future internet

MDPI

Article

Automatic Detection of Sensitive Data Using Transformer-Based Classifiers

Michael Petrolini , Stefano Cagnoni * and Monica Mordonini

Department of Engineering and Architecture, University of Parma, Parco Area delle Scienze 181a, 43124 Parma, Italy; michael.petrolini@studenti.unipr.it (M.P.); monica.mordonini@unipr.it (M.M.)
* Correspondence: stefano.cagnoni@unipr.it

Abstract: The General Data Protection Regulation (GDPR) has allowed EU citizens and residents to have more control over their personal data, simplifying the regulatory environment affecting international business and unifying and homogenising privacy legislation within the EU. This regulation affects all companies that process data of European residents regardless of the place in which they are processed and their registered office, providing for a strict discipline of data protection. These companies must comply with the GDPR and be aware of the content of the data they manage; this is especially important if they are holding sensitive data, that is, any information regarding racial or ethnic origin, political opinions, religious or philosophical beliefs, trade union membership, data relating to the sexual life or sexual orientation of the person, as well as data on physical and mental health. These classes of data are hardly structured, and most frequently they appear within a document such as an email message, a review or a post. It is extremely difficult to know if a company is in possession of sensitive data at the risk of not protecting them properly. The goal of the study described in this paper is to use Machine Learning, in particular the Transformer deep-learning model, to develop classifiers capable of detecting documents that are likely to include sensitive data. Additionally, we want the classifiers to recognize the particular type of sensitive topic with which they deal, in order for a company to have a better knowledge of the data they own. We expect to make the model described in this paper available as a web service, customized to private data of possible customers, or even in a free-to-use version based on the freely available data set we have built to train the classifiers.

Keywords: GDPR; sensitive data; personal data; natural language processing; BERT; transformers

check for updates

Citation: Petrolini, M.; Cagnoni, S.; Mordonini, M. Automatic Detection of Sensitive Data Using Transformer-Based Classifiers. *Future Internet* **2022**, *14*, 228. https://doi.org/10.3390/fi14080228

Academic Editors: Vijayakumar Varadarajan, Rajanikanth Aluvalu and Ketan Kotecha

Received: 4 July 2022
Accepted: 25 July 2022
Published: 27 July 2022

1. Introduction

Almost every interaction one has on the web can be used by companies to organize marketing operations and to develop commercial strategies for selling their products. To protect the European users' privacy, the EU has adopted the General Data Protection Regulation (GDPR), which unifies privacy legislation across all European members. This regulation affects all companies that process data of European residents regardless of the place in which they are processed and provides for a strict data protection discipline, with severe penalties that can reach 4% of a company's global turnover. Because of this, it is extremely important that these companies comply with the GDPR and are therefore aware of the content of the data they handle. However, this goal is not always easy to achieve: not all companies have adopted measures to check and manage their data, causing loss of knowledge about their content and ownership. A further matter of concern regards the personal data classes recognized within this regulation. In particular:

- Personal data is all information relating to an identified or identifiable natural person, with particular reference to an identifier such as her/his name, identification number, location data, IP address, date and/or place of birth, or online identifier. These data

items are usually relatively simple to retrieve using dictionaries or regular expressions, considering that they usually appear within well-established and easy-to-read data structures.
- Sensitive data, on the other hand, include personal data such as racial or ethnic origin, political opinions, religious or philosophical beliefs, trade union membership, data relating to one's sexual life or sexual orientation, as well as data on present and future physical and mental health. This includes information on healthcare services regardless of the source, for example a doctor. These data types are hardly structured, and most frequently they are part of a document such as an email message, a review or a post. This makes it extremely difficult to know if a company is in possession of sensitive data, with the risk that they will not be properly protected.

The goal of this project is to develop a model capable of automatically detecting whether a document contains sensitive data, as well as the specific type of sensitive topic contained therein: politics, religion, health or sexual habits.

2. General Data Protection Regulation

The General Data Protection Regulation [1] is a European Union regulation regarding privacy and data protection, whose goal is to secure personal data of citizens and residents in the EU, both inside and outside its borders. This regulation applies to companies, businesses and organizations, inside and outside the EU that process personal data of EU residents. The GDPR provides a common base of rules shared by all EU countries, making it easier for companies to comply with its standards; this simplification has come at the price of a stricter data protection, with sanctions amounting up to 4% of a company's global turnover.

According to the European Commission, personal data consist of any information related to an individual in private, professional or public ambit. This information can exist in different forms: text, images, videos, audios, etc. This new regulation expands and characterizes the data definitions in the old directives, as well as adds new ones.

The GDPR defines three types of data that must be protected:

- **Personal data:** in the GDPR, article 4, paragraph 1:

  > *'personal data' means any information relating to an identified or identifiable natural person ('data subject'); an identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person;*

  Therefore, personal data involve any information that can lead to the identification of a natural person. The limits of this definition are still subject to debate: according to it, even a generic physical description of an individual, if too specific, could be treated as personal data. Direct information is the most common type of personal data, and it is easier to detect than its indirect counterpart.
- **Sensitive data:** a special category of personal data that requires strict protection, as stated in the GDPR, article 9, paragraph 1:

  > *Processing of personal data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, or trade union membership, and the processing of genetic data, biometric data for the purpose of uniquely identifying a natural person, data concerning health or data concerning a natural person's sex life or sexual orientation shall be prohibited.*

  The definition of sensitive data has been intentionally kept generic to include a broader group of information types.
- **Personal data relating to criminal convictions and offences:** the processing of this personal data type must be kept under control of an official authority or authorised by

Union or Member State laws providing for appropriate safeguards for the rights and freedoms of data subjects.

In this work, we are going to focus only on personal and sensitive data, since they are the most common data types saved in companies' databases.

