3.42)$ main-point sentences, which is in line with the ONS guidelines discussed in Section 3.2.1. Using these extracted sentences, we can identify on average $3.41 \pm (1.94)$ named entities and in turn generate a total of $16.19 \pm (12.74)$ cloze-style statements for each bulletin.

4.4.2 Common answers and answer-types:

From Figure 5.1, we observe that four out of the five most common types of masked entities are numerical (or involve numbers), such as *DATE*, *PERCENT*, *CARDINAL* and *MONEY*. If we consider the aggregate labels, we can see that *TEMPORALMASK* and *NUMERICMASK* are closely balanced. However, observing Figure 4.3 we do not see any *NUMERICMASK* tags in the Top-10 most common answers. This hints that, even if *NUMERICMASK*s make up a considerable component of the probe, they are made up by distinct, non-repeating, values. Unlike *TEMPORALMASK*, where there is a clear repetition in answers, with common answers as *2019*, *2018*, ...

We would like to point out the recency bias observable in the cloze statements. The ONS website updates on a rolling basis and appears to archive older posts. It is then natural for our extracted statements to be biased towards recent events. It is also worth noting that most bulletins appear to only cover up to the year 2020 and do not frequently include comments for the year 2021. We expect this delay in bulletins to be persistent.

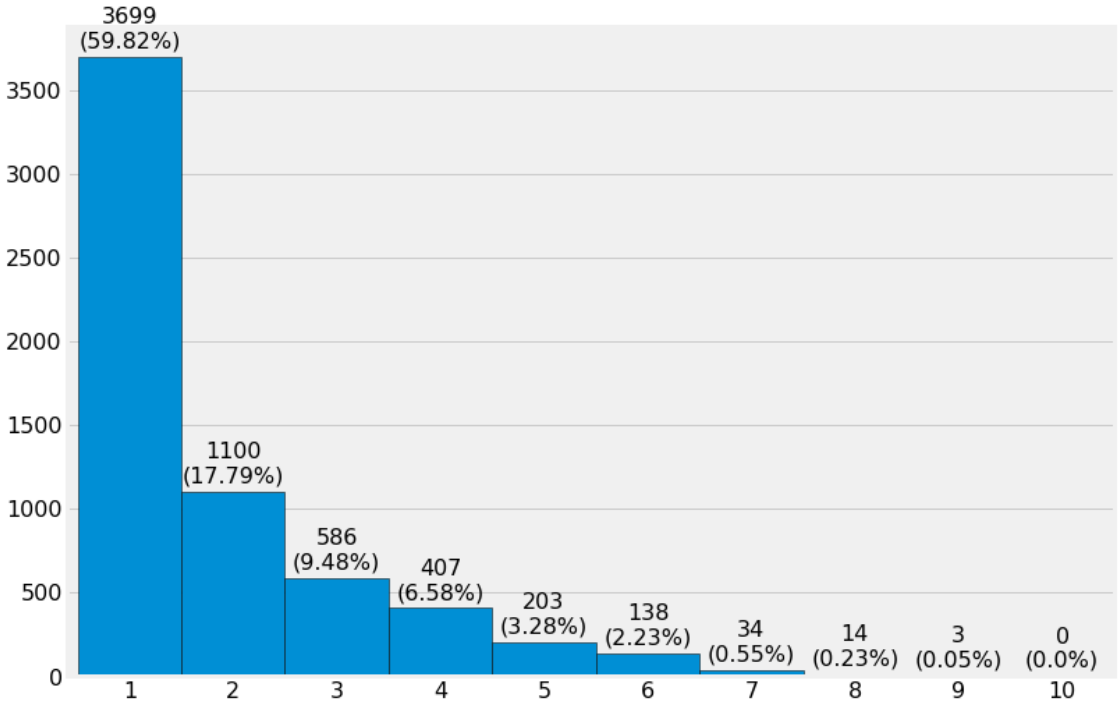


Figure 4.5: Distribution of number of words in answer text

4.4.3 Multi-token answers:

It is worthwhile investigating the named entities we mask since they will need to be predicted during test time. Figure 4.5 showcases how lengthy each named entity is. We define entity length as the total number of words that make up the answers. Although the majority (60%) of answers are only made up by a single word; there is still a considerable amount being made up by more than 2 words (22.4%). We urge the reader to refer to Section A.4 to examine sampled cloze statements together with the answer text for every answer length.

We observe masked entities such as "year ending on December 1995" as well as more noisy counterparts such as "year ending Quarter 2 (Apr to June) 2015". Such entities, although technically correct, appear to be out of the scope of answers LMs can generate. These require human level phrasing and grammar to match exactly. Expecting our answers to include "(Apr to June)" or even the more complete "(April to June)" is highly unrealistic and dependent on the writing style of the authors of the datasets. This means that exact match metrics will be under-representative of the true performance of our modelling.

We also observe cases where sentences such as "the end of the financial year ending March 2015" are being considered and masked as a single answer. We would expect our NER system to only consider "March 2015" as the NE of interest, but we can still see how the more complete sentence is still a valid categorisation. We consider these entries as noisy clozes in our probe. The user can choose to remove these by restricting the `answer_text` to be of length smaller than some N. We did not consider this for our work.

We discuss how we attempt to work around multi-token answers in Section 5.2.

Answer Type	Answer Text	Frequency
PLACEMASK	England	197
TEMPORALMASK	2019	169
PLACEMASK	Wales	157
TEMPORALMASK	2018	136
PLACEMASK	London	85
TEMPORALMASK	2017	78
TEMPORALMASK	2020	55
TEMPORALMASK	annual	54
TEMPORALMASK	2016	51
TEMPORALMASK	2014	41

Table 4.3: Top 10 most common cloze answers

4.4.4 Required operations:

We also present, in Figure 4.6, a collection of cloze statements together with their data. These were randomly sampled based on their answer length. We choose to present three examples, with one-, three- and five-word length answers. Locating the golden data source required our own input by inspecting all respective related datasets and identifying the one that could answer the cloze.

Figure 4.6a and Figure 4.6c both present a cloze statement relating to the use of controlled substances in the United Kingdom. The appropriate dataset that includes the required estimates is also presented.[11] We point out that the original bulletin is paired with only a single dataset. However, identifying the appropriate worksheet within is non-trivial. The dataset is made up of 41 worksheets with names such as 1.01, 1.02, . . . , 5.02. Identifying the golden worksheet, required reading a lookup table, available in Figure A.1. Even for us, it is not immediately clear which is the relevant sheet as some sheets cover very similar content.

Figure 4.6a expects a single-word answer, *"42%"*. We point out how the cloze refers to "*the previous year*". It is not immediately clear what this signifies but we make the assumption that due to how bulletins are published, this refers to the most recent available data point (hence change between years 2019 and 2020). This is correct but cannot be known unless the computation is carried out. The computation required can be broken down into: 1) identifying the relevant "Amphetamines" row; 2) identifying the relevant "year" columns (which is unclear); 3) computing the percentage change as shown.

Figure 4.6c on the other hand expects a five-word answer, *"the year ending December 1995"*. To answer this cloze, one must: 1) identify the relevant "Amphetamines" row; 2) compute sequentially (from recent to old) the percentage change; 3) determine the longest sequence where there is a monotone decrease; 4) return the last year where this is true. However, even if these steps are correctly performed, we would not reach the expected conclusion as the dataset stops at "April 2001", whereas the answer expects at least entries covering "December 1995". It is not directly clear to us how one can confirm this statement, as we cannot identify any worksheet that covers such a large span of time.

