Privacy Pitfalls of Online Service Terms and Conditions: a Hybrid Approach for Classification and Summarization

Emilia Lukose
Dept. of Computer Science
University of Surrey
Guildford, U.K.
el00490@surrey.ac.uk

Suparna De
Dept. of Computer Science
University of Surrey
Guildford, U.K.
s.de@surrey.ac.uk

Jon Johnson
UCL Social Research Institute
University College London
London, UK
jon.johnson@ucl.ac.uk

Abstract

Verbose and complicated legal terminology in online service terms and conditions (T&C) means that users typically don't read these documents before accepting the terms of such unilateral service contracts. With such services becoming part of mainstream digital life, highlighting Terms of Service (ToS) clauses that impact on the collection and use of user data and privacy are important concerns. Advances in text summarization can help to create informative and concise summaries of the terms, but existing approaches geared towards news and microblogging corpora are not directly applicable to the ToS domain, which is hindered by a lack of T&C-relevant resources for training and evaluation. This paper presents a ToS model, developing a hybrid extractive-classifier-abstractive pipeline that highlights the privacy and data collection/use-related sections in a ToS document and paraphrases these into concise and informative sentences. Relying on significantly less training data (4313 training pairs) than previous representative works (287,226 pairs), our model outperforms extractive baselines by at least 50% in ROUGE-1 score and 54% in METEOR score. The paper also contributes to existing community efforts by curating a dataset of online service T&C, through a developed web scraping tool.

1 Introduction

Despite legislative advances such as the European Union's General Data Protection Regulation (GDPR)[1] regarding specific, informed and unambiguous consent for the collection and use of personal data on the Internet (Kubíček et al., 2022), understanding how online services can read, edit, distribute and sell user data, as documented in their Terms of Service (ToS), remains out of reach for the typical user, with most (98%) consenting to the terms without reading the documents in their entirety (Obar and Oeldorf-Hirsch, 2018). Two major factors contributing to this are the length of the documents and the ambiguous and complicated terminology used (Manor and Li, 2019), with users unable to interpret the implications of the terms of such a legally-binding unilateral contract. In addition to the implication for users' rights, the distribution and use of user data is also important for companies looking to use third-party services in their product.

This can be exemplified with the case of the Global Science Research (GSR) company tasking Cambridge Analytica to build psychological profiles of users through a quiz app, which also collected information from the users' Facebook friends, allowing the company to acquire data from millions of unwitting Facebook users[2]. This data was then matched with existing voter datasets, enabling aggressive voter-targeting operations in the 2016 US presidential election[3]. Delving into the app's ToS reveals that it states: "We collect any information that you choose to share with us ...this may include, inter alia, the name, demographics, [. . .] of your profile and of your network." In addition to this, they permit GSR to "edit and sell" user data by accepting the conditions (Research, 2014). When queried in 2018 if it had read and evaluated the terms and conditions for the app, Facebook responded: "We did not read all the terms and conditions"[4].

With a lack of regulations around standard terms in which consumer contracts should be drafted (Drawzeski et al., 2021), a condensed equivalent of the salient points of a ToS document can em-

[1] https://eugdpr.org/

[2] https://www.theguardian.com/news/2018/mar/17/cambridge-analytica-facebook-influence-us-election

[3] https://www.theguardian.com/us-news/2015/dec/11/senator-ted-cruz-president-campaign-facebook-user-data

[4] https://www.mercurynews.com/2018/04/26/facebook-didnt-read-terms-and-conditions-for-app-behind-cambridge-analytica

Proceedings of the Natural Legal Language Processing Workshop 2022, pages 65 - 75
December 8, 2022 ©2022 Association for Computational Linguistics

power users to understand their rights and avoid privacy invasion and legal disagreements. Summarization, which condenses text into a shorter form whilst keeping the most crucial and informational parts intact, is an intuitive approach for replacing unnecessary (and in some cases, intentionally convoluted) long text with a digestible summary.

Building on recent community approaches to annotate and curate ToS sentence/summary pairs (Manor and Li, 2019; Keymanesh et al., 2020), we propose a hybrid extractive-classifier-abstractive model that can extract ToS sentences related to privacy and data collection/use and paraphrases these into concise and informative ToS highlights. The hybrid model forms part of a Web application and browser plugin that enables users to view an at-a-glance summary of any online service (specified through its URL) T&C. We also contribute to community efforts for curating a ToS dataset by developing a web scraping engine to build a novel ToS dataset from 163 different online services. The proposed hybrid model addresses limitations of existing works as it relies on significantly less training data (4313 training pairs) than previous representative works in hybrid extractive-abstractive methods (See et al., 2017) (287,226 pairs). The summarization results are compared against the baseline unsupervised, extractive techniques, achieving significant improvements in performance (50% improvement in ROUGE-1 score versus the best performing baseline, and 54% in METEOR score).

2 Related Work

This section explores the state-of-the-art community efforts and research within the domain of T&C and text summarization. The “Terms of Service; Didn’t Read" (TOS;DR)[5] community-driven project highlights alarming statements in ToS. Services are given grades ranging from A-E based on the severity of the terms listed; E being very serious concerns. Summaries are manually submitted by the TOS;DR community, which limits the scope of summarization to only those that already exist in the database. Moreover, manually reading and analysing long documents of terms is a laborious and time-consuming task. The TL;DRLegal[6] website hosts community-submitted software license summaries that are peer-reviewed by the website managers, with the same manual-process limitations as TOS;DR.