3. Background

3.1. Sensitive Data Discovery

The implications of the GDPR are a topic of great relevance in many fields [2–4] because its restrictions affect the way data are collected, stored and processed. Sensitive Data Discovery is the process of locating and identifying specific types of personal data from unstructured and structured data sources to be able to protect or remove them according to the regulations' requirements. This is a crucial step for security teams to be GDPR-ready, to ensure the privacy of their organizations' customers and employees and to prevent data breaches and leaks. Since new data are created on a daily basis, data discovery is an ongoing endeavor that security professionals must proactively pursue to build a strong and secure foundation. Sensitive data discovery becomes essential for heavily-audited organizations, such as healthcare, financial services, telecommunication companies and government offices. These organizations adhere to strict sensitive data discovery protocols and follow specific industry-based rules and regulations.

Sensitive Data Discovery is a topic of great relevance that has been discussed in some publications; in [5], the authors describe two different algorithms for discovering the most frequent patterns in a data set of sensitive records: one is based on the *exponential mechanism* of McSherry and Talwar [6], while the other is based on the analysis of the established technique of adding Laplace noise to released statistics to preserve global statistics while protecting individual pieces of information [7]. In an extensive review paper [8], all the methods taken into consideration are divided into two main categories: rule-based approaches (lookup tables, regular expressions, metadata analysis) and machine learning methods. Considering the latter, most sensitive-topic and data-detection systems are based on deep learning or other computational intelligence methods. One of the main reasons for the prevalence of neural-network-based approaches lies in the sub-symbolic distribution of knowledge over thousands of weights that make neural networks generally robust to small input changes, weight alterations or disconnections. This is certainly one of the key properties that allow neural networks to perform well in the presence of noisy or uncertain data. Related approaches [9,10] use fuzzy logic methods [11], mainly for hiding the correlations between sensitive data. This approach is particularly relevant when dealing with medical records coming from different centers, where direct references between records should be avoided to preserve privacy.

With respect to our project's goals and a possible significant comparison of the methods they describe with ours, most papers are either outdated or deal with definitions of sensitive data that differ from those given in the GDPR.

A few commercial tools are available for detecting sensitive data, for example, (IBM Security Guardium [12], Azure Information Protection [13] and Rubrik [14], etc.). Unfortunately, all these applications are proprietary software, so it is not possible to study their approach to the problem in depth or to adapt them to different regulations.

3.2. Transformers

Transformers are a recent deep learning paradigm [15] that have become very popular in several research and application fields, such as Computer Vision (CV), natural language processing (NLP) and speech processing. Recent works show that Transformer-based pre-trained models (PTMs) [16] can achieve state-of-the-art performances on various tasks, especially in NLP.

The basic Transformer architecture is a sequence-to-sequence mapping model that includes an encoder and a decoder, both consisting of a stack of L identical blocks. Each encoder block, in turn, consists of a multi-head self-attention module and a position-wise feed-forward network (FFN). Decoder blocks additionally insert cross-attention modules between the multi-head self-attention modules and the position-wise FFNs. Furthermore, the self-attention modules in the decoder are adapted to prevent one position from attending to subsequent positions in a non-causal way. The overall vanilla Transformer architecture is shown in Figure 1.

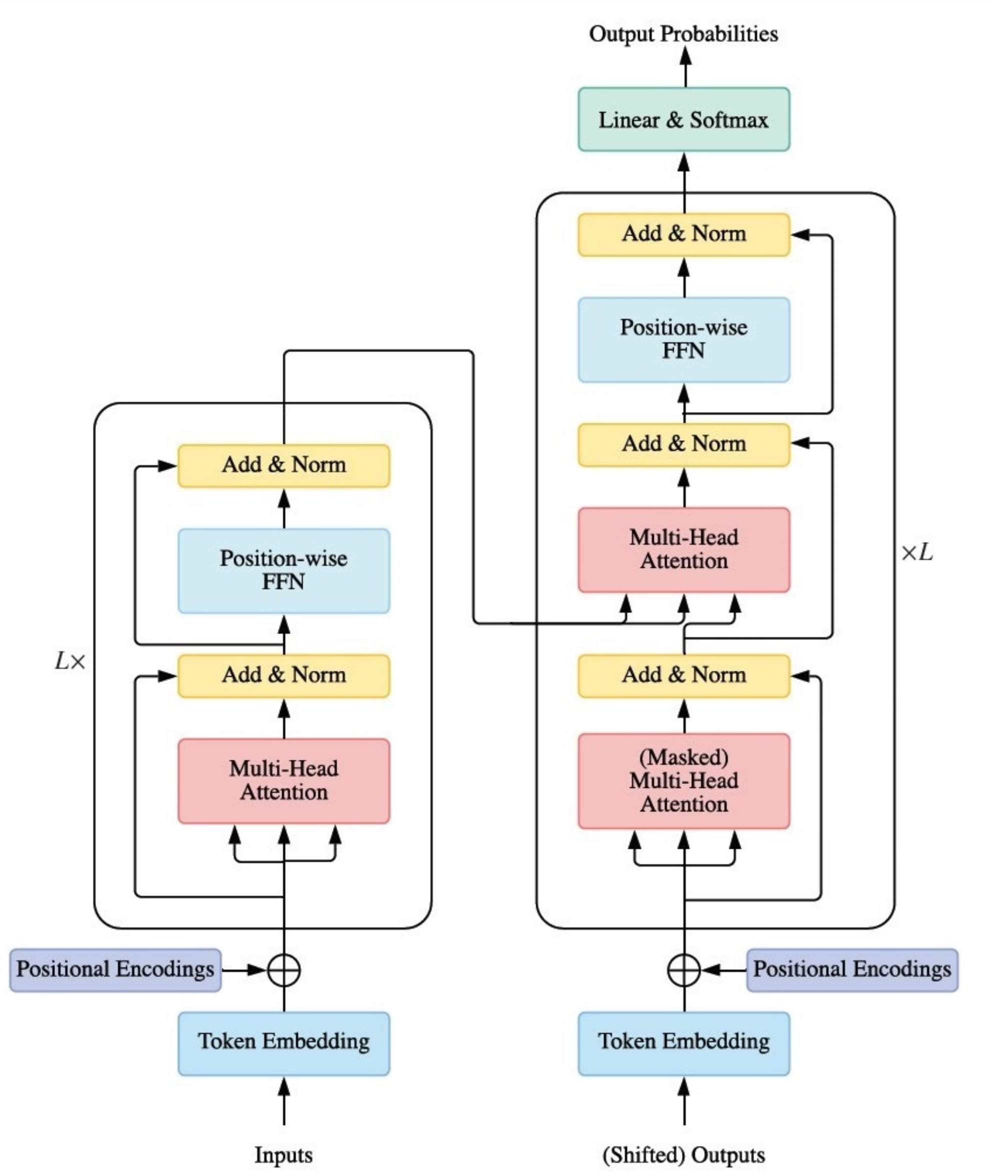

Figure 1. Vanilla Transformer architecture (adapted from [15]). The encoder, on the left, and the decoder, on the right, both consist of L identical blocks.