Lastly, Figure 4.6b showcases a considerably easier case. Here we present a cloze statement relating to SARS-CoV-2 antibodies in school staff. To answer this we need to: 1) identify the "secondary school type" row; 2) identify the two columns relating to the "95% confidence interval";

[11] Some columns have been hidden to aid readability

Model	Precision	Recall	F1
`en-core-web-trf`	0.86	0.83	0.84
`en-core-web-lg`	0.80	0.81	0.80
`en-core-web-sm`	0.74	0.72	0.73

Table 4.4: spaCy NER performance

3) aggregate and return. The dataset is only made up by 6 columns and they are accompanied with appropriate headers. Identifying the relevant ones can be approached by exact matching.[12] It is not directly clear however how the concatenation should work. We know, given the context of a confidence interval, that we report it at (lower, upper) — but what if we did not have this prior knowledge?

4.5 Evaluation

As our probe is made up by a large collection of unlabelled, unsupervised datasets — we could not measure the global performance of our processing pipeline nor the performance of the cloze question generation. Thus, we constructed two smaller testing sets that were used to evaluate our work and understand the effects of changes in our approach.

To evaluate performance of the NER recognition step and cloze generation, we randomly sampled two bulletins from each of the four main categories. This yielded eight bulletins, with a total of 38 main points.

To evaluate performance of our dataset filtering and structured data processing we sampled 25 datasets by choosing them based on their number of columns, D.

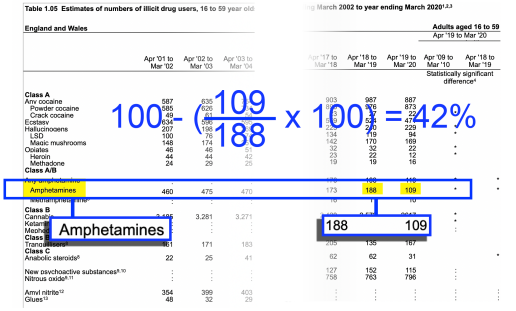
- 5 from group: $D \leq 5$
- 5 from group: $5 < D \leq 50$
- 5 from group: $50 < D \leq 100$
- 5 from group: $100 < D \leq 2500$
- 5 from group: $D > 250$

4.5.1 NER performance

We investigated the retrieved entities using three available language models within spaCy. These were the *English pipeline optimized for CPU* small and large variants (`en-core-web-sm`, `en-core-web-lg` respectively) and *English transformer pipeline* (`roberta-base`; `en-core-web-trf`). Consistently, `roberta-base` outperformed the other variants with the small variant being the worst. We saw that `en-core-web-trf` had less issues identifying multi-token entities. As we only had to run this once, we opted for the highest performing NER model, `en-core-web-trf`, even if computation time was significantly longer. Precision, recall and F-1 scores are available in Table 4.4.

[12] Our bi- and tri-gram exact matching approach will pair "95% confidence interval" correctly.

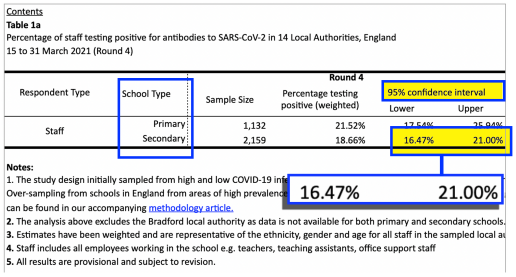
Cloze:
Amphetamine use in the last year in adults aged 16 to 59 years fell by **NUMERICMASK** compared with the previous year (to 109,000 people), continuing the long-term decline since the year ending December 1995.



Answer: 42%

(a) Single-word answer

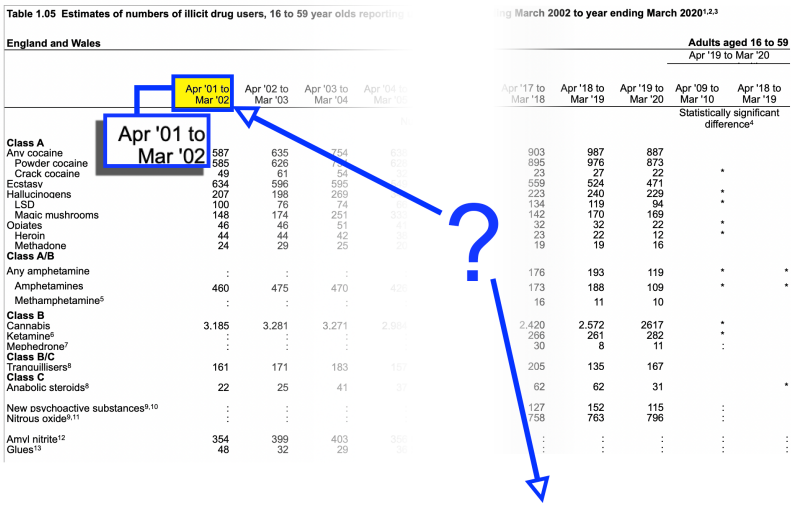
Cloze:
In March 2021 (Round 4), 21.52\% of primary school staff (95\% confidence intervals: 17.54\% to 25.94\%) and 18.66\% of secondary school staff (95\% confidence intervals: **NUMERICMASK**) tested positive to SARS-CoV-2 antibodies.



Answer: 16.47% to 21.00%

(b) Three-word answer

Cloze:
Amphetamine use in the last year in adults aged 16 to 59 years fell by 42% compared with the previous year (to 109,000 people), continuing the long-term decline since **TEMPORALMASK**.



Answer: the year ending December 1995

(c) Five-word answer

Figure 4.6: Different length answers together with the required operations to deduce answer.
(TOP LEFT) Single-word: 1) requires identification of correct "Amphetamines" row, 2) "previous year" is ambiguous (no specific year), 3) mismatch between units in table and cloze (nums. in thousands), 4) requires the conversion of counts to percentages.
(TOP RIGHT) Three-word: 1) requires identification of correct "Secondary" row, 2) answer spans two cells, 3) concatenate cell values
(BOTTOM) Five-word cannot be answered as dataset does not go back in time enough, if it did: 1) requires identification of correct "Amphetamines" row, 2) find longest time where numbers monotonically decrease.
Some columns have been hidden for readability

4.5.2 Structured data processing hyper-parameters

- To determine an appropriate cutting point for what is considered an "empty" or unusable dataset, we sampled and investigated 5 spreadsheets each from groups that contained two, three, four and five columns.
  - We consistently saw that sheets with were made up by less than three columns were unusable. They contained mostly text notes or references to publications. Their main purpose was not for analysis rather accompany spreadsheets and provide extra information.
  - We did not observe any cases where bigger tables shared the above description.
  - We thus determine that the cut-off point of excluding sheets with less than 3 columns a valid threshold.
- To determine an approach to clean cell entries, we investigated the spreadsheets of our test set.
  - We saw that ONS authors use the "-" character as a place-holder for an empty value.
  - ONS used "*" to indicate statistical significance.
  - ";", ":", "%" were also seen used in various settings.
  - We also observed entries were the value "NaN", "N/A" was used.
  - On rare occasions, there were some non-printable ASCII characters.
  - We replace the above with an empty string.
- To determine an appropriate threshold or factor to scale the number of missing values and remove unrelated rows, we investigated the spreadsheets of our test set.
  - We tried utilising the median, mean and upper quartile as well as three scaling factors, 1.2, 1.5, 1.75.
  - Consistently the upper quartile was over-aggressive with the final filtering — removing usable contents from smaller sub-tables.
  - Behaviour between the mean and median as a baseline to compare with was similar but saw that consistently we missed unrelated rows with hyperlinks and buttons — commonly found in the start of the spreadsheets.
  - Large numbers usually resulted in bad rows being ignored and not removed, whereas smaller numbers were very aggressive in their filtering.
  - Out of the 25 investigated datasets, 2 showed consistent issues and were not detected appropriately with either approach.
  - 1.5 and 1.75 both set reasonable thresholds but we prefer to err on the side of caution, in an attempt to minimise information loss.
- To handle "NaN" values in the column headers, we rely on the value of the corresponding bottom cell. We understand how sometimes this would mean we are imputing a data point

instead of a column header but we argue that it is very important to have populated column names due to how relevant they are in the table embeddings (W. Chen et al. 2020; Herzig, Nowak, et al. 2020).