A notable work in ToS data curation is that by Manor & Li (Manor and Li, 2019) with 446 sets of contract sections and corresponding reference summaries from TOS;DR and TL;DRLegal, thus presenting the first dataset in this genre.

Automated summarization techniques have been applied successfully to curated news (e.g. CNN/DailyMail corpus (Hermann et al., 2015)), scientific articles (Yasunaga et al., 2019) or microblogging (e.g. Large Scale Chinese Short Text Summarization (LCTCS) (Hu et al., 2015)) corpora. Categorized either as extractive (Nallapati et al., 2017; Keymanesh et al., 2020) or abstractive (See et al., 2017; Gehrmann et al., 2018) methods, existing summarization approaches are however, not directly applicable to the T&C domain. Extractive approaches work by selecting the most salient sentences for the summary (Xiao et al., 2020) and deciding on their order of presentation. They rely on the structural features of documents, i.e. typically news, scientific articles, where the title and abstract/first few lines of the document, contain a snapshot of the key content. These heuristics do not translate well for ToS documents, which have differing structures for different jurisdictions and where the terminological nuances in legal language are difficult to capture (Drawzeski et al., 2021). Moreover, as the resultant extractive summary matches source sentences word-for-word, complex legal terms in the summary may still confuse the reader (Manor and Li, 2019). Existing works for privacy policies and ToS include the extractive approach of a supervised Convolutional Neural Network (CNN) model (Keymanesh et al., 2020) to predict which content has the most risk of unsafe data practices, that is followed by extracting a calculated amount of sentences with the highest risk score. The model did not perform well when compared to the TOS;DR summaries, as a fully-extractive approach cannot mimic the human-like qualities in the TOS;DR summaries, and also suffers as it generates "legalese" rather than plain English, making it less accessible.

Abstractive methods, on the other hand, generate concise summaries by compressing and paraphrasing, but are weak at content selection and prone to information loss (Xiao et al., 2020). These supervised approaches also require a large corpus of parallel document/summary pairs for training neural models and their evaluation. Unlike the

[5] https://tosdr.org/, CC BY-SA 3.0

[6] https://tldrlegal.com/

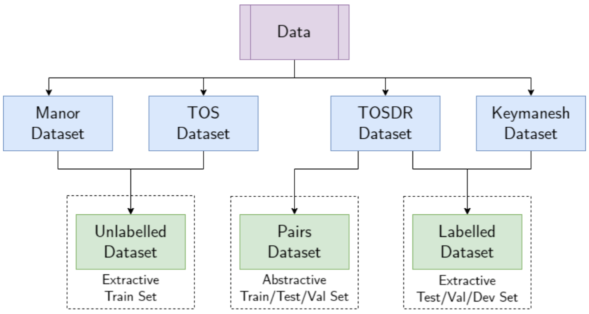


Figure 1: An overview of the datasets used for training and evaluation of the ToS Hybrid Model. Manor dataset - open source dataset from Manor & Li (Manor and Li, 2019), Keymanesh dataset - (Keymanesh et al., 2020).

news/microblogging genres, where such large curated datasets are available for training, resources for ToS documents are currently not large enough, being "intended for evaluation, rather than training" (Manor and Li, 2019). Other abstractive models include pointer-generator models with coverage mechanism (See et al., 2017), which use pointing (Vinyals et al., 2015), and a hybrid extractive-abstractive approach to improve accuracy and handle unknown words.

3 Data

This section describes the datasets (shown in Figure 1) created and compiled for training and evaluation of the hybrid ToS model.

3.1 ToS Dataset

163 ToS documents retrieved from 387 website domains, representing a range of online service categories compiled from Kaggle[7] and The Moz[8] datasets. This dataset of text files (31,752 sentences), each corresponding to a terms page, is made available on github (*https://github.com/supdey/tos-dataset*) as a contribution to the community effort on ToS dataset curation. The mean of 217.5 sentences and 4775.5 words per document and 22.1 words per sentence (std 20.2) supports similar observations in the literature about ToS documents being very long on average.

3.2 TOS;DR Dataset

The TOS;DR community dataset containing 17,109 data entries, consists of quotes from ToS documents paired with human-written summaries ("titles") and is used in combination with other datasets for model training and evaluation. Each title has an assigned class: good, bad, blocker (also bad) and neutral.

3.3 Sentence Classification Dataset

Labelled and Unlabelled Datasets used for the extractive component. The Labelled Dataset combines the TOS;DR and Keymanesh[9] datasets, with 15,839 labelled sentences.

Both datasets are modified to replace the "bad" and "blocker" classes with "1", signifying importance, with the "good" and "neutral" classes replaced with "0" signifying unimportance.

The Unlabelled Dataset combines ToS sentences from the ToS and the Manor & Li dataset, as training data for weak-supervised learning in the extractive component.

3.4 Terms and Reference Summaries Dataset

The Pairs Dataset used for the abstractive component is created by selecting rows with quote-summary pairs from the TOS;DR dataset. An analysis of the abstraction level of the summaries in terms of the number of n-grams that only appear in the reference summaries and not in the quote sentences shows that 67.5% of words and 91.6% of bigrams in the summaries did not appear in the original quote, showing significant abstraction.

4 Methodology

The methodology for the ToS hybrid model proposes to automate the summarization and grading (sentence extraction) process, allowing a broader scope of companies and websites to be analyzed while also removing the manual step of summary review.

4.1 Extractive Component

The extractive component creates a classifier for labelling ToS sentences as important or unimportant, in order to extract "important" sentences from a ToS document. An overview of the training process is shown in Figure 2.