Transformers adopt an attention mechanism based on the Query-Key-Value (QKV) model. Given the packed matrix representations of:

- The matrix of *queries* $\mathbf{Q} \in \mathbb{R}^{NxD_k}$, which contains the vector representation of a word within a sentence;
- The matrix of *keys* $\mathbf{K} \in \mathbb{R}^{MxD_k}$, which contains the vector representations of all words within a sentence;
- The matrix of *values* $\mathbf{V} \in \mathbb{R}^{MxD_v}$ that are related to **K** and, just like **K**, are vector representations of all words within a sentence.

The scaled-dot-product attention function used by the Transformer is given by

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{D_k}}\right)\mathbf{V} = \mathbf{A}\mathbf{V} \tag{1}$$

where N and M denote the lengths of queries and keys (or values) and D_k and D_v denote the dimensions of keys (or queries) and values, respectively. $\mathbf{A} = softmax\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{D_k}}\right)$ is often called the *attention mask*; softmax is applied in a row-wise manner. The dot-products of queries and keys are divided by $\sqrt{D_k}$ to alleviate the vanishing-gradient problem affecting the softmax function.

Instead of simply applying a single attention function, Transformers use multi-head attention, where the D_m-dimensional original queries, keys and values are projected onto D_k, D_k and D_v dimensions, respectively, learning H (preset number of heads) different sets of projections. For each of the projected queries, keys and values, an attention output is computed according to Equation (1). The model then concatenates all the outputs and projects them back to a D_m-dimensional representation:

$$MultiHeadAttn(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = Concat(head_1, \cdots, head_H)\mathbf{W}^O \tag{2}$$

$$where\ head_i = Attention(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V)$$

where $W_i^Q \in \mathbb{R}^{D_m \times D_k}$, $W_i^K \in \mathbb{R}^{D_m \times D_k}$, $W_i^V \in \mathbb{R}^{D_m \times D_v}$ and $W^O \in \mathbb{R}^{HD_v \times D_m}$. In a Transformer, there are three types of attention in terms of query sources and key-value pairs:

- *Self-attention*. In the Transformer encoder, we set **Q** = **K** = **V** = **X** in Equation (2), where **X** is the output of the previous layer.
- *Masked Self-attention*. In the Transformer decoder, self-attention is restricted so that queries at each position can only attend to all key-value pairs up to and including that position to enforce causality. To enable parallel training, this is typically obtained by applying a mask function to the non-normalized attention $\hat{\mathbf{A}} = exp(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{D_k}})$, where the positions that are not to be accessed are masked out by setting $\hat{A}_{ij} = -\infty$ if $i < j$. This kind of self-attention is often referred to as autoregressive or causal attention.
- *Cross-attention*. The queries are projected from the outputs of the previous decoder layer, whereas the keys and values are projected using the outputs of the encoder.

Regarding the position-wise FFN, it is a fully connected feed-forward module that operates separately and identically on each position

$$FFN(\mathbf{H'}) = ReLU(\mathbf{H'}\mathbf{W}^1 + \mathbf{b}^1)\mathbf{W}^2 + \mathbf{b}^2$$

where $\mathbf{H'}$ is the output of the previous layer and $\mathbf{W}^1 \in \mathbb{R}^{D_m \times D_f}$, $\mathbf{W}^2 \in \mathbb{R}^{D_f \times D_m}$, $\mathbf{b}^1 \in \mathbb{R}^{D_f}$, $\mathbf{b}^2 \in \mathbb{R}^{D_m}$ are trainable parameters. Typically, the intermediate dimension D_f of the FFN is set to be larger than D_m.

To build a deep model, a Transformer applies a residual connection [17] to each module, followed by Layer Normalization. For instance, each Transformer encoder block can be written as

$$\mathbf{H'} = LayerNorm(SelfAttention(\mathbf{X}) + \mathbf{X})$$

$$\mathbf{H} = LayerNorm(FFN(\mathbf{H'}) + \mathbf{H'})$$

where $SelfAttention(.)$ denotes the self-attention module and $LayerNorm(FFN(.))$ denotes the layer normalization operation.

3.3. BERT

BERT [18] is an advanced word-embedding model that uses a bidirectional Transformer [19] to pre-train a language model on a large corpus and allow for fine-tuning the pre-trained model on other tasks. As a sentence encoder, it can accurately produce the context representation of a sentence [20]. BERT is a deep bidirectional system that is capable of handling unlabeled text by jointly taking into consideration both the left and right context of a sequence in all layers. A deep bidirectional model is more powerful than a shallow left-to-right and right-to-left model. Given that BERT's goal is to generate a language model without generating a prediction, it uses only the encoder part of the Transformer architecture. The Transformer encoder can read the entire input sequence at once; thus it is considered bidirectional. This peculiarity enables the model to learn the context of the element of a sequence based on its full neighborhood. To overcome the problems affecting previous unidirectional models, such as RNN [21] or n-gram language models [22], BERT takes advantage of both these training strategies, as shown in Figure 2:

- BERT removes the unidirectional constraint by performing a *Mask Language Model (MLM)* task, which randomly masks some of the tokens from the input and tries to reconstruct (predict) the full original input. Unlike left-to-right language model pre-training, the objective of MLM enables the word representation to fuse the left and the right context, allowing one to pre-train a deep bidirectional Transformer. MLM is the key feature of BERT that has allowed it to outperform previous embedding methods.
- Many important tasks, such as Question Answering (QA) and Natural Language Inference (NLI), involve understanding the relationship between two sentences, denoted as ⟨Sentence A, Sentence B⟩, which is not directly captured by language modeling. To train a model that understands sentence relationships, BERT is pre-trained on a binarized *Next Sentence Prediction (NSP)* task: when sentence pairs ⟨Sentence$A$, Sentence$B$⟩ are selected as pre-training examples, half of the time, B is the actual sentence following A, while, in the other cases, B is a random sentence taken from the sentence corpus. The model must then predict for each ⟨Sentence A, Sentence B⟩ pair whether Sentence B is the actual sentence following Sentence A.