  - We tried to understand what the spreadsheets look like for cases where this approach breaks by trialling both methods on our test set.
  - We saw that frequently, actual column headers and data content is separated by an empty row. If we are going to impute this cell values then we look at the original table to see if there was an empty row to separate them. We can then be more certain that this is indeed the proper column name.
  - However, there are cases where this does not work.
  - Out of the 25 datasets, we wrongly impute a column name for 12 headers out of a total of 147 columns.
  - We choose to continue this imputation as we consider empty column names to be a big issue in the performance of the language models.

- To determine which columns are properties of sub-tables we investigated various thresholds
  - We considered using the median value of empty cells in each column as we had good performance when identifying noisy rows.
  - We considered 75%, 85% and 95% as cutting points. If a column contained more empty cells than these cutting points then we choose to impute it.
  - We saw that requiring 75% and 95% empty cells frequently underperfomed for small and large tables respectively.
  - Tables with more complicated structure and sub-tables suffered the most. Columns were not identified and not imputed correctly.
  - Tables however with more simple structure were behaving as expected.
  - For these simpler datasets, 85% was the only threshold that did not misidentify any columns.

Chapter 5

Experimental Results

In this final chapter, we evaluate the performance of current state-of-the-art models on our probe. We compare how the usage of tabular data as context aids the predictive ability of the language models in question. Discussion on how the models are implemented can be found in Section 5.1, how multi-token mask modelling is handled in Section 5.2. We provide baseline performance in Section 5.3 and investigate the predictions in Section 5.4.

5.1 Model implementation

We consider 5 architectures as benchmarks. These are the BERT (Devlin et al. 2019), RoBERTa (Liu et al. 2019), ALBERT (Lan et al. 2020), T5 (Raffel et al. 2020) and TAPAS (Herzig, Nowak, et al. 2020) models. We utilise the Transformers[1] library to deploy our models. We operate under a zero-shot learning scenario and do not perform any pre-training on our side. This was largely motivated due to the limited computational resources available to us.

We point out that the TaPas implementation available on Transformers was not ready for the mask modelling task. We want to thank Niels Rogge[2], for their continuous communication in getting TaPas ready for MLM. The model did not have the required weights readily available and required conversion from Google's implementation[3]. Performing this conversion was not direct as there were compatibility issues with the `pipeline` functionality of Transformers as well as the `_import_structure` expected in the weight loading. After flagging these issues up we could then work on getting `TAPASForMaskedLM` to operate as expected. We point out how this was a considerable bottleneck[4] and how Niel's advice was invaluable to getting the model to work. After these changes TaPas is now available for the MLM task in Transformers.

We also want to point out that TaPas has significant memory requirements, even during inference time. For this reason, we could only investigate the `TAPAS-small` variant. Deploying larger variants consistently crashed our Colab Pro session. We note that the reported results[5] for

[1] `https://huggingface.co/transformers/`

[2] `https://github.com/NielsRogge`

[3] `https://github.com/google-research/tapas`

[4] It was unclear whether the MLM could work with the TaPas model, so we investigated if it would be possible to convert out cloze statements into question-answering.

[5] `https://github.com/google-research/tapas`

`TAPAS-small` are considerably lower (0.37) when compared to `TAPAS-large` (0.51) and expect this to be reflected in our results.

5.2 Multi-token mask modelling

Generating multi-token answers using a bi-directional architecture is a difficult task, (Lewis et al. 2019; Jiang et al. 2020). These models predict individual words given left and right context, unlike uni-directional models where one can autoregressively decode the next token conditioned on the previous ones. Decoding multi-token masks remains an open problem.

(Schick et al. 2021) propose an approach to multi-token predictions but their solution expects that we know the length of the final prediction a-priori. Their approach works in the classification setting they operate in but cannot be extended easily. We attempted to leverage the proposed solution by (Ghazvininejad et al. 2019; Jiang et al. 2020) but implementation and subsequent results appeared tenuous.

To enforce our models to consider answers that were made up with more than a single token we modified the cloze statements. The idea is to take our cloze statements that contained a single mask token and convert them into five identical ones that contained an increasing amount of mask tokens. The aim here is to end up with five independent predictions of answers of up to five tokens. We would then compute the pseudo log-likelihood[6], (Jiang et al. 2020), and pick the one that maximises it.

The pseudo log-likelihood is the sum of log probabilities of each predicted token conditioned on the other tokens, (Salazar et al. 2020): $v(j-i+1) = \sum_{k=i}^{j} \log c_k$ where c_k is the confidence of the k^{th} predicted token, and $v(m)$ is the overall prediction confidence with m initial mask tokens. Given that the bi-directional models condition on both left and right context, we need to sequentially predict each token within the sentence.

For clarity, we show a simplistic example, the case where we consider three-token answers.

John studies at [MASK]
→ John studies at [MASK] [MASK] [MASK]
→ John studies at [MASK] [MASK] London
→ John studies at [MASK] mathematics London
→ John studies at University mathematics London
→ John studies at University College London

At the first step, we would generate in parallel three predictions for each `[MASK]` position in the sentence. We then pick the one with the highest confidence and set it within the sentence. We then have our cloze sentence but now only two `[MASK]` are present. Similarly, we generate two new predictions and again set the one with the highest confidence. We repeat until all `[MASK]` are filled in the sentence. We now end with a complete sentence with no more `[MASK]` tokens. We now pick the one with the lowest confidence and re-generate a new prediction. We repeat this process until the sequence is converged[7] and no token is being changed. This is our final prediction for this given length.

[6] Thank you Patrick Lewis for explaining this!

[7] This is similar to the Expectation-Maximisation algorithm, where on each step we increase the likelihood of observing this sequence.

Finally, after generating five different predictions of increasing lengths, we need to pick the best one. To do this, we use again the pseudo log-likelihood and length normalise[8] the quantities to enable cross comparison, (Ghazvininejad et al. 2019).

However, empirical results were sub-par. While the sampling process has a theoretical justification, it also requires N forward passes. This drastically increases the computation time and makes generating predictions a very slow process. Additionally, we investigated that answers that maximised this log-likelihood were infrequently the appropriate ones. We unfortunately had to abandon this approach and resorted to single-token prediction.

5.3 Baselines

Model \ Probe	Cloze Only		Cloze and Context	
	EM@1	EM@5	EM@1	EM@5
BERT-base	8.35%	13.10%	4.63%	5.81%
BERT-large	9.95%	14.94%	5.04%	8.84%
RoBERTa-base	11.19%	17.33%	6.44%	9.91%
RoBERTa-large	12.39%	18.32%	6.79%	11.25%
ALBERT-base	8.85%	13.01%	2.21%	3.18%
ALBERT-large	9.04%	14.48%	3.08%	5.16%
T5-base	10.36%	14.68%	7.21%	11.13%
T5-large	13.38%	20.05%	11.01%	16.84%
TaPas-small-masklm	—	—	1.44%	3.92%

Table 5.1: Model accuracy (exact match at 1; exact match at 5) on generated clozes. Separate cases for when context was and was not used. Five models were assessed: 1) BERT (base, large); 2) RoBERTa (base, large); 3) ALBERT (base, large); 4) T5 (base, large); 5) TaPas (small). TaPas-small was the only variant we tried due to memory requirements. T5-large performed with the highest accuracy for both use cases (context, no context) followed by RoBERTa-large. TaPas cannot be ran without use of context. Introduction of context drastically reduces prediction quality.

Table 5.1 provides the baseline accuracy of the models we assessed. We run the experiments with and without the use of spreadsheets as context. We disclaim that the models show consistently worse performance when context is supplied. Context should theoretically aid their predictive ability but it appears to do the opposite. We discuss this in Section 5.5.