4.1.1 Weak Supervision for Sentence Labelling

The workflow of the weak supervision approach used to label sentences programmatically is shown

[7] https://www.kaggle.com/datasets/bpali26/popular-websites-across-the-globe

[8] https://moz.com/top500

[9] www.github.com/senjed/Summarization-of-Privacy-Policies

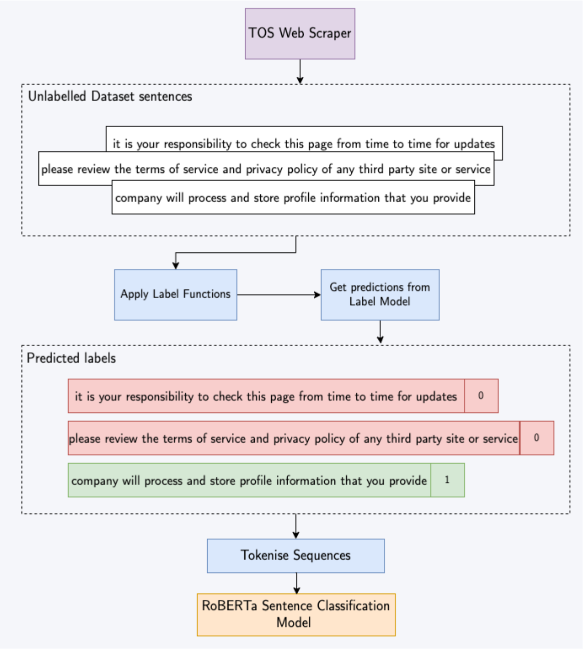

TOS Web Scraper
Unlabelled Dataset sentences
it is your responsibility to check this page from time to time for updates
please review the terms of service and privacy policy of any third party site or service
company will process and store profile information that you provide
Apply Label Functions
Get predictions from Label Model
Predicted labels
it is your responsibility to check this page from time to time for updates
0
please review the terms of service and privacy policy of any third party site or service
0
company will process and store profile information that you provide
1
Tokenise Sequences
RoBERTa Sentence Classification Model

Figure 2: An overview of the training process for the extractive component, which feeds into a RoBERTa classifier to extract 'important' sentences from a ToS document.

in Figure 3.

It aims to learn a classification model that takes a ToS sentence $x \in \mathcal{X}$ and predicts its label $y \in \mathcal{Y}$, where $\mathcal{Y} = \{0, 1\}$. The training data used for this task is the 'Unlabelled' Dataset shown in Figure 1. The labels are generated from user-defined blackbox labelling functions, $\lambda : \mathcal{X} \rightarrow \mathcal{Y} \cup \{-1\}$, that take in a sentence and output an important (1) or unimportant (0) label, where -1 is used to denote that the function abstains. These functions, shown in Table 1 are programmatic rules and heuristics, which use methods such as keyword-searching and pattern-matching with regex. It is possible for labelling functions not to label every data point; they can also overlap and conflict with each other by assigning the same or different labels to a single point.

The labelling functions are developed as a result of examining the Labelled Dataset, which is split into a 60:30:10 test, validation and development set. The development set is used to inform the decisions behind the labelling functions, with the sentences analysed to find common vocabulary, phrases and verbs after stopword removal, for sentences labelled important and unimportant. This is kept separate from the training data in order to avoid overfitting by introducing rules that are too specific. The validation set is used for hyperparameter tuning and checking performance without looking at the test set scores. The test set is used for final evaluation. The labelling functions are evaluated by examining their: (1) coverage: percentage of the dataset that the function labels, (2) overlaps: dataset percentage that the function and at least one other function also labels, and (3) conflicts: the percentage of the dataset that the function and at least one other function disagree on, with the goal being to increase coverage, while avoiding false positives. Some of the labelling functions involve using regex to detect when verbs such as “collect" and “sell" are used next to references to personal data. A list of these verbs are generated using NLTK[10], which is able to return synonyms for given words using WordNet (Miller, 1995). While developing labelling functions, random rows are checked to determine whether the labelling matched intuition or if false positives are being introduced. Moreover, labelling functions are compared by grouping data points by their predicted labels to determine which has the most impact.

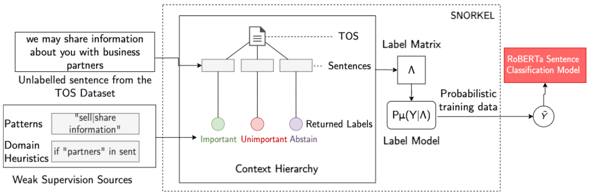


Figure 3: Weak supervision for sentence labelling, adapted from (Ratner et al., 2020).

For m unlabelled sentences and n labelling functions (in this case, $n = 8$), the labelling functions are applied to the sentences to produce a matrix of labelling function outputs (denoted as Label Matrix in Figure 3): $\Lambda \in (\mathcal{Y} \cup \{-1\})^{m \times n}$. This label matrix is then fed to the LabelModel $P_\mu(Y|\Lambda)$, parameterised by a vector of source correlations and accuracies μ. The LabelModel uses a modelling approach similar to that proposed in (Ratner et al., 2018), to produce a single vector of probabilistic training labels $\tilde{Y} = (\tilde{y_1}..., \tilde{y_m})$, where $\tilde{y_i} \in [0, 1, -1]$. After the abstains have been filtered out, the training labels are used to train a Robustly Optimised BERT Pretraining Approach (RoBERTa) classifier (Liu et al., 2019).