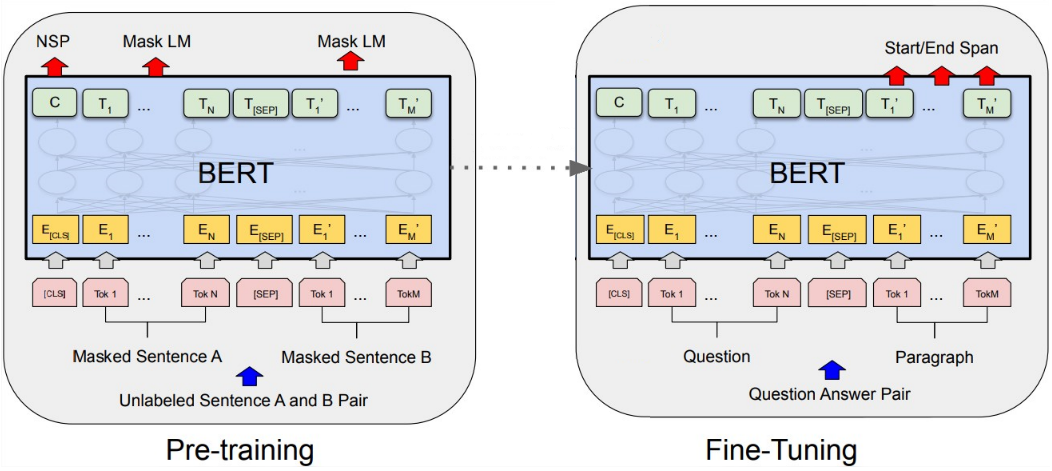

Figure 2. Overall pre-training and fine-tuning procedures for BERT. Except for the output layers, the same architecture is used in both pre-training and fine-tuning. The same pre-trained model parameters are used to initialize models for different downstream tasks. During fine-tuning, all parameters are tuned. [CLS] is a special symbol added in front of every input example, and [SEP] is a special separator token, e.g., separating questions and answers. (Adapted from [18]).

The pre-training procedure has been based on the BookCorpus (800 M words) [23] and English Wikipedia. The model thus pre-trained can then be used in many NLP tasks, applying a relatively inexpensive (if compared to the pre-training phase) fine-tuning process. Fine-tuning is pretty straightforward: for each task, the task-specific inputs and outputs are plugged into BERT, and all the transformer parameters are then fine-tuned end-to-end.

Two BERT models are available, differing in size:

- $BERT_{base}$: it features 12 blocks of Transformers, a hidden layer with 768 neurons, 12 self-attention heads for a total size of the pre-trained model amounting to 110 million parameters.
- $BERT_{large}$: it features 24 blocks of Transformers, a hidden layer with 1024 neurons, 16 self-attention heads for a total size of the pre-trained model amounting to 340 million parameters.

The $BERT_{large}$ model requires much more memory than $BERT_{base}$. As a result, running $BERT_{large}$ on an average GPU architecture would require using so small a batch size that it would negatively affect model accuracy, regardless of the other learning parameters. Therefore, we used $BERT_{base}$ as our base model.

4. Problem Formulation

When we started working on this project, the first problem we had to face was to obtain a data set of sensitive data with enough samples to train and test our models. Unfortunately, at the moment, there are no public and widely accepted data sets of sensitive data available. In fact, all the Sensitive Data Discovery solutions to this day are based on private data sets that are not freely accessible. This is fully understandable since, given the nature of this type of information, it would be illegal to publish it. To overcome this obvious limitation, we decided to divide the problem into two separate tasks that would allow us to operate on data lacking any illegal reference. The following is an example of a sentence that contains sensitive data:

I've heard that John Doe had a heart attack last week and now he's at the hospital.

We can see that, to embed sensitive data, a sentence must contain two elements:

- A sensitive topic, that is one of the topics described in the definition of sensitive data.
- Personal data that allow one to link the sensitive topic to a natural person.

Detecting personal data is relatively easy, since most of them can be retrieved using searches based on regular expressions (email/IP addresses, social security number, etc.) or dictionaries (names, surnames, etc.). We can then reduce the problem of detecting sensitive data to the problem of detecting sensitive topics. In the example above, the same sentence contains both personal data and a sensitive topic, but that is not always the case: in a long document, we could have information related to a natural person in sentences other than the ones containing sensitive topics or maybe not even in any sentence at all (for instance, it may be represented by the sender/recipient of an email). For this reason, it is important to locate personal data within the whole document and examine whether the above conditions are satisfied at the document level. This induces the risk of generating false positives. In our case, our project's goal was to generate a warning whenever a document might contain sensitive data, which means we needed to bring the number of false negatives as close to zero as possible. In this way, we were also able to collect sentences regarding our topics of interest without violating the GDPR, because they would not reveal the identity of the natural person involved.

5. Data Sets

We considered four main classes of sensitive topics:

- *Politics:* any argument related to the racial or ethnic origin of an individual, along with any political opinion or trade union membership.
- *Health:* any information related to physical and/or mental health.
- *Religion:* any topic related to religious or philosophical beliefs.
- *Sexuality:* any argument involving the sexual life or orientation of an individual.