We observe the highest performance out of the `T5-large` model, followed by `RoBERTa-large` and `RoBERTa-base`. `T5-large` also shows the highest robustness to the introduction of context, with the lowest drop in performance. Surprisingly `TaPas-small-masklm`, the only model with a table-specific head that manages table entailment, performed the worst under the use of context. It was not possible to run `TaPas-small-masklm` without the use of context. Passing empty strings as context was not possible due to the Transformers implementation and their error checking processes. It was also not possible to deploy larger variants of `TaPas` due to its memory requirements and the limited computational resources we operate with.

[8] We divided our computed pseudo log-likelihood with the number of words in the sentence; this tries to reduce the bias towards short answers.

5.4 Investigating predictions

We provide more in-depth insights into the predictions of the `T5` and `RoBERTa` models. We first investigate the `T5` predictions followed by the corresponding `RoBERTa` analysis.

Figure 5.1a showcases the top 10 most common correct and incorrect predictions under the `T5` models. For both `T5-large` and `T5-base`, we have identical correct and incorrect predictions with only slight change in order between the less-frequent terms. We point out that terms such as "*england*" appear to be often predicted correct but at the same time frequently mispredicted. We group the cases where the correct answer is "*england*" and investigate the incorrect predictions.

It appears that both `T5` variants frequently incorrectly assign the label "*the uk*", "*england, wales and northern ireland*" or "*england and wales*". If we subset all entries where the true label is "*england*" then (9.67%) of `T5-base`'s and (35.22%) of `T5-large`'s incorrect predictions contain the word "*england*". This shows that the predictions are related to the true label but not strictly correct. That said, the true label "*england*" is commonly mispredicted as "*the united states*", "*scotland*" or "*wales*". We can deduce that the model understands that the true label is a country but has difficulty distinguishing which one it is within the United Kingdom.

Analysing the `RoBERTa` predictions now shows that again "*england*" is the most common incorrect prediction, which is understandable given how frequently it occurs, Table 4.3. However, we now observe that the incorrect predicted labels are single token answers such as "*scotland*", "*britain*" and "*ireland*". This, together with the Table 5.2 shows that `RoBERTa` does not manage to handle or produce multi-word answers, in contrast to `T5`. Interestingly, `RoBERTa` manages to outperform `T5` on single token words. The measured performance we observe then is reduced by its inability to predict any longer answers. If we could combine `RoBERTa`'s performance on single token and `T5`'s ability to generate higher-order answers then the combined predictions should be considerably better. The *seq2seq* model then shows superiority when tasked with answers that require more than a single word.

To investigate the effect of multi-token answers, we assess how many correct predictions we have under different true label-lengths. It is clear that, the longer the correct answer is, the harder it is for our model to predict it. However, interestingly, `RoBERTa` does not manage to get any answer that is longer than a single word correct. Surprisingly, `T5`, manages to even get some six-word answers. This is a single occurrence of the answer "*the financial year ending March 2015*". If we check five-word answers, then this pattern continues and we see `T5` being able to predict answers such as "*the year ending March YEAR*", where YEAR covers 2017, 2018 and 2019.

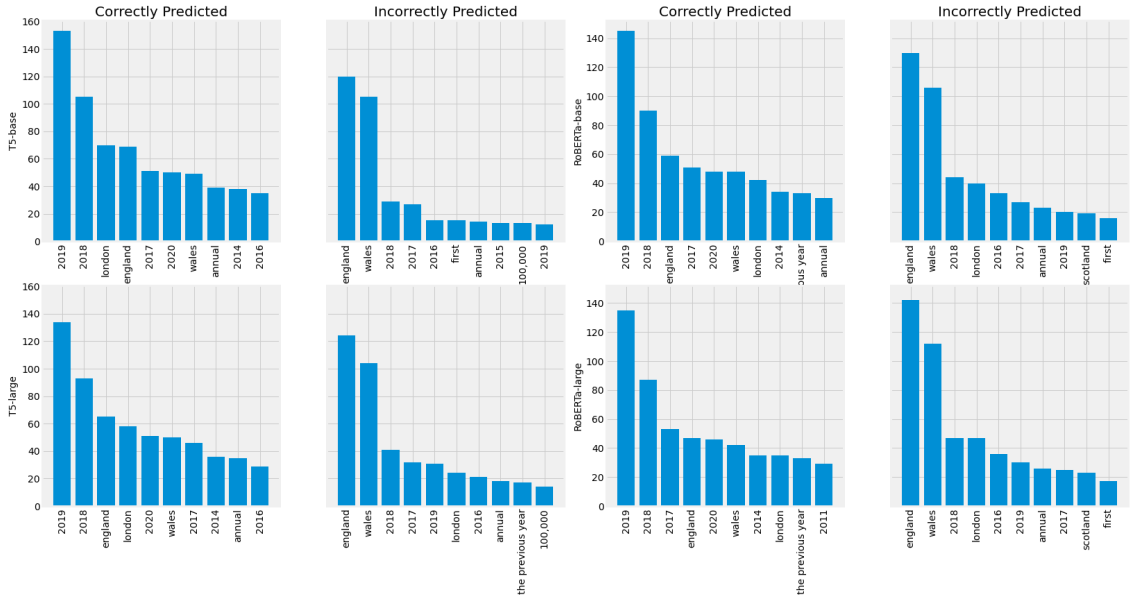
5.5 Why context reduces performance

If we investigate the context we start to see considerable issues. Even if our pre-processing manages to get the spreadsheets into a parsable format, fixing the multiple column header issues and removing most of the unrelated information — we do not manage to flag and retrieve the relevant rows. We have two distinct cases: 1) no relevant context can be found; 2) context is found but its too long. We discuss the first case first.

The way we aim to identify relevant content within a dataset, relies heavily on the expectation that column headers and table contents will have appropriate strings that can be compared with

Answer length	T5-base	T5-large	RoBERTa-base	RoBERTa-large
1	14.98	18.72%	18.80%	20.82%
2	4.16	7.57%	0.00%	0.00%
3	5.34	8.78%	0.00%	0.00%
4	1.97	2.96%	0.00%	0.00%
5	1.99	4.48%	0.00%	0.00%
6	0.00	0.74%	0.00%	0.00%
7	0.00	0.00%	0.00%	0.00%
8	0.00	0.00%	0.00%	0.00%
9	0.00	0.00%	0.00%	0.00%

Table 5.2: Investigating accuracy of T5 and RoBERTa models based on the length of the true label.



(a) T5 predictions

(b) RoBERTa predictions

Figure 5.1: Top-10 T5 and RoBERTa predictions without the use of context. Split into correctly and incorrectly predicted together with the model variants. Minor processing is performed to convert both predictions and true labels to all lower-case.

our cloze statement. This is infrequently the case. As explained in Section 4.3.2, the row identification relies on first finding the appropriate columns. If we instead checked all table contents, then this process would be prohibitively slow. Given that identification of relevant content is then a sequential process, any mishap in a previous step will negatively affect the final outcome.

In addition to the heavy processing required to get the column headers to a steady state, we are not confident that the columns will be relevantly named. As expressed in Section 4.3.4, entries are often abbreviated. We did not implement a mechanism that converts abbreviations to their extended form and thus cannot link the two. Likewise, we are unsure if the cloze statement contains the entities in their abbreviated or expanded form either.

If however our pipeline identifies a relevant section within the spreadsheet, it is often the case that it is too long. As we operate with a finite token capacity, $(512 - \text{cloze length})$, we should be providing at most 15 cells or $3 - 4$ columns and $4 - 5$ rows. In the cases where some relevant content is found, it is common that a column is identified but no specific row is selected. We then return all rows. This causes considerable issues as the average size of the worksheets is over 50 rows. We considered only using worksheets that were small (< 10 rows) but this was unproductive. Most worksheets that contain the necessary statistical data were considerably larger and we would reduce our collection drastically. There is also a third issue, where the spreadsheet cannot be used to answer the specific statement. We see this in Figure 4.6c.