4.1.2 Classifier for Sentence Extraction

The classifier is able to generalise beyond the outputs of the labelling function, increasing cov-

[10] https://www.nltk.org/

Function Name	Polarity	Explanation	Example Match
Important Keyword Lookup	1	Match references to advertisements & web beacons.	website uses cookies scripts and web beacons
Data Regex	1	Match if verbs such as “collect" and “sell" are associated with personal data	we use tracking tools to collect information from you
Waive Regex	1	Match sentences related to user’s rights waiver	you waive your right to participate in any class group
Self-reference Regex	0	Match sentences mentioning the terms document it belongs to	these terms of use went into effect in June
Unimportant Phrase Lookup	0	Match sentences containing unnecessary information such as support information or outlining user rights	contact us for press inquiries & more information
Unimportant Word Lookup	0	Match sentences containing words indicating users should check other areas of the website	the table below explains the cookies we use
Rules Regex	0	Match sentences informing users that they should not perform certain actions. To identify “risky" terms, these sentences are classified as unimportant	you must be 13 years or older to use this site
No Data Regex	0	Opposite of Data Regex - match sentences informing users that the service is NOT using their data	we do not sell user data

Table 1: Labelling functions’ definitions used to determine if a ToS sentence is important (polarity = 1) or unimportant (polarity = 0).

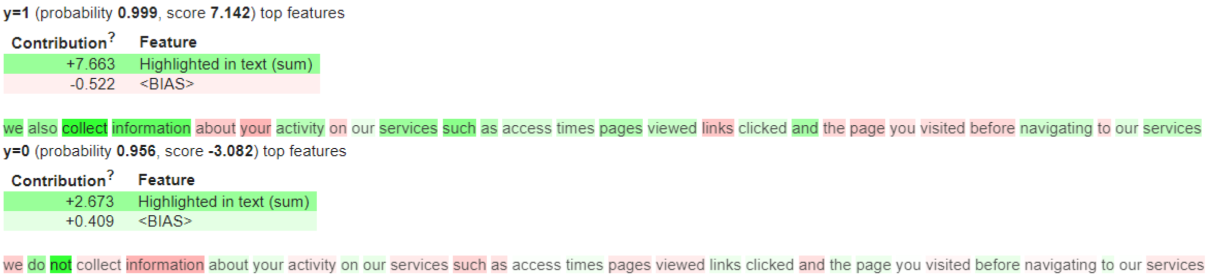
erage and robustness on unseen ToS sentences. RoBERTa, a modified pre-trained Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) model, is used as the end classifier model for “important" sentence extraction, due to its modifications such as the removal of the next sentence prediction objective, longer training time, bigger batches, training on longer sequences and dynamically changing the masking pattern applied to the training data, which is known to improve performance on downstream tasks. Exploration of the LabelModel outputs shows that the dataset is highly imbalanced; therefore, the classifier model hyperparameters are fine-tuned by using rebalanced labelled data. The optimal hyperparameter settings are found to be: number of epochs: 2, a batch size of 16 and the AdamW optimiser with a learning rate of 3e-5.

Figure 4 shows the working of the classifier model with two sample sentences, showcasing the contribution of the top features contributing towards a prediction. In the top image, the green-highlighted text shows which words contributed towards the “important" label. Higher transparency implies less contribution. The words “collect information", highlighted strongly, indicate a major contribution from this phrase towards classifying this sentence as risky. In contrast, with the word "also" changed to "do not" in the same sentence (bottom image of Figure 4), changes its classification to an “unimportant sentence", with a strong indication that the word "not" had major contribution to this decision - this is expected given that the sentence is negated. Moreover, the word “information" is slightly highlighted in red, showing that it contributes towards an “important" classification.

4.2 Abstractive Component

With the goal of this research being not to summarise the entire contents of a ToS but to first extract the important sentences and then to paraphrase each one, a sequence-to-sequence (seq2seq) model with attention is chosen due to its ability to retain context. The model architecture is shown in Figure 5.

Sentence tokens from terms documents are fed one-by-one into an encoder containing a bidirectional LSTM layer, which produces a sequence of encoder hidden states h_i. In each step t, the decoder (a single-layer LSTM) receives the word embedding of the previous word. During model training, this is the previous word of the reference summary,



y=1 (probability **0.999**, score **7.142**) top features

Contribution?	Feature
+7.663	Highlighted in text (sum)
-0.522	<BIAS>

we also collect information about your activity on our services such as access times pages viewed links clicked and the page you visited before navigating to our services

y=0 (probability **0.956**, score **-3.082**) top features

Contribution?	Feature
+2.673	Highlighted in text (sum)
+0.409	<BIAS>

we do not collect information about your activity on our services such as access times pages viewed links clicked and the page you visited before navigating to our services

Figure 4: Two sentences predicted by the fine-tuned classifier; colours refer to contribution and not the actual classes themselves. (Top) A sentence classified as “important"; (bottom) a negated sentence classified as "unimportant".