We used *Reddit* (http://reddit.com, accessed on 27 July 2022) as the data source for the project because its hierarchical *subreddit*-based structure makes it easy to detect specific topics both directly and through related or derived sub-topics. Table 1 lists the main subreddits about topics related to the four main classes of sensitive topics we took into consideration. For each subreddit, we then scraped their "hottest" posts, along with the related comments, obtaining a set of more than 20,000 biased elements having high probability of being related to sensitive topics, equally distributed over the latter.

Table 1. Grouping of sensitive topics in the relative macro-topics and list of the subreddits from which the posts were retrieved.

	Politics	**Health**	**Religion**	**Sexuality**
Sensitive topics	*Ethnic origin* *Political beliefs* *Trade union membership*	*Genetic data* *Biometric data* *Health state*	*Religious beliefs* *Philosophical beliefs*	*Sexual life* *Sexual orientation*
Subreddit	*r/politics* *r/Libertarian* *r/ukpolitics* *r/Ethnicity* *r/union* *r/LabourUK* *r/socialism* *r/Conservative* *r/Labour* *r/Anarchism* *r/communism* *r/antiwork* *r/ConservativesOnly* *r/democrats* *r/DemocraticSocialism* *r/PoliticalCompass* *r/Republican* *r/PoliticsPeopleTwitter*	*r/Health* *r/healthcare* *r/mentalhealth* *r/medicalschool* *r/medicine* *r/Doctor* *r/biology* *r/Coronavirus* *r/nursing*	*r/religion* *r/Christianity* *r/Christian* *r/TrueChristian* *r/atheism* *r/islam* *r/philosophy* *r/Objectivism* *r/Buddhism* *r/askphilosophy* *r/PhilosophyofReligion* *r/Judaism* *r/Catholicism* *r/hinduism* *r/Izlam* *r/exmormon* *r/exmuslim*	*r/lgbt* *r/gay* *r/lesbian* *r/bisexual* *r/asktransgender* *r/transgender* *r/askgaybros* *r/actuallesbians* *r/ainbow* *r/LesbianActually* *r/gaybros* *r/LGBTeens* *r/queer* *r/sexuality* *r/sex* *r/relationships* *r/ldssexuality* *r/asexuality*

To obtain a set of "neutral" sentences, i.e., unrelated to the sensitive topics considered, we scraped the hottest posts from the whole Reddit space, filtering out any post from the sensitive subreddits, obtaining a data set of almost 7000 elements. Therefore, the final data set contained almost 30,000 records.

Next, we tokenized the collected sentences, removing, at the same time, all sentences with less than 10 words or 100 characters because they are too short to clearly express a topic and those with more than 50 words and 200 characters because they are likely to deal with more than one topic.

5.1. Training Set

To generate our training data set, we sampled a subset of the sentences and had it classified through the Mechanical Turk crowd-sourcing platform: sentence labeling was performed by a group of 161 English-speaking workers with a HIT approval rating ≥95%, ensuring high quality annotations. To maximize the data set quality, we finally performed a quality check on a small sample of each worker's annotations, rejecting the entire set of annotations in case we estimated an error rate ≥20% on the analyzed sample. In the annotation task, after an introductory explanation about the subject, we asked each worker to determine if a sentence was referring to one of the four available categories of sensitive topics or if it dealt with a general topic. We opted for creating a data set including instances labeled as belonging to a single class. This has had beneficial effects, making the manual labeling of the instances easier and possibly less controversial. As we show in Section 7, this choice has not compromised the system's capacity of identifying multiple topics.

The resulting data set contains almost 48,000 samples, including generic and sensitive topics; the sensitive topics are then further divided into the four main classes of arguments, as shown in Table 2.

Table 2. Training data set composition.

	Sensitive Topics				Generic Topics	Total
	Politics	Religion	Health	Sexuality		
Samples	6767	6073	6673	4048	23,978	
Total	23,561				23,978	47,539

5.2. Test Set

To test the effectiveness of the machine learning models on the task, we also generated a test data set selected from the sentences that had not been labeled yet. A team of three users experienced on the GDPR regulation determined which of those sentences effectively held potentially sensitive topics and in which macro-class they would best fit. For each sample, we used a majority vote policy to determine its class. The resulting test set contains 2400 items distributed as shown in Table 3.

Table 3. Test data set composition.

	Sensitive Topics				Generic Topics	Total
	Politics	Religion	Health	Sexuality		
Samples	391	337	253	219	1200	
Total	1200				1200	2400

6. Model Architectures

We compared two model architectures, both based on BERT, which had been shown to be effective in NLP problems, in particular for sentiment analysis [24,25]. For the project implementation, we used the *BertModel* provided by the package *transformers*, from the AI community *Hugging Face* (http://huggingface.co, accessed on 3 July 2022), with the corresponding pre-trained *bert-base-cased* weights.

6.1. Flat Multi-Label Model

The first model we considered is a single-classifier multi-label architecture (Figure 3, left) in which each of the four sensitive topics is represented by an independent output that can be interpreted as the likelihood with which the input can be associated to the corresponding label. With this approach, we determine at the same time whether the sentence contains sensitive topics and, in that case, which is its main topic. If the model is not able to associate any label to the input sentence, i.e., if the likelihood is low for all classes taken into consideration, this implies that it does not contain any sensitive topic.

This model generates a quick response since it is composed of a single classification module. As well, it can be applied with no modifications to the identification of multiple sensitive topics in the same sentence, such as political discussions about health assistance or religious opinions about sexuality.

6.2. Hierarchical Model

We also considered a hierarchical architecture (Figure 3, right), including two distinct classifiers:

- A binary classifier that determines whether a sentence contains sensitive topics.
- A multi-class classifier, activated only when the binary classifier detects a sensitive topic, that determines to which sensitive macro-category the sentence belongs.

Following this approach, the two models tackle simpler specific tasks, aiming at an increased accuracy of the final system, with the downside represented by the need to train two classifiers, which therefore increases both the training and the response time.