5.5.1 How does poor context affect the predictions

The introduction of context reduces the accuracy of our predictions, Table 5.1. We attribute this to the quality of the context. For the case of `TaPas` it is especially clear that poor context can have a very negative impact on its prediction ability. Given that `TaPas` is closely linked to the `BERT` model but has further table reasoning abilities, we expect it to behave similarly if not better.

However, its performance shows that there is a great reliance on the validity and relevancy of the supplied context. It is then evident that, providing it with irrelevant or low-quality context, it can have difficulty generating appropriate predictions. We refer back to Section 2.6 and specifically on the key assumption of EQA. We presuppose that the answer to the cloze sentence is made up by spans of token within the given document, in our case, context. This is often not the case hence this assumption fails.

Investigating its predictions, we can see that the generated answers appear to be values of random cells from the context. Even for cases where a relevant context is identified, the predictions are incorrect. The rare cases where `TaPas` provides a correct answer (88/6154) are exclusively years and specifically 2019 or 2018 entries which do not require multi-hop reasoning to answer. We believe there was minimal utilisation of the context in these cases. Rather, we expect these to have occurred from a recency bias of the model.

We wanted to measure how false context affects `TaPas` and `BERT`. We designed a mock example where we provide a cloze statement: "[MASK] is the capital of England". Instead of providing a correct table with the corresponding capitals we switch the capital of France for that of England. It appears that neither model can overwrite their learned knowledge during training (establishing that "London" is the capital of England) and use the context exclusively. It appears then knowledge follows some hierarchy, with models prioritising the knowledge learned during training. However,

at least `TaPas` provides the correct cell as the second most confident answer, followed by incoherent predictions. If we introduce another column, say "Language" the predictions do not change. This is regardless to the new column's position and in turn within the context which is promising some robustness.

Country	Capital
England	Paris
France	London

Input: [MASK] is the capital of England [SEP] Country is England; Capital is Paris [SEP] Country is France; Capital is London

TaPas (top-5): *"london"*, *"paris"*, *","*, *"which"*, *")"*

BERT (top-5): *"country"*, *"london"*, *"city"*, *"capital"*, *"england"*

Table 5.3: How false context can affect predictions

Chapter 6

Conclusions and further work

6.1 Summary

Our main aim was to construct a new benchmark that can be used in the task of fact verification of claims that require structured data to answer. We extracted a large collection of published bulletins from the Office of National statistics and collect the statistical data in tabular format required to answer them. We investigated their structure and determined a corresponding way to generate cloze-style statements. We produced a processing unit that parses the semi-structured data and alleviates many of the stylistic and formatting issues of the spreadsheets. We point out the API limitations that are missing for the optimal execution of the task. We test a selection of models and assess their predictions on the masked language model assignment. The inconsistent and unmethodical structuring of the datasets requires an advanced mechanism that detects relevant content within the tables even if table properties such as column headers are missing at large. This was the main culprit for the low performance of our models. Finally, cloze-style questions are considerably less likely to cause the models to learn underlying data generation processes. Without the use of any true or false labels, learning common patterns or answers is a harder task. We consider our probe as a valuable benchmark as it requires a suitable information retrieval mechanism as well as an architecture that can multi-hop reason over open-domain tables. Success in the probe then expects a solution that can do both of these tasks at a high level.

6.1.1 Further work

Establishing a method that can detect relevant and key content in a table by inspecting the appropriate cloze statement is evidently a difficult task if the tables in question lack substantial structure. The lack of column headers can have a very large impact on the retrieval step. For this reason it is key to design a method that can work around this issue at scale. We see the task of relevant content retrieval as the principal obstruction in this work. We propose two potential solutions.

First, we observed that the spreadsheets usually contain worksheets within, that discuss abbreviations used, notes around modifications and other relevant information. In our work we ignored these in an attempt to keep a narrow scope but we do believe there is valuable information to be

extracted. Part of the column header issues we stumbled upon could have been potentially solved by building a robust mechanism that can detect the use of abbreviations and expand them to the form found in the target cloze. Likewise, a parser that can read and comprehend the content description worksheet, as in Figure A.1, usually found as the first sheet, can be a better approach to detecting the most relevant sheet for our cloze. Currently, we only attempt to identify relevant worksheets of a spreadsheet by approximate string matching. This evidently is not sufficient. Content worksheets contain short descriptions about the contents of each sheet and can be used to point us to the right direction.

Secondly, it is important to communicate these equivocal pre-processing steps (like column header imputation, cell imputation) to our modelling. Currently the models assume that the context we provide is correct. We believe it is important for the embeddings to have a mechanism that can flag potentially incorrect values in an attempt to allow the model to focus on the true data. Introducing this dimension in the embeddings does require case-specific tuning but will allow for a more reliable and robust estimation.

Bibliography

Taylor, W. L. (1953). ""Cloze Procedure": A New Tool for Measuring Readability". In: *Journalism & Mass Communication Quarterly* 30, pp. 415–433.

Cohen, Sarah et al. (Apr. 2011). "Computational Journalism: A Call to Arms to Database Researchers". In: pp. 148–151.

Flew, Terry et al. (2012). "THE PROMISE OF COMPUTATIONAL JOURNALISM". In: *Journalism Practice* 6.2, pp. 157–171. DOI: 10.1080/17512786.2011.616655. eprint: https://doi.org/10.1080/17512786.2011.616655. URL: https://doi.org/10.1080/17512786.2011.616655.

Mikolov, Tomas et al. (2013). *Efficient Estimation of Word Representations in Vector Space*. arXiv: 1301.3781 [cs.CL].

Surdeanu, M. et al. (2014). "Overview of the English Slot Filling Track at the TAC 2014 Knowledge Base Population Evaluation". In:

Vlachos, Andreas et al. (June 2014). "Fact Checking: Task definition and dataset construction". In: *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*. Baltimore, MD, USA: Association for Computational Linguistics, pp. 18–22. DOI: 10.3115/v1/W14-2508. URL: https://aclanthology.org/W14-2508.

Kiros, Ryan et al. (2015). "Skip-Thought Vectors". In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes et al. Vol. 28. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2015/file/f442d33fa06832082290ad8544a8da27-Paper.pdf.

Li, Yaliang et al. (2015). *A Survey on Truth Discovery*. arXiv: 1505.02463 [cs.DB].

Bahdanau, Dzmitry et al. (2016). *Neural Machine Translation by Jointly Learning to Align and Translate*. arXiv: 1409.0473 [cs.CL].

Rajpurkar, Pranav, Jian Zhang, et al. (2016). *SQuAD: 100,000+ Questions for Machine Comprehension of Text*. arXiv: 1606.05250 [cs.CL].

Chen, Danqi et al. (2017). *Reading Wikipedia to Answer Open-Domain Questions*. arXiv: 1704.00051 [cs.CL].

Elsahar, Hady et al. (May 2018). "T-REx: A Large Scale Alignment of Natural Language with Knowledge Base Triples". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA). URL: https://aclanthology.org/L18-1544.

Lazer, David M. J. et al. (2018). "The science of fake news". In: *Science* 359.6380, pp. 1094–1096. ISSN: 0036-8075. DOI: 10.1126/science.aao2998. eprint: https://science.sciencemag.

org/content/359/6380/1094.full.pdf. URL: https://science.sciencemag.org/content/359/6380/1094.

Radford, Alec et al. (2018). "Improving Language Understanding by Generative Pre-Training". In:

Rajpurkar, Pranav, Robin Jia, et al. (2018). *Know What You Don't Know: Unanswerable Questions for SQuAD*. arXiv: 1806.03822 [cs.CL].

Speer, Robyn et al. (2018). *ConceptNet 5.5: An Open Multilingual Graph of General Knowledge*. arXiv: 1612.03975 [cs.CL].