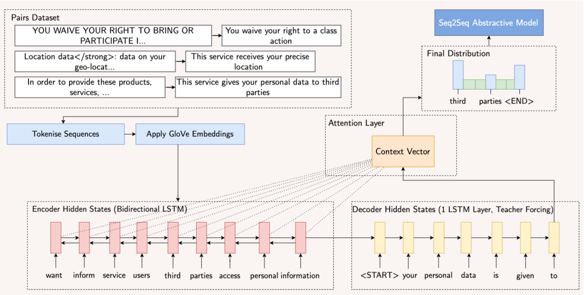

Pairs Dataset
YOU WAIVE YOUR RIGHT TO BRING OR PARTICIPATE I...
You waive your right to a class action
Location data</strong>: data on your geo-locat...
This service receives your precise location
In order to provide these products, services, ...
This service gives your personal data to third parties
Seq2Seq Abstractive Model
Final Distribution
third
parties
<END>
Tokenise Sequences
Apply GloVe Embeddings
Attention Layer
Context Vector
Encoder Hidden States (Bidirectional LSTM)
want
inform
service
users
third
parties
access
personal information
Decoder Hidden States (1 LSTM Layer, Teacher Forcing)
<START>
your
personal
data
is
given
to

Figure 5: Overview of the training process for the abstractive component.

while during testing, it is the previous word output by the decoder. The decoder has decoder state s_t. The attention distribution (probability distribution over the source words) a^t is calculated using Bahdanau Attention (Bahdanau et al., 2015):

$$e_i^t = v^T \tanh(W_h h_i + W_s s_t + b_{\text{attn}}) \quad (1)$$

$$a^t = \text{softmax}(e^t) \quad (2)$$

where v, W_h, W_s and b_{attn} are learnable parameters. The distribution is used to produce the context vector h_t^*, which is the weighted sum of the encoder hidden states as shown in Figure 5. This can be seen as a fixed-size representation of what has been read from the terms sentence for this step. The calculation of the context vector is as follows:

$$h_t^* = \sum\nolimits_i a_i^t h_i \quad (3)$$

This context vector is then concatenated with the decoder state s_t to produce the vocabulary distribution P_{vocab}. This is the probability distribution over all words in the vocabulary, which is calculated as follows:

$$P_{\text{vocab}} = \text{softmax}(V'(V[s_t, h_t^*] + b) + b') \quad (4)$$

where V, V', b and b' are learnable parameters. This distribution provides the final distribution that is used to predict words w:

$$P(w) = P_{\text{vocab}}(w) \quad (5)$$

The loss for timestep t during training is the negative log likelihood of the target word w_t^* for that timestep:

$$\text{loss}_t = -\log P(w_t^*) \quad (6)$$

The overall loss for the whole sequence is calculated as follows:

$$\text{loss} = \frac{1}{T} \sum\nolimits_{t=0}^{T} \text{loss}_t \quad (7)$$

The LSTM networks each use 256-dimensional hidden units and pre-trained GloVe word embedding (Pennington et al., 2014), that has been pre-trained on a dataset of one billion tokens and a vocabulary of 400,000 words. The vocabulary size is limited by filtering out rare words to prevent overfitting. For the source sentences, a word is considered “rare" when the number of occurrences throughout all the texts is less than 4; for the summaries, this is 6, due to the average summary length being shorter. The resulting vocabulary size for the source sentences is 2,413, and for the target sentences 308. Regular and recurrent dropout are applied to the LSTM layers to reduce overfitting, with dropout values set to 0.4, with the exception for the recurrent dropout of the decoder LSTM which is set to 0.2. Softmax activation is used in the final dense decoder layer, as the output can be interpreted as a probability distribution vector that helps determine the final output of sequence tokens. The optimiser is Root Mean Squared Propagation (RMSProp). The loss function used is sparse categorical cross-entropy due to the Y inputs consisting of integer sequences that are mutually exclusive. This function also has memory and computation usage benefits, as the classes are defined by single integers as opposed to entire vectors.

5 Results and Evaluation

5.1 Dataset and Ground-Truth Construction

The ground truth dataset is created by combining the TOS;DR and Keymanesh datasets, with labels of 1955 important and unimportant sentences. To get plain English summaries, cleaned sentences (following pre-processing) from the Keymanesh dataset are matched with sentences in the TOS;DR dataset to create the ground-truth summaries. For sentences with no corresponding reference summary, the ground-truth was taken as the cleaned, stopword-removed version of the sentence text, ensuring that all sentences labelled 'important' feature in the evaluation. The resulting dataset has 263 rows of plain English summaries and 1692 rows of the cleaned, stopword-removed version of the quote text.

The evaluation baselines are executed on the entire ToS contracts retrieved by the web-scraping tool, which returned the ToS of 102 services taken from the Keymanesh dataset.

After filtering for services that contain at least one 'important label' in the Keymanesh dataset, 45 services, with 10231 sentences, are used for evaluation.

5.2 Summarization Baselines

We compare the performance of our hybrid summarisation model with the following unsupervised baselines:

- TextRank (Mihalcea and Tarau, 2004): uses the PageRank algorithm to extract the most important keywords from a ToS, based on the similarity between phrases.
- KLSum (Haghighi and Vanderwende, 2009): minimises the Kullback-Lieber (KL) divergence between the ToS and proposed summary by greedily selecting sentences.
- Lead-K (See et al., 2017): extracts the first k sentences until the word limit is reached.
- K-Random: picks random sentences until a word limit is reached. This baseline was run 10 times to get the average results.

Following pre-processing, sentences from TextRank and KLSum were limited by the average sentence count from ground-truth summaries (i.e. 4). Summaries from Lead-K and K-Random were limited by the average word count (93) from ground-truth summaries.

5.3 Evaluation Metrics

The summarisation was evaluated by computing the average F1-score for ROUGE-1, ROUGE-2, and ROUGE-L metrics (Lin and Hovy, 2002), as well as the METEOR score (Denkowski and Lavie, 2014).

