Exploring Abstract Concepts for

Images Privacy Classification in Social Media

BY

GABRIELE GALFRE'
Bachelor of Science in Computer Engineering, Politecnino di Torino, 2017

THESIS

Submitted as fulfillment of the requirements
for the degree of Master of Science in Computer Engineering
in the Graduate College of
Politecnico di Torino, 2019

Torino, Piemonte

Defense Committee:

Cornelia Caragea, Chair and Advisor, University of Illinois at Chicago

Enrico Magli, Advisor, Politecnico di Torino

Erdem Koyuncu, University of Illinois at Chicago

To my family.

ACKNOWLEDGMENT

Firstly I would like to thank my advisor Professor Caragea, for having been a guide and a constant source of help and suggestions, essential for the realization of this research. I would like to thank my Italian advisor from Turin Professor Magli for the support offered remotely.

This work represents the end of my career as a student and the most important achievement in my life by now. I would like to thank every person that was close to me during this long path and that helped me facing every difficulty with the best attitude.

I would like to mention here all my friends, old and new. They have been essential in creating the supportive, engaging and comfortable environment that allowed me to experience the best from the last few years. I have found in them a family with whom learn, grow, share experiences and have fun. I would like to thank in particular Federico, Arturo, Riccardo, Tommaso, Alessio and Aamir, with whom I shared most of my experience here in Chicago, and that became my closest family along this year. A special thank to Rafiki's squad and their members Squirrel Artu, Panda Dave and Edo the Frog for having been the best mates in the best adventure.

A special thank is dedicated to Beatrice, the person with whom I have shared the most in my life. Also the only person from Italy that faced the hardest winter ever in Chicago just to visit me in this 13 months and a half away from my home country.

Last but not least, an extremely disarming thank to my parents Patrizia and Mauro and to my amazing sister Giorgia. They taught me how to become what I am now in the best way

possible and they have always supported me, being close to me when I was in need and letting me go when I decided to. I will be extremely grateful to them for my whole life.

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

CHAPTER **PAGE**

TABLE OF CONTENTS (continued)

CHAPTER **PAGE**

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (continued)

FIGURE **PAGE**

SUMMARY

The work presented in this dissertation focuses on the task of predicting the privacy class of images posted on social media. In particular, the results we are going to show aim at supporting the hypothesis that abstract concepts are better suited for capturing the private nature of content shared online. The definition of abstractness we adopted along this work refers to the idea of something that is elevated from anything concerning the sphere of perceptions, difficult to be appreciated through our senses or impossible to conceptualize as something even remotely physical.

We developed a novel approach to investigate this hypothesis about abstractness in the context of a specific task. Specifically we applied this type of analysis on the textual user tags associated to a total of around 3 thousands selected images recently posted on Flickr. The privacy classification task we target is binary and consists in labeling posts as "public" or "private". In order to provide a solid foundation to our experimental setup, we evaluated the performances of different types of classification models, achieving results following a similar pattern. To the best of our knowledge we are the first facing this kind of investigation, trying to define some guidelines for the development of a methodology that could be applied to many different topics and used as proof for intensifying the focus of researchers toward concepts' abstractness.

In the effort of expanding the initial resources about words' abstractness, our contribution dealt with the task of scoring terms by abstractness, evaluating several techniques. Our ap-

proach made use of a dataset and samples' representations never tried before. After extensive analysis the results have been used for scoring, as precisely as possible, a set of unlabeled words, exploited for further experimentation in the privacy prediction task.

The results of this thesis' work are demonstrating the truthfulness of the hypothesis introduced, supporting it from different points of view. We conclude our analysis providing some insights about the directions the future works could follow starting from our conclusions.

CHAPTER 1

INTRODUCTION

1.1 Goal and Motivation

The goal of this thesis is to prove that abstract concepts are strongly correlated to privacy features, in the specific context of online media sharing.