Sun, Haitian et al. (Oct. 2018). "Open Domain Question Answering Using Early Fusion of Knowledge Bases and Text". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 4231–4242. DOI: 10.18653/v1/D18-1455. URL: https://aclanthology.org/D18-1455.

Thorne, James, Andreas Vlachos, Christos Christodoulopoulos, et al. (2018). *FEVER: a large-scale dataset for Fact Extraction and VERification*. arXiv: 1803.05355 [cs.CL].

Yoneda, Takuma et al. (Nov. 2018). "UCL Machine Reading Group: Four Factor Framework For Fact Finding (HexaF)". In: *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*. Brussels, Belgium: Association for Computational Linguistics, pp. 97–102. DOI: 10.18653/v1/W18-5515. URL: https://aclanthology.org/W18-5515.

Bauer, Lisa et al. (2019). *Commonsense for Generative Multi-Hop Question Answering Tasks*. arXiv: 1809.06309 [cs.CL].

De Cao, Nicola et al. (June 2019). "Question Answering by Reasoning Across Documents with Graph Convolutional Networks". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 2306–2317. DOI: 10.18653/v1/N19-1240. URL: https://aclanthology.org/N19-1240.

Devlin, Jacob et al. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv: 1810.04805 [cs.CL].

Ghazvininejad, Marjan et al. (2019). *Mask-Predict: Parallel Decoding of Conditional Masked Language Models*. arXiv: 1904.09324 [cs.CL].

Hanselowski, Andreas et al. (2019). *UKP-Athene: Multi-Sentence Textual Entailment for Claim Verification*. arXiv: 1809.01479 [cs.IR].

Lewis, Patrick et al. (July 2019). "Unsupervised Question Answering by Cloze Translation". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 4896–4910. DOI: 10.18653/v1/P19-1484. URL: https://aclanthology.org/P19-1484.

Liu, Yinhan et al. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. arXiv: 1907.11692 [cs.CL].

Malon, Christopher (2019). *Team Papelo: Transformer Networks at FEVER*. arXiv: 1901.02534 [cs.CL].

Niewinski, Piotr et al. (Nov. 2019). "GEM: Generative Enhanced Model for adversarial attacks". In: *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*. Hong

Kong, China: Association for Computational Linguistics, pp. 20–26. DOI: 10.18653/v1/D19-6604. URL: https://aclanthology.org/D19-6604.

Nishida, Kosuke et al. (2019). *Answering while Summarizing: Multi-task Learning for Multi-hop QA with Evidence Extraction*. arXiv: 1905.08511 [cs.CL].

Petroni, Fabio, Tim Rocktäschel, et al. (2019). *Language Models as Knowledge Bases?* arXiv: 1909.01066 [cs.CL].

Thorne, James, Andreas Vlachos, Oana Cocarascu, et al. (Nov. 2019). "The FEVER2.0 Shared Task". In: *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*. Hong Kong, China: Association for Computational Linguistics, pp. 1–6. DOI: 10.18653/v1/D19-6601. URL: https://aclanthology.org/D19-6601.

Yang, Wei et al. (June 2019). "End-to-End Open-Domain Question Answering with BERTserini". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 72–77. DOI: 10.18653/v1/N19-4013. URL: https://aclanthology.org/N19-4013.

Atanasova, Pepa et al. (July 2020). "Generating Fact Checking Explanations". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 7352–7364. DOI: 10.18653/v1/2020.acl-main.656. URL: https://aclanthology.org/2020.acl-main.656.

Chen, Wenhu et al. (2020). *TabFact: A Large-scale Dataset for Table-based Fact Verification*. arXiv: 1909.02164 [cs.CL].

Eisenschlos, Julian Martin et al. (2020). *Understanding tables with intermediate pre-training*. arXiv: 2010.00571 [cs.CL].

Guu, Kelvin et al. (2020). *REALM: Retrieval-Augmented Language Model Pre-Training*. arXiv: 2002.08909 [cs.CL].

Herzig, Jonathan, Paweł Krzysztof Nowak, et al. (2020). *TAPAS: Weakly Supervised Table Parsing via Pre-training*. arXiv: 2004.02349 [cs.IR].

Hidey, Christopher et al. (July 2020). "DeSePtion: Dual Sequence Prediction and Adversarial Examples for Improved Fact-Checking". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 8593–8606. DOI: 10.18653/v1/2020.acl-main.761. URL: https://aclanthology.org/2020.acl-main.761.

Jiang, Zhengbao et al. (2020). *X-FACTR: Multilingual Factual Knowledge Retrieval from Pretrained Language Models*. arXiv: 2010.06189 [cs.CL].

Khouja, Jude (July 2020). "Stance Prediction and Claim Verification: An Arabic Perspective". In: *Proceedings of the Third Workshop on Fact Extraction and VERification (FEVER)*. Online: Association for Computational Linguistics, pp. 8–17. DOI: 10.18653/v1/2020.fever-1.2. URL: https://aclanthology.org/2020.fever-1.2.

Lan, Zhenzhong et al. (2020). *ALBERT: A Lite BERT for Self-supervised Learning of Language Representations*. arXiv: 1909.11942 [cs.CL].

Petroni, Fabio, Patrick Lewis, et al. (2020). *How Context Affects Language Models' Factual Predictions*. arXiv: 2005.04611 [cs.CL].

Poerner, Nina et al. (Nov. 2020). "E-BERT: Efficient-Yet-Effective Entity Embeddings for BERT". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, pp. 803–818. DOI: 10.18653/v1/2020.findings-emnlp.71. URL: https://aclanthology.org/2020.findings-emnlp.71.

Raffel, Colin et al. (2020). *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. arXiv: 1910.10683 [cs.LG].

Salazar, Julian et al. (July 2020). "Masked Language Model Scoring". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 2699–2712. DOI: 10.18653/v1/2020.acl-main.240. URL: https://aclanthology.org/2020.acl-main.240.

Schlichtkrull, Michael et al. (2020). *Joint Verification and Reranking for Open Fact Checking Over Tables*. arXiv: 2012.15115 [cs.CL].

Yin, Pengcheng et al. (July 2020). "TaBERT: Pretraining for Joint Understanding of Textual and Tabular Data". In: *Annual Conference of the Association for Computational Linguistics (ACL)*.

Zhang, Hongzhi et al. (Nov. 2020). "Table Fact Verification with Structure-Aware Transformer". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 1624–1629. DOI: 10.18653/v1/2020.emnlp-main.126. URL: https://aclanthology.org/2020.emnlp-main.126.

Anonymous (2021). "Prompt Tuning or Fine-Tuning - Investigating Relational Knowledge in Pre-Trained Language Models". In: *Submitted to 3rd Conference on Automated Knowledge Base Construction*. under review. URL: https://openreview.net/forum?id=o7sMlpr9yBW.

Hardalov, Momchil et al. (2021). *A Survey on Stance Detection for Mis- and Disinformation Identification*. arXiv: 2103.00242 [cs.CL].

Herzig, Jonathan, Thomas Müller, et al. (2021). *Open Domain Question Answering over Tables via Dense Retrieval*. arXiv: 2103.12011 [cs.CL].

Holtzman, Ari et al. (2021). *Surface Form Competition: Why the Highest Probability Answer Isn't Always Right*. arXiv: 2104.08315 [cs.CL].

Ostrowski, Wojciech et al. (2021). *Multi-Hop Fact Checking of Political Claims*. arXiv: 2009.06401 [cs.CL].

Schick, Timo et al. (2021). *It's Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners*. arXiv: 2009.07118 [cs.CL].

Appendix A

Appendix

A.1 Generating counterfactual data for table entailment

Although not our primary task, we showcase how to generate data for the classification task of whether a statement is factual or not given some structured data. To train a language model around this, we require negative/ counterfactual data. These are statements that are factually incorrect but resemble the truth. Given that we provide a dataset with cloze statements together with the appropriate answer and answer type — one can sample a replacement to generate negative data.