ROUGE-N measures the number of matching n-grams between the generated summaries and the ground-truth summary, with ROUGE-1 referring to unigram overlaps and ROUGE-2 referring to bigram overlaps. ROUGE-L calculates the Longest Common Subsequence - identifying the longest overlapping sequence of tokens. The METEOR metric was found to be a better evaluation system as this rewards not only exact word matches but also matching stems, synonyms and paraphrases.

5.4 Results

5.4.1 Summarization Results

Model evaluation results are shown in Table 2. The ToS hybrid model significantly outperforms the extractive baselines. When compared against the best performing baseline for each metric, there is a 49.7% improvement in ROUGE-1, 114.6% in ROUGE-2, 53.5% in ROUGE-L and 53.6% improvement in METEOR scores. This indicates that the ToS hybrid model can generate summaries that are easier to read and understand. This is important given that the aim of the TOS;DR summaries is to be simple and concise.

While the Lead-K baseline performed well in summarization tasks in existing works (See et al., 2017) for news articles and headlines datasets, even outperforming abstractive models using pointer-generators, the results of our work show it to have the worst performance. The success of the lead-3 baseline in (See et al., 2017) can be attributed to the structure of news articles, which contain the most crucial information at the beginning, and the use of the first three sentences of the article as a summary by lead-3. In contrast, the structure of a ToS document often begins with definitions of phrases used throughout the document and an introduction to the service(s) offered. This is often not considered important information, as it is merely an explanation of the document's contents. This observation shows that using a dataset specifically focused on the domain of T&C for this task significantly boosts performance, highlighting the need for collecting more ToS data and the usefulness of the developed ToS web-scraper tool.

Table 2: Evaluation results of the ToS Hybrid Model in comparison to the baselines.

Model	ROUGE-1	ROUGE-2	ROUGE-L	METEOR
ToS Hybrid Model	**19.45**	**7.21**	**18.41**	**16.25**
TextRank	12.99	3.36	11.99	10.58
KLSum	12.94	1.79	11.85	8.58
K-Random	10.72	1.71	10.15	8.94
Lead-K	10.41	1.47	9.88	8.67

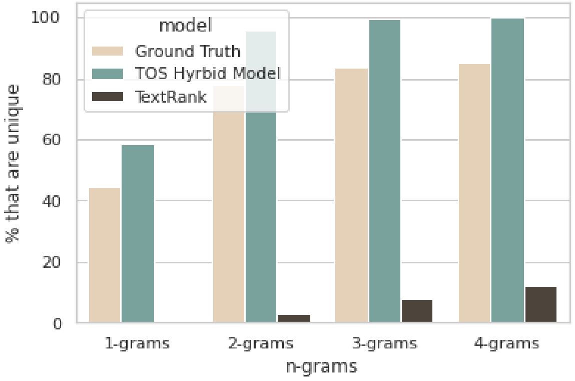


Figure 6: Unique n-grams in the of the Ground-Truth, ToS Hybrid Model and TextRank summaries.

5.4.2 Abstraction and Compression Level

The summaries from the ground-truth, ToS Hybrid Model and TextRank have been compared against the original ToS documents for each service, to assess their effectiveness in terms of the level of abstraction and compression. The abstraction level is calculated by the number of n-grams that only appear in the summaries and not in the ToS documents (See et al., 2017). As shown in Figure 6, the hybrid model has high levels of abstraction and shows that 58.7% of the words in the summaries are not present in the ToS document, demonstrating its ability to generate new words. As expected, the summary of the TextRank model does not contain any new words as it is an extractive approach. The ground-truth summaries appear to be slightly less abstractive than that of our hybrid model; we can assume this is because the ground-truth contains stopword-cleaned sentences where it is unable to find a summary from the Pairs Dataset. The mean compression rate is 0.026 (std 0.014), showing that the summaries are significantly shortened.

TOS: <*truncated*> In order to provide the Service, UpCloud may process personal data on behalf of the Customer as a data processor for the purposes of providing the Service UpCloud processes certain personal data also as a data controller Such personal data includes, inter alia, data of the Customer's contact persons, invoicing details and other personal data of Customer's contact persons which UpCloud processes in order to maintain the customer relationship ... In connection with the use of the Service, the Customer may transfer various data to UpCloud for processing on behalf of the Customer <*truncated*>

Reference Summary:

- blocking first party cookies may limit your ability to use the service
- the service provider makes no warranty regarding uninterrupted timely secure or error free service
- the service is provided as is and to be used at your sole risk
- your data may be processed and stored anywhere in the world
- third parties are involved in operating the service
- your personal data may be sold or otherwise transferred as part of a bankruptcy proceeding or other type of financial transaction

TOS Hybrid Model:

- this service gives your personal data to third parties involved in its operation
- your personal data is used to employ targeted third party advertising
- your personal information is used for many different purposes
- your personal data may be sold or otherwise transferred as part of bankruptcy proceeding or other type of financial transaction
- the service provider makes no warranty regarding uninterrupted timely secure or error free service
- your personal data may be used for marketing purposes
- you are tracked via web beacons tracking pixels browser fingerprinting and or device fingerprinting

KLSum:

- upcloud shall provide the customer with the service which is specified in the order.
- upcloud shall offer compensation to the customer for all unscheduled interruptions in the provision of the service in accordance with the sla in force from time to time.
- in order to provide the service upcloud may process personal data on behalf of the customer as a data processor for the purposes of providing the service.
- in connection with the use of the service the customer may transfer various data to upcloud for processing on behalf of the customer.