A wide set of approaches have been experimented in the past for the prediction of images' private nature. Many of them, especially the ones involving automatic annotation of samples through visual features, are based on techniques that describe samples throughout characteristics that are concrete by nature. These methods achieve good results, but assume that this type of characterization is central for this type of task.

Our intuition is instead that abstract concepts are better candidate for the extraction of privacy related information from media. The main idea is that characterizing this type of resources with features related to sentiments, emotions and any other abstract object or notion would enable to discriminate more easily privacy classes.

Our investigation therefore is not trying to achieve state-of-the-art results in any of the task approached. It is instead specialized in creating the right condition and setup in order to be able to perform interesting comparisons from which derive supporting proofs.

Our main goal is in fact to create or support the foundation necessary to give the right push to the research efforts in the field of extraction of abstract information from different types of media.

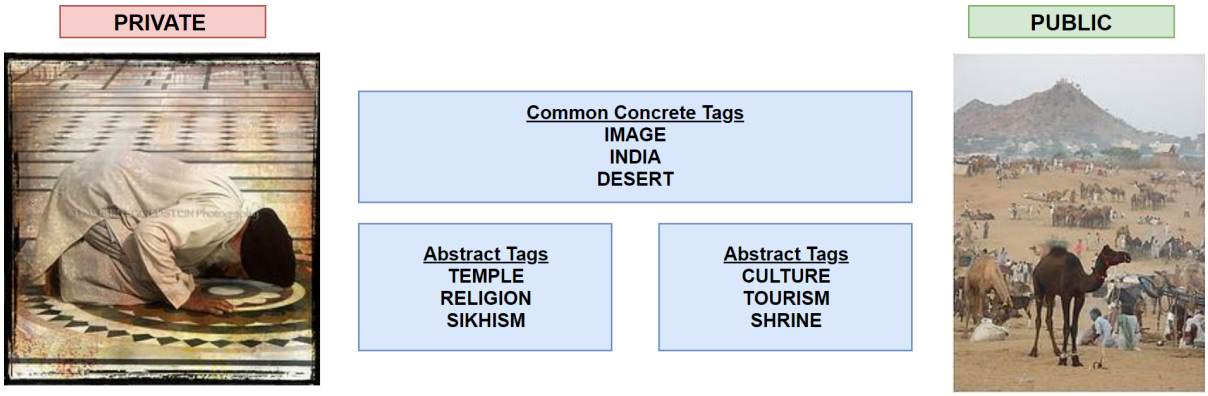

Figure 1: Example of importance of abstract information in the privacy prediction.

1.2 Thesis Organization

The dissertation of the work executed has been organized as follows.

We are introducing the topic faced by our investigation in chapter 2, summarizing the researches made by other original works. More specifically we are going to discuss about the two main fields interested by our analysis: privacy prediction task in social media and automatic scoring of words by abstractness. Furthermore, we will address the approaches and

architectures exploited in our research for the representation of textual and visual data, through a brief description of the contributions that have been essential for us.

In chapter 3 we are going to largely describe the approaches followed by our work in the context of privacy classification of images from social media. The focus will be on the process followed for the selection and manipulation of the dataset exploited and its analysis, as well as on the methodologies used for the execution of the classification. Particular attention will be dedicated on the description of all the models evaluated and the specific choice of the parameters.

Afterwards, in chapter 4 we will introduce the secondary task of scoring terms with abstractness values. Similarly to the previous chapter, we will describe the choices made in terms of dataset and approaches adopted for the realization of the task. In addition, for the sake of the final experiments that we will define in the following chapter, an in depth analysis of the results obtained for this task will be proposed and some conclusion derived. In the light of what emerges from this evaluation we will perform the scoring of a specific set of words, that is going to be useful later.

Chapter 5 represents the crucial part of this work, introducing our original methodologies tailored and developed appositely for the investigation of the truthfulness of the thesis we are supporting. The focus will be on the specific reasons and intuitions we decided to follow, in order to obtain the desired type of insights about the problem.