Sampling from the same answer type can generate harder to answer data. For example, if the ground statement is structured around a numerical answer type then sampling a location or country name would make it much easier to detect. Likewise, statements that share answer types would require table reasoning to answer. We believe that this dataset can be naturally extended to the table entailment task, similar to (Lewis et al. 2019; W. Chen et al. 2020).

A.2 Available datasets within the ONS API

Complete list of datasets accessible through the ONS API as of August 2021.

```
'ageing-population-estimates', 'ageing-population-projections',
'older-people-economic-activity', 'older-people-net-internal-migration',
'older-people-sex-ratios', 'projections-older-people-in-single-households',
'projections-older-people-sex-ratios', 'weekly-deaths-age-sex',
'weekly-deaths-region', 'online-job-advert-estimates',
'faster-indicators-shipping-data', 'cpih01', 'mid-year-pop-est',
'weekly-deaths-health-board', 'weekly-deaths-local-authority',
'regional-gdp-by-year', 'wellbeing-local-authority',
'index-private-housing-rental-prices', 'suicides-in-the-uk',
'childrens-wellbeing', 'gdp-to-four-decimal-places',
'generational-income', 'health-accounts',
'wellbeing-quarterly', 'house-prices-local-authority',
```

```
'regional-gdp-by-quarter', 'labour-market',
'tax-benefits-statistics', 'ashe-tables-26',
'ashe-tables-25', 'ashe-tables-27-and-28',
'trade', 'ashe-tables-7-and-8',
'ashe-tables-3', 'ashe-tables-11-and-12',
'ashe-tables-20', 'ashe-tables-9-and-10',
'ashe-table-5', 'life-expectancy-by-local-authority',
'gdp-by-local-authority', 'gva-by-industry-by-local-authority'
```

A.3 Bulletin categories and subcategories

Complete list of the bulletin categories and subcategories available. There are 4 main categories and a total of 31 unique subcategories.

```
category: businessindustryandtrade
    'business', 'changestobusiness',
    'constructionindustry', 'itandinternetindustry',
    'internationaltrade', 'manufacturingandproductionindustry',
    'retailindustry', 'tourismindustry'

category: economy
    'economicoutputandproductivity', 'environmentalaccounts',
    'governmentpublicsectorandtaxes', 'grossdomesticproductgdp',
    'grossvalueaddedgva', 'inflationandpriceindices',
    'investmentspensionsandtrusts', 'nationalaccounts',
    'regionalaccounts'

category: employmentandlabourmarket
    'peopleinwork', 'peoplenotinwork'

category: peoplepopulationandcommunity
    'birthsdeathsandmarriages', 'crimeandjustice',
    'culturalidentity', 'educationandchildcare',
    'elections', 'healthandsocialcare',
    'householdcharacteristics', 'housing',
    'leisureandtourism', 'personalandhouseholdfinances',
    'populationandmigration', 'wellbeing'
```

A.4 Sampled clozes and answers with varying length

Here we present three sampled cloze statements and their corresponding answer texts. These are grouped based on the number of words the answers are made up from.

- Answer Length 1
  - Cloze: The ASMR due to IHDs in TEMPORALMASK was significantly lower in England than Wales (94.7 and 109.3 deaths per 100,000 people respectively). **Answer: 2019**
  - Cloze: The fall in production was driven by decreases of NUMERICMASK in mining and quarrying, 0.3% in manufacturing, and 0.9% in water supply and sewerage; these were offset partially by a rise in electricity and gas of 0.5%. **Answer: 15.0%**
  - Cloze: The police recorded 18,706 cruelty to children/young persons offences in PLACEMASK and Wales in the year ending March 2019; however, some of these offences will be child physical abuse rather than neglect. **Answer: England**

- Answer Length 2
  - Cloze: At birth, males in Wokingham could expect to live 15.5 years longer in "Good" health than males in Blackpool (TEMPORALMASK compared with 55.0 years). **Answer: 70.5 years**
  - Cloze: In 2017, the removal of air pollution by urban green and blue space in PLACEMASK equated to a saving of £162.6 million in associated health costs. **Answer: Great Britain**
  - Cloze: The average age of first-time mothers increased to 28.8 years in 2016, from TEMPORALMASK in 2015. **Answer: 28.6 years**
- Answer Length 3
  - Cloze: Of those adults who have a smartphone for private use, 17% did not have security on their smartphone and NUMERICMASK did not know whether they had security. **Answer: a further 32%**
  - Cloze: There was a statistically significant slowdown in the long-term improvement in age-standardised mortality rates for both England and Wales (TEMPORALMASK) around the early 2010s, in line with previous analysis. **Answer: 1990 to 2018**
  - Cloze: In March 2021 (Round 4), 21.52% of primary school staff (95% confidence intervals: 17.54% to 25.94%) and 18.66% of secondary school staff (95% confidence intervals: NUMERICMASK) tested positive to SARS-CoV-2 antibodies. **Answer: 16.47% to 21.00%**
- Answer Length 4
  - Cloze: TEMPORALMASK, the growth in employment in high street pubs and bars was three percentage points higher than in pubs and bars in non-high street locations. **Answer: Between 2015 and 2018**
  - Cloze: For 16 of the 24 cancers, 1-year survival for patients diagnosed TEMPORALMASK was slightly higher than for the overlapping period 2009 to 2013 in at least 1 of the sexes. **Answer: between 2010 and 2014**
  - Cloze: 26% of adults reported that they would have insufficient means to last TEMPORALMASK, and 10% longer than a week if they lost their main source of income. **Answer: longer than a month**
- Answer Length 5
  - Cloze: The year ending March 2020 Crime Survey for England and Wales (CSEW) estimated that 1.6 million adults aged 16 to 74 years had experienced sexual assault by rape or penetration (including attempts) since TEMPORALMASK. **Answer: the age of 16 years**
  - Cloze: Amphetamine use in the last year in adults aged 16 to 59 years fell by 42% compared with the previous year (to 109,000 people), continuing the long-term decline since TEMPORALMASK. **Answer: the year ending December 1995**

  - Cloze: 61% of adults said that they had not, in the previous year, run out of money before TEMPORALMASK or month, up from 52% in July 2010 to June 2012. **Answer: the end of the week**

- Answer Length 6
  - Cloze: There has been no significant change in the prevalence of sexual assault measured by the CSEW TEMPORALMASK (2.6%) and the year ending March 2017 (2.0%) surveys. **Answer: between the year ending March 2005**
  - Cloze: More than half of respondents (57%) had downloaded the NHS Test and Trace app, which was a significant increase from 45% TEMPORALMASK. **Answer: between 19 and 24 April 2021**
  - Cloze: In TEMPORALMASK, disabled women were almost twice as likely to have experienced any sexual assault in the last year (5.7%) than non-disabled women (3.0%). **Answer: the three years ending March 2018**

- Answer Length 7
  - Cloze: The latest figures from IDENTITYMASK show little change in the prevalence of domestic abuse in recent years. **Answer: the Crime Survey for England and Wales**
  - Cloze: This proportion has been gradually increasing since mid-February (44% in TEMPORALMASK). **Answer: the period 10 to 14 February 2021**
  - Cloze: There were 50,335 deaths involving the coronavirus (COVID-19) that occurred TEMPORALMASK, registered up to 4 July 2020 in England and Wales; of these, 46,736 had COVID-19 assigned as the underlying cause of death. **Answer: between 1 March and 30 June 2020**

- Answer Length 8
  - Cloze: From year ending Quarter 2 (Apr to June) 2010 to TEMPORALMASK, median house prices increased by over 20% in 26 towns and cities, all located in the south of England. **Answer: year ending Quarter 2 (Apr to June) 2015**
  - Cloze: From TEMPORALMASK to year ending Quarter 2 (Apr to June) 2015, median house prices increased by over 20% in 26 towns and cities, all located in the south of England. **Answer: year ending Quarter 2 (Apr to June) 2010**
  - Cloze: Sales of flats in the towns and cities rose from 18.3% in TEMPORALMASK to 30.5% in year ending Quarter 2 (Apr to June) 2015 as a proportion of all residential property sales. **Answer: year ending Quarter 4 (Oct to Dec) 1995**