Lead-K:

- these terms apply to all our existing users as of and any new users as of
- please note not everybody has a law degree or speak english natively so we have added helpful tl dr in plain english that summarize each term
- these are only meant to help you understand the general meaning and are not legally binding terms of service <*truncated*>

Figure 7: ToS Hybrid Model, KLSum and Lead-K summaries compared for UpCloud's ToS (17 May 2018).

5.4.3 Case Study and Results Discussion

Figure 7 shows sample summaries generated by the ToS hybrid model, KLSum and Lead-K, with the underlying ToS document containing 290 sentences. The ToS hybrid model succeeds in producing a bullet-point format summary written in plain English. When compared against the reference summary, we can see that both of them mention the use of third parties. There are two identical sentences from both summaries; indicating that the abstractive model has overfitted to the Pairs Dataset, which contains TOS;DR template summaries. The ground-truth dataset contains "important" sentences which may not be considered important by some users - e.g., the reference summary sentence "blocking first party cookies may limit your ability to use the service". However, this is subjective, and with the availability of more training data, the abstractive model can learn to summarize *any* sentence within a ToS. On comparing the ground-truth to the hybrid model's summary outputs, there are cases where it does not seem to summarize certain sentences accurately and instead may output a different sentence similar to the TOS;DR template summaries. A likely reason for this is the lack of training data for the abstractive component. The Pairs dataset has 5,326 rows, of which 4,313 are used for training. This is 98% less training data than that in the pointer-generator model in (See et al., 2017). Moreover, the extractive-abstractive models in (Nallapati et al., 2017) used 3.8M training examples.

6 Conclusions

In this paper, we proposed a domain-aware hybrid extractive-abstractive model that highlights privacy and data collection sections in a ToS document and paraphrases these into concise and informative sentences. A novel dataset is also created using a developed web-scraping tool, with the purpose of automatically fetching ToS documents from any online service. The dataset used for classification training was found to be highly imbalanced; despite this, the hybrid model performed well in ROUGE and METEOR scores when compared against unsupervised, extractive baselines. To resolve the imbalance problem, the data was resampled before being used in the classifiers for training, which reduced the false negative rate by 64%. However, this did increase the false positive rate, which implies that the extractive classifier is more inclined to incorrectly label sentences as important. Given that the abstractive model is only trained on important sentences, this can lead to incorrect warnings by the ToS model. In the context of this paper, to maintain the integrity of legal concepts, this can still be seen as the preferable outcome since users can verify statements made by the model by reviewing the original ToS if necessary.

More training data for both the classifier and abstractive model can improve performance; this can be obtained by the developed web-scraping tool, in addition to future TOS;DR community contributions. This would result in more data for the classifier post-resampling, which in turn would help the imbalance issue and false positive rate. Another direction for future work would be testing the generated summaries for comprehension through qualitative user studies, from participants recruited through platforms such as MTurk or Prolific.

Acknowledgements

This research is supported by the Science and Technology Facilities Council (STFC) DiRAC-funded 'Understanding the multiple dimensions of prediction of concepts in social and biomedical science questionnaires' project, grant number ST/S003916/1. The second author also acknowledges support through the UKRI Strategic Priority Fund as part of the wider Protecting Citizens Online programme (Grant number: EP/W032473/1) associated with the National Research Centre on Privacy, Harm Reduction and Adversarial Influence Online (REPHRAIN).

References

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR, Conference Track Proceedings*.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 376–380, Baltimore, Maryland, USA. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North*, pages 4171–4186.

Kasper Drawzeski, Andrea Galassi, Agnieszka Jablonowska, Francesca Lagioia, Marco Lippi, Hans Wolfgang Micklitz, Giovanni Sartor, Giacomo Tagiuri, and Paolo Torroni. 2021. A corpus for multilingual analysis of online terms of service. In *Proceedings of the Natural Legal Language Processing Workshop*, NLLP '21 @EMNLP '21, pages 1–8, Punta Cana, Dominican Republic. ACM.

Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109, Brussels, Belgium. Association for Computational Linguistics.

Aria Haghighi and Lucy Vanderwende. 2009. Exploring content models for multi-document summarization. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 362–370, Boulder, Colorado. Association for Computational Linguistics.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1693–1701, Cambridge, MA, USA. MIT Press.

Baotian Hu, Qingcai Chen, and Fangze Zhu. 2015. LCSTS: A large scale Chinese short text summarization dataset. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1967–1972, Lisbon, Portugal. Association for Computational Linguistics.

Moniba Keymanesh, Micha Elsner, and Srinivasan Parthasarathy. 2020. Toward domain-guided controllable summarization of privacy policies. In *Proceedings of the Natural Legal Language Processing Workshop, co-located with the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2020)*, NLLP '20 @KDD'20, pages 18–24, San Diego, CA. ACM.

Karel Kubíček, Jakob Merane, Carlos Cotrini, Alexander Stremitzer, Stefan Bechtold, and David Basin. 2022. Checking websites' gdpr consent compliance for marketing emails. *Proceedings on Privacy Enhancing Technologies (PoPETs)*, 2022(2):282–303.

Chin-Yew Lin and Eduard Hovy. 2002. Manual and automatic evaluation of summaries. In *Proceedings of the ACL-02 Workshop on Automatic Summarization - Volume 4*, AS '02, page 45–51, USA. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Laura Manor and Junyi Jessy Li. 2019. Plain english summarization of contracts. In *Proceedings of the Natural Legal Language Processing Workshop*, NLLP '19 @NAACL 2019, pages 1–11, Minneapolis, Minnesota. ACM.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 3075–3081. AAAI Press.