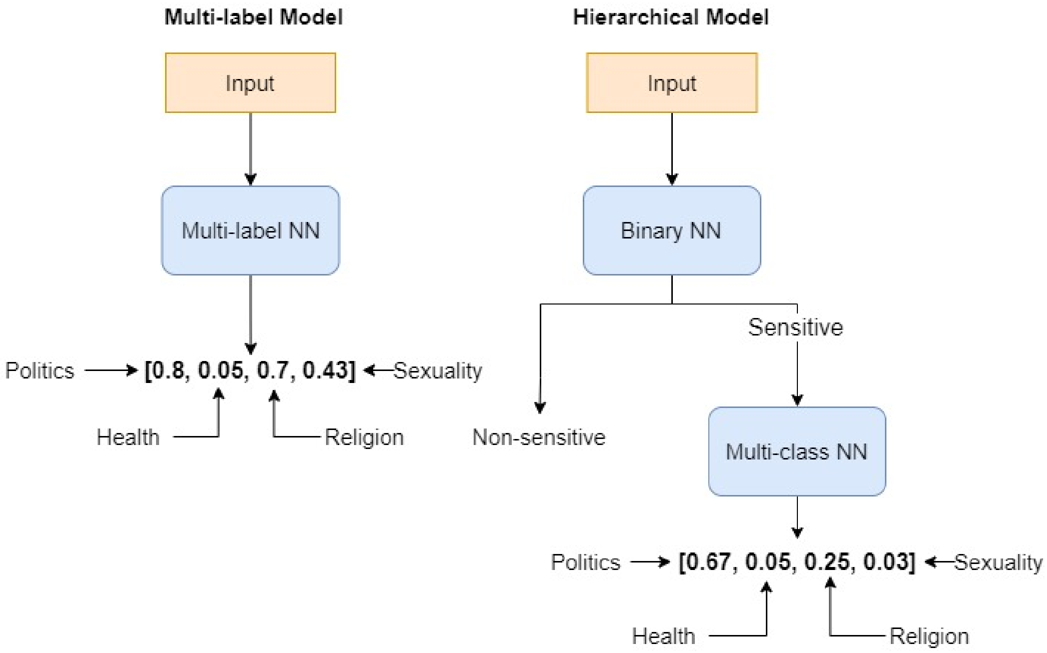

Figure 3. Overview of the two proposed model architectures.

6.3. Models' Structure

The NNs we developed, one for the flat multi-label architecture and two for the hierarchical one, share the same structure:

- A pre-trained $BERT_{base}$ model that is fine-tuned over the training data set.
- A Dropout layer to prevent the BERT model from overfitting the training data.
- A Linear layer.

The main difference between the models, apart from a subset of the training hyperparameters that will be discussed later, is the activation function of the last Linear layer:

- The multi-label NN uses a sigmoid function, which is defined as

$$\text{Sigmoid}(x) = \sigma(x) = \frac{1}{1+e^{-x}}$$

  The sigmoid function is suited for multi-label models because the resulting outputs are independent of one another, thus allowing for the assignment of multiple labels (or no label at all) to the input data.

- The binary and the multi-class models both use a Softmax function, which is defined as

$$\text{Softmax}(x)_i = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}}$$

  where x is a vector of K real numbers. The Softmax function returns a probability distribution, which means that the outputs produced using this function are interrelated and add up to one, satisfying the formal requirements for representing probability distributions.

7. Training and Evaluation

To find the best hyperparameters combination, we performed a grid search for each neural network, resulting in the setup outlined in Table 4.

Table 4. Best hyperparameters combination for each neural network.

		Optimizer	Batch Size	Learning Rate	Dropout
Hierarchical	*Binary*	RMSprop	32	1×10^{-5}	0.3
	Multi-class	Adam	32	1×10^{-5}	0.5
Flat Multi-label		RMSprop	16	1×10^{-5}	0.3

We trained all models using the data set generated as described in Section 5.1 for two epochs. This number of iterations may seem very small, but BERT developers themselves suggest fine-tuning it for a number of epochs between two and four, since otherwise the probability of overfitting the model [18] is very high. Training each model has taken roughly 4 h using an Intel Core i5 8250U CPU with 12 GB of memory.

To account for class imbalance, we assess the classifier performance using Precision, Recall and F_1-score:

$$F_1 = 2 \times (precision \times recall)/(precision + recall)$$

The computational effort required to produce an output is also an important metric for models incorporated in real-time systems such as a web service. Therefore, we also measured some statistics about the time, in seconds, needed to generate a prediction for a single sample.

8. Results

To evaluate the effectiveness of the two architectures, we compare them on both the sensitive topic detection task and in its classification into one of the four macro-categories. To do so, we use the test data set obtained as described in Section 5.2.

The flat multi-label model is based on a single neural network whose output consists of a vector of four elements, each representing a coefficient that expresses how likely the input sentence is to belong to one of the macro-categories of sensitive topics; in case one or more of the outputs exceed 0.5, the related label is assigned to the input sentence. Using this approach, if no labels can be associated to the input sentence, then the sentence topic is classified as non-sensitive.

We first compared both models on the task of sensitive-topic detection, treating the union of the four macro-categories as the *Sensitive* class. Regarding the flat multi-label model, we considered each prediction with at least one output above the threshold as the detection of a sensitive topic. Table 5 shows that both models reached similar results, with the best F_1-score of 0.95 obtained by the binary neural network of the hierarchical model.

Table 5. Comparison between the binary neural network of the hierarchical model and the flat multi-label model.

	Precision	Recall	F_1-Score	Runtime (s)
Binary	0.97	0.93	0.95	0.11 (±0.01)
Flat Multi-label	0.96	0.92	0.94	0.12 (±0.01)

Next, we compared the two models on the sensitive topic recognition task, using only the samples of the test set not labeled as *Other*. The output of the flat multi-label model has been considered correct if the output corresponding to the correct prediction is greater than 0.5, thus allowing the association of the input with multiple labels at one time. The results in Table 6 show that, while the multi-class neural network of the hierarchical model has been able to obtain almost the same performance as its binary counterpart even if the task is obviously harder, the flat multi-label one obtained a lower F_1-score in this task: the recall value of the flat multi-label model is lower than the recall of the flat multi-class model of the hierarchical classifier on every macro-category. This might depend on the class unbalance; however, the hierarchical model seems to have better overcome this problem, possibly because it has been trained only on sensitive data according to the two-stage strategy of the hierarchical classifier.