- Answer Length 9
  - Cloze: In England, people identifying as Muslim, Hindu, Sikh, or Jewish had higher age-standardised mortality rates (ASMRs) for deaths involving coronavirus (COVID-19) than those identifying as Christian in TEMPORALMASK. **Answer: the period 24 January 2020 to 28 February 2021**

– Cloze: General government gross debt was £1,601.3 billion at TEMPORALMASK (87.5% of GDP), an increase of £79.9 billion compared with the end of the financial year ending March 2014 **Answer: the end of the financial year ending March 2015**

– Cloze: General government gross debt was £1,601.3 billion at the end of the financial year ending March 2015 (87.5% of GDP), an increase of £79.9 billion compared with TEMPORALMASK **Answer: the end of the financial year ending March 2014**

A.5 Amphetamine spreadsheet example

1. Extent and trends in drug use

1.01 Table 1.01 Proportion of 16 to 59 year olds reporting use of drugs ever in their lifetime, year ending December 1995 to year ending March 2020
1.02 Table 1.02 Proportion of 16 to 59 year olds reporting use of drugs in the last year, year ending December 1995 to year ending March 2020
1.03 Table 1.03 Proportion of 16 to 59 year olds reporting use of drugs in the last month, year ending December 1995 to year ending March 2020
1.04 Table 1.04 Estimates of numbers of illicit drug users, 16 to 59 year olds reporting use of drugs ever in their lifetime, year ending March 2002 to year ending March 2020
1.05 Table 1.05 Estimates of numbers of illicit drug users, 16 to 59 year olds reporting use of drugs in the last year, year ending March 2002 to year ending March 2020
1.06 Table 1.06 Estimates of numbers of illicit drug users, 16 to 59 year olds reporting use of drugs in the last month, year ending March 2018 to year ending March 2020
1.07 Table 1.07 Proportion of 16 to 24 year olds reporting use of drugs ever in their lifetime, year ending December 1995 to year ending March 2020
1.08 Table 1.08 Proportion of 16 to 24 year olds reporting use of drugs in the last year, year ending December 1995 to year ending March 2020
1.09 Table 1.09 Proportion of 16 to 24 year olds reporting use of drugs in the last month, year ending December 1995 to year ending March 2020
1.10 Table 1.10 Estimates of numbers of illicit drug users 16 to 24 year olds reporting use of drugs ever in their lifetime, year ending March 2007 to year ending March 2020
1.11 Table 1.11 Estimates of numbers of illicit drug users 16 to 24 year olds reporting use of drugs in the last year, year ending March 2007 to year ending March 2020
1.12 Table 1.12 Estimates of numbers of illicit drug users 16 to 24 year olds reporting use of drugs in the last month, year ending March 2018 to year ending March 2020
1.13 Table 1.13: Proportion of 16 to 74 year olds reporting use of any drug in the last year, year ending March 2020

2. Frequency of drug use in the last year

2.01 Table 2.01 Proportion of 16 to 59 and 16 to 24 year olds who use drugs frequently in the last year, year ending March 2015 to year ending March 2020
2.02 Table 2.02 Frequency of illicit drug use in the last year, 16 to 59 and 16 to 24, year ending March 2020
2.03 Table 2.03 Proportion of last year cannabis, powder cocaine and ecstasy users who were frequent users, 16 to 59 year olds, year ending March 2004 to year ending March 2020
2.04 Table 2.04 Frequency of cannabis use in the last month, 16 to 59 year olds who had used cannabis in the last month, year ending March 2011 and year ending March 2016 to year ending March 2020
2.05 Table 2.05 Frequency of powder cocaine use in the last month, 16 to 59 year olds who had used powder cocaine in the last month, year ending March 2018 to year ending March 2020

3. Drug use by personal, household and area characteristics and lifestyle factors

3.01 Table 3.01 Proportion of 16 to 59 year olds reporting use of illicit drugs in the last year by personal characteristics, year ending March 2020
3.02 Table 3.02 Proportion of 16 to 59 year olds reporting use of illicit drugs in the last year by household and area characteristics, year ending March 2020
3.03 Table 3.03 Proportion of 16 to 59 year olds reporting use of illicit drugs in the last year by age group, year ending December 1995 to year ending March 2020
3.04 Table 3.04 Proportion of 16 to 59 year olds reporting use of illicit drugs in the last year by sex, year ending December 1995 to year ending March 2020
3.05 Table 3.05 Proportion of 16 to 59 year olds reporting use of illicit drugs in the last year by frequency of nightclub visits in the past month, year ending December 1997 to year ending March 2020
3.06 Table 3.06 Proportion of 16 to 59 year olds reporting use of illicit drugs in the last year by frequency of pub/bar visits in the past month, year ending December 1997 to year ending March 2020
3.07 Table 3.07 Proportion of 16 to 59 year olds reporting use of illicit drugs in the last year by marital status, year ending December 1995 to year ending March 2020
3.08 Table 3.08 Proportion of 16 to 59 year olds reporting use of illicit drugs in the last year by English region and Wales, year ending December 1995 to year ending March 2020
3.09 Table 3.09 Proportion of 16 to 59 year olds reporting use of illicit drugs in the last year by Output area classification, year ending March 2014 to year ending March 2020
3.10 Table 3.10 Proportion of 16 to 59 year olds reporting use of non-prescribed prescription-only painkillers for medical reasons in the last year by personal characteristics, year ending March 2016 to year ending March 2020
3.11 Table 3.11 Proportion of 16 to 59 year olds reporting use of non-prescribed prescription-only painkillers for medical reasons in the last year by household and area characteristics, year ending March 2016 to year ending March 2020

4. New psychoactive substances (NPS) and nitrous oxide

4.01 Table 4.01 Prevalence of new psychoactive substances (NPS) use, 16 to 59 and 16 to 24, by sex, year ending March 2015 to year ending March 2020
4.02 Table 4.02 Estimates of numbers of new psychoactive substance (NPS) users in the last year, 16 to 59 and 16-24, by sex, year ending March 2015 to year ending March 2020
4.03 Table 4.03 Prevalence of nitrous oxide use, 16 to 59 and 16 to 24 year olds, by sex, year ending March 2013 to year ending March 2020
4.04 Table 4.04 Estimates of numbers of nitrous oxide users in the last year, 16 to 59 and 16-24, by sex, year ending March 2013 to year ending March 2020
4.05 Table 4.05 Frequency of new psychoactive substance (NPS) use in the last year, 16 to 59 year olds who had used NPS in the last year, year ending March 2018 to year ending March 2020
4.06 Table 4.06 Prevalence of NPS in the last year by associated behaviours, 16 to 59 and 16 to 24 year olds, year ending March 2017 to year ending March 2020
4.07 Table 4.07 Nature and sources of new psychoactive substance (NPS) used (including Nitrous Oxide), 16 to 59 and 16 to 24 year olds, year ending March 2017 to year ending March 2020
4.08 Table 4.08 Immediate sources of new psychoactive substances (NPS) (including nitrous oxide) used and illicit drugs, 16 to 59 and 16 to 24 year olds, year ending March 2020

5. Perceived ease of obtaining illicit substances

5.01 Table 5.01 Ease of obtaining illegal drugs and new psychoactive substance (NPS)/Nitrous Oxide within 24 hours, if wanted to obtain, year ending March 2017 to year ending March 2020
5.02 Table 5.02 Ease of obtaining illegal drugs within 24 hours, if wanted to obtain, year ending March 2020

Figure A.1: Contents worksheet from Amphetamine example, made up by 41 worksheets; naming scheme follows section numbering; cannot identify contents without the use of this Contents worksheet.