Jonathan A Obar and Anne Oeldorf-Hirsch. 2018. The biggest lie on the internet: Ignoring the privacy policies and terms of service policies of social networking services. *Information, Communication and Society*, pages 128–147.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics (ACL).

Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2020. Snorkel: Rapid training data creation with weak supervision. *The VLDB Journal*, 29:709–730.

Alexander Ratner, Braden Hancock, Jared Dunnmon, Frederic Sala, Shreyash Pandey, and Christopher Ré. 2018. Training complex models with multi-task weak supervision. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019*, pages 4763–4771. AAAI Press.

Global Science Research. 2014. Thisisyourdigitallife app application end user terms and conditions.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems (NIPS)*, volume 28.

Liqiang Xiao, Lu Wang, Hao He, and Yaohui Jin. 2020. Copy or rewrite: Hybrid summarization with hierarchical reinforcement learning. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20)*, volume 34, pages 9306–9313, Punta Cana, Dominican Republic. AAAI press.

Michihiro Yasunaga, Jungo Kasai, Rui Zhang, Alexander R. Fabbri, Irene Li, Dan Friedman, and Dragomir R. Radev. 2019. Scisummnet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks. In *33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019*, pages 7386–7393. AAAI press.

A Summarization Framework Implementation

As shown in the figure below, the ToS summarization framework has been implemented as a Web application, with the extractive and abstractive components of the ToS hybrid model interacting with a Web component.

The framework consists of four major components:

ToS Web Scraper: The Web scraper tool has been developed to address the lack of reference

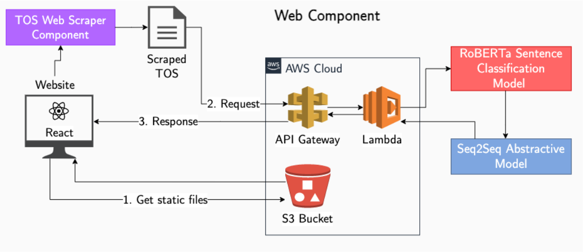


Figure 8: An overview of how the ToS Hybrid Model (the RoBERTA Classifier and Seq2Seq model) interact with the web component. The ToS Hybrid Model is called by the Lambda functions and returns a string of ordered, summarised sentences in the response.

datasets of ToS documents. It is also a feature of the website, allowing users to retrieve a terms document from any website or company, by either entering the source URL or searching by the company name. It accepts an URL as input, which is parsed to form a valid URL. If the given URL does not directly link to a terms document, the scraper first tries to search for potential URLs linking to T&C, through regular expression (regex) for HTML link elements containing words and phrases associated with T&C, e.g. "privacy policy", "terms", and "legal". The HTML of the URL is retrieved using the Selenium[11] Python library and parsed using BeautifulSoup[12]. Content cleaning steps include removing HTML tags for the navigation bar, footer, headings, images and labels. It is common for terms documents to contain a list of terms with a sentence heading such as “You agree to:" or “You agree not to:". This can be problematic when separating sentences, as the distinction between “agree to" and “agree not to" is quite important, not only for identifying risky terms but also unimportant terms for the extractive component. To fix this issue, the HTML structure of the page is utilized. By identifying list elements that come after a text ending with a semicolon, the scraper prefixes each list item with the text preceding the semicolon.This allows the extractive and abstractive models to identify the context surrounding each list item and whether they have a positive or negative meaning. Additional tag cleaning includes the removal of implicit headings - this refers to headings that are not HTML tags but still titles for various sections of the document. These headings are removed using regex for commonly occurring title structures.

Web Component: User interaction with the framework is enabled through the Web component which consists of a website and accompanying Chrome browser plugin, with an Amazon Web Services (AWS) back-end component. A plugin allows quick look-up of a summary of terms when a user is already on a T&C page. Chrome was chosen for the plugin implementation due to it being the leading Internet browser (64% global market share[13]). It chains the outputs of the extractive component to the inputs to the abstractive component in a hybrid network architecture. The website is accessed through the S3 bucket static files. Functionality is shared between the website and the plugin through API routes (using the API Gateway service) connecting to two Lambda functions: one for the web-scraping component to be accessible to the website, and one for summarization (encompassing the extractive and abstractive components), which is used by both the website and plugin.

Extractive Component: following tag cleaning, the classification model is loaded and the sentences are vectorized. The Labelled Dataset is split into 60:30:10 test:validation:development sets, with the development set determining the heuristics for the labelling functions definitions. The labelling functions assign $[0|1|-1]$ labels to each of these sentences. After these labels are fed into a LabelModel, the sentences and assigned labels are ready to be used as training data for the classifier. Sentences with probabilities $>$50% for the important class are filtered and re-ordered, with the most important sentences at the top. Stopwords and single-character words are removed. The validation set is used for training the RoBERTa classifier and the test set is used for evaluating the label model, classifier and baseline models during the Evaluation.

Abstractive Component: the encoder, decoder and summary tokeniser are loaded at initialization. The tokeniser converts the texts to sequences and pads them up to the maximum length, 60, with the encoder making predictions for each sequence. The `START` token is used as a first input to the decoder, which predicts the next words until an `END` token is generated or the maximum length has been reached. The tokenised sequence is returned as a readable format and the final summary is joined by newlines.

[11] https://pypi.org/project/selenium/

[12] https://pypi.org/project/beautifulsoup4/

[13] https://gs.statcounter.com/