Table 6. Comparison between the multi-class neural network of the hierarchical model and the flat multi-label model.

	Precision	Recall	F_1-Score	Runtime (s)
Multi-class model	0.94	0.94	0.94	0.11 (±0.01)
Health	0.91	0.95	0.93	
Politics	0.96	0.92	0.94	
Religion	0.94	0.96	0.95	
Sexuality	0.94	0.93	0.93	
Multi-label model	0.95	0.87	0.91	0.12 (±0.01)
Health	0.92	0.87	0.89	
Politics	1.00	0.81	0.90	
Religion	0.94	0.93	0.93	
Sexuality	0.94	0.89	0.91	

An important detail to be noticed is that out of 2400 sentences, only four have been labeled with two or more labels, and in all cases, one was the correct label, as shown in Table 7. This means that in the comparison the selectivity of the output reduces the advantage for the flat classifier to generate possible multi-class outputs almost to zero (Notice that in computing the F_1 score, producing more than one output favors an increase of both precision and recall).

Table 7. Text and prediction of the four sentences associated with more than one label by the flat classifier.

Text	Actual Class	Predicted Labels
I think they allow medical intervention in life and death cases, but I know someone who is a Christian Scientist and is not getting vaccinated.	Health	Health, Religion
Joe Biden literally said on national TV he would not trust a vaccine developed under the Trump administration.	Politics	Health, Politics
I wish people's superstitious beliefs were not so often involved in politics, and I wish politicians did not feel the need to declare or pretend they are religious to gain support.	Religion	Politics, Religion
My family is not open to the idea of me being trans, but I already take antidepressants, so they aren not against me taking medication.	Sexuality	Health, Sexuality

Finally, we analyze both models on the complete task: detecting if a sentence contains sensitive topics and choosing the appropriate macro-category. We do so on the entire test set and according to the same result assessment policies applied in the previous comparisons. The results in Table 8 confirm that the hierarchical model can reach a slightly better F_1-score than the flat multi-label classifier, causing a negligible increase in the average decision time per sentence.

In the comparison between the two models on the sensitive sentence detection, the sensitive class selection and the complete classification tasks, we have shown that the hierarchical classifier can reach a slightly higher accuracy than the flat multi-label classifier. In the end, neither model clearly outperforms the other: if one needs to maximize prediction accuracy, the hierarchical model appears to be preferable even if there is not such a big difference between the two. In fact, if one is only interested in detecting sensitive topics and not in their categorization, such a difference becomes negligible.

Table 8. Comparison between the hierarchical and the flat multi-label model on the complete detection and classification task.

	Precision	Recall	F_1-Score	Runtime (s)
Hierarchical model	0.92	0.93	0.92	0.13 (±0.01)
Other	0.93	0.97	0.95	
Health	0.89	0.82	0.85	
Politics	0.93	0.86	0.89	
Religion	0.94	0.95	0.94	
Sexuality	0.91	0.86	0.89	
Multi-label model	0.91	0.91	0.91	0.12 (±0.01)
Other	0.92	0.96	0.94	
Health	0.87	0.87	0.87	
Politics	0.94	0.81	0.87	
Religion	0.92	0.93	0.93	
Sexuality	0.90	0.89	0.89	

On the other hand, considering computation cost, it is important to consider that the hierarchical classifier uses two neural networks to produce a response and, therefore, requires a slightly longer time than the flat multi-label one, which includes just one NN. In a corporate environment where the amount of data can easily become massive and where it is often required to have a quick response time, even this slight difference may become significant in deciding which model to choose.

9. Conclusions

Detecting sensitive data is critical for companies who handle user-related data. Machine Learning models can help in this task, reducing the risk of neglecting such data while highly increasing the amount of documents that can be analyzed in a reasonable amount of time. With this project, we have shown that recent machine learning approaches, even available as free off-the-shelf tools such as BERT, have a very high potential in this field. Even if our work represents only a small step towards a final solution to the sensitive data detection and classification problem, we hope that it can help stimulate future discussion on the topic. In particular, the data set we have developed as a by-product of this project could become the most relevant part of our study if its free availability as a benchmark will aid future research on sensitive data detection and classification, providing a common ground for a fair result comparison of applications to these fields.

Looking ahead, this project has highlighted many future possible research extensions because sensitive data protection will keep being an extremely important topic inside and outside the EU borders, and companies will more and more need to have knowledge about the ever-increasing amount of data they manage.

In particular:

- In this project, we have applied a policy which determines that a document contains sensitive data if it meets both the following requirements:
  - It contains at least a sentence classified as containing a sensitive topic.
  - It refers to personal data, that is, any information that can point to a natural person.

  According to our policy, these two elements do not necessarily need to be related with each other to consider the document sensitive. This made us focus mainly on the topic detection and classification task.

  This approach also allowed us to minimize false negatives, which is also a crucial requirement for the relevance of the negative effects such cases can have. Of course, it is not the most accurate strategy. Possible future research could study the development of a more accurate merging policy for the information that can be extracted from text within this context. Such a policy should take into account and analyze the relationships between sensitive topics and personal data more in depth.

- Some of the sensitive topics, especially *Politics* and *Sexuality*, tend to change very rapidly due to the high social interest around them and to unpredictable events, such as the COVID-19 pandemic and the Ukrainian war, that make people's general moods and beliefs change significantly and rapidly. Such events could lead to a decay in terms of the model's ability to recognize the most up-to-date instances of sensitive topics. This could be likely to happen with the data set we have used, built from a single source (Reddit), despite considering many different sub-groups. It could be useful to analyze the model performance decay to apply proper continuous fine-tuning strategies.

  The topic of continual learning is, in fact, one of the hottest topics in machine learning.

Author Contributions: Conceptualization, M.P., S.C. and M.M.; methodology, M.P., S.C. and M.M.; software, M.P.; validation, M.P., S.C. and M.M.; formal analysis, M.P., S.C. and M.M.; investigation, M.P.; resources, S.C. and M.M.; data curation, M.P.; writing—original draft preparation, M.P., S.C. and M.M.; writing—review and editing, S.C. and M.M.; visualization, M.P.; supervision, S.C. and M.M.; project administration, S.C. and M.M.; funding acquisition, M.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Omigrade S.r.l. (https://www.omigrade.it/, accessed on 27 July 2022).

Data Availability Statement: The training and test data sets are publicly available on Github (https://github.com/michaelpetrolini/sensitive-data-finder/tree/master/datasets, accessed on 27 July 2022).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

GDPR	General Data Protection Regulation
EU	European Union
BERT	Bidirectional Encoder Representations from Transformers
NLP	Natural Language Processing
PTM	Pre-trained Model
FFN	Feed-Forward Network
ReLU	Rectified Linear Unit
HIT	Human Intelligence Task
RNN	Recurrent Neural Network

References

1. European Commission. *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the Protection of Natural Persons with Regard to the Processing of Personal Data and on the Free Movement of Such Data, and Repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA Relevance)*; European Commission: Brussels, Belgium, 2016.
2. Wachter, S. Normative challenges of identification in the Internet of Things: Privacy, profiling, discrimination, and the GDPR. *Comput. Law Secur. Rev.* **2018**, *34*, 436–449. [CrossRef]
3. Mondschein, C.F.; Monda, C. The EU's General Data Protection Regulation (GDPR) in a research context. In *Fundamentals of Clinical Data Science*; Springer: Cham, Switzerland, 2019; pp. 55–71.
4. Kretschmer, M.; Pennekamp, J.; Wehrle, K. Cookie banners and privacy policies: Measuring the impact of the GDPR on the web. *Acm Trans. Web (TWEB)* **2021**, *15*, 1–42. [CrossRef]
5. Bhaskar, R.; Laxman, S.; Smith, A.; Thakurta, A. Discovering frequent patterns in sensitive data. In Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 24–28 July 2010; pp. 503–512.
6. McSherry, F.; Talwar, K. Mechanism Design via Differential Privacy. In Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07), Providence, RI, USA, 21–23 October 2007; pp. 94–103. [CrossRef]

7. Dwork, C.; McSherry, F.; Nissim, K.; Smith, A. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography*; Halevi, S., Rabin, T., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; pp. 265–284.
8. Kužina, V.; Vušak, E.; Jović, A. Methods for Automatic Sensitive Data Detection in Large Datasets: A Review. In Proceedings of the 2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO), Opatija, Croatia, 27 September–1 October 2021; pp. 187–192.
9. Pattanayak, S.; Ludwig, S.A. Improving Data Privacy Using Fuzzy Logic and Autoencoder Neural Network. In Proceedings of the 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Lafayette, LA, USA, 18–21 June 2019; pp. 1–6. [CrossRef]
10. Attaullah, H.; Anjum, A.; Kanwal, T.; Malik, S.U.R.; Asheralieva, A.; Malik, H.; Zoha, A.; Arshad, K.; Imran, M.A. F-classify: Fuzzy rule based classification method for privacy preservation of multiple sensitive attributes. *Sensors* **2021**, *21*, 4933. [CrossRef] [PubMed]
11. Bucolo, M.; Fortuna, L.; La Rosa, M. Complex dynamics through fuzzy chains. *IEEE Trans. Fuzzy Syst.* **2004**, *12*, 289–295. [CrossRef]
12. IBM Security Guardium Data Protection. Available online: https://www.ibm.com/products/ibm-guardium-data-protection (accessed on 27 July 2022).
13. Azure Information Protection. Available online: https://azure.microsoft.com/solutions/information-protection/ (accessed on 27 July 2022).
14. Rubrik. Available online: https://www.rubrik.com/ (accessed on 27 July 2022).
15. Lin, T.; Wang, Y.; Liu, X.; Qiu, X. A survey of transformers. *arXiv* **2021**, arXiv:2106.04554.
16. Qiu, X.; Sun, T.; Xu, Y.; Shao, Y.; Dai, N.; Huang, X. Pre-trained models for natural language processing: A survey. *Sci. China Technol. Sci.* **2020**, *63*, 1872–1897. [CrossRef]
17. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
18. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
19. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the NIPS'16: Proceedings of the 30th International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
20. Peters, M.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K. Deep contextualized word representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, LA, USA 1–6 June 2018; Volume 1, pp. 2227–2237.
21. Kaur, M.; Mohta, A. A Review of Deep Learning with Recurrent Neural Network. In Proceedings of the 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 27–29 November 2019; pp. 460–465. [CrossRef]
22. Daniel Jurafsky, J.H.M. N-gram Language Models. In *Speech and Language Processing*; 2021; Third edition draft, pp. 1–29. Available online: https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf (accessed on 27 July 2022).
23. Zhu, Y.; Kiros, R.; Zemel, R.; Salakhutdinov, R.; Urtasun, R.; Torralba, A.; Fidler, S. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 19–27.
24. Angiani, G.; Cagnoni, S.; Chuzhikova, N.; Fornacciari, P.; Mordonini, M.; Tomaiuolo, M. Flat and Hierarchical Classifiers for Detecting Emotion in Tweets. In Proceedings of the AI*IA 2016 Advances in Artificial Intelligence: XVth International Conference of the Italian Association for Artificial Intelligence, Genova, Italy, 29 November–1 December 2016; Adorni, G., Cagnoni, S., Gori, M., Maratea, M., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 51–64.
25. Tan, S.; Zhang, J. An empirical study of sentiment analysis for chinese documents. *Expert Syst. Appl.* **2008**, *34*, 2622–2629. [CrossRef]