In Chapter 6 we are going to analyze the results of the experimental setup introduced. Several observation will be produced in support of our thesis.

The last chapter is dedicated to the final conclusions we have been able to derive and to possible future development of this work.

CHAPTER 2

RELATED WORKS

This chapter will focus on the work of several other researches in the fields of the investigation we are carrying on. The main topics we are concentrating on are the prediction of the privacy of social media content and the automatic scoring of abstractness. The ideas and approaches adopted in both topics are described, highlighting the contributions we have based our work on or taken inspiration from. In conclusion, the attention will be dedicated to those researches that developed interesting tools for the encoding of information of both textual and visual nature, which have been essential in our research for the representation of the samples.

2.1 Privacy Prediction in Social Media

The rapid increase in images shared on the Web fascinated researchers to focus on establishing adequate privacy predictive models to help protect users' sensitive information. Researchers also provided data on the awareness of people in relation to privacy risks associated with images shared online [1; 2]. Following this line of thought, several works were carried out to study users' privacy concerns in social network platforms, privacy decisions about sharing resources, and the risk associated with them [3; 4; 5; 6]. Additionally, several works on privacy analysis examined privacy decisions and considerations in mobile and online photo sharing [7; 8; 9]. For example, [10] studied the effectiveness of information about location and tags in predicting privacy settings of images. They also carried a study to verify whether the visual features are

relevant or not to an image's privacy and found that content is one of the discriminatory factors affecting image privacy, especially for images depicting people.

For the sake of the research we have worked on in this dissertation, two main automated image privacy lines of approach are worth citing, concerning some interesting results.

The first are visual based approaches. Several works used features derived from the images' visual content and showed that they are informative for predicting images' privacy settings. Given the recent success of convolutional neural networks, several works [11; 12; 13; 14; 15; 16] showed promising privacy prediction results, if compared with visual features such as SIFT and GIST. Other works, adopting the same type of features from convolutional neural networks, also started to explore personalized privacy prediction models [17; 18; 19].

The second type of approach, that is particularly important for our task, is the tag based one. Previous work in the context of tag-based access control policies and privacy prediction for images showed initial success in correlating user tags with access control rules. For example, [20; 21], [22], and [23] explored learning models for privacy prediction in images using user tags. They found that user tags are very informative for predicting images' privacy. However, the scarcity of tags for many online pictures [24] and the workload associated with user-defined tags influence badly the accuracy of analysis of images' sensitivity based on this dimension. Recently,additional studies [12; 13] showed that the images' tags automatically obtained from the visual content of images using CNNs can improve the performance of image privacy prediction. Yet, since these type of model are trained on datasets concerning the recognition of

objects and places in images, they are not able to capture the privacy orientation of the image while generating the tags.

The main purpose of our work is in fact targeting the hypothesis that privacy is not exclusively correlated to the material content of an image. On the contrary we believe that concepts with abstract nature are better candidate to capture the privacy orientation of

2.2 Abstractness Prediction

The idea of abstractness has been an interesting topic for many researchers. Some studies have investigated what it represents in term of cognition process in the human mind [25; 26] and how it affects decision making and learning tasks when they involve the representation of abstract and concrete concepts [27]. Fascinating theories have been formulated about what these two aspects of knowledge represents essentially in terms of brain activity. Several authors agreed on defining as concrete what can be experienced directly through senses and physical actions, while abstract are those concepts that need a certain level of rational processing to be represented.

A lot of effort has been dedicated to collecting words scored by concreteness [28; 29] as well as by other psycho-linguistic features. In this regard it is worth citing the MRC database [30] of manually annotated words, representing the first attempt of providing a solid base to these studies. Brysbaert et al. [31] recently collected a large dataset focusing on providing concreteness scores for 40 thousands words.

Various approaches exploiting supervised learning techniques have been experimented for concreteness scoring [32; 33], exploiting representations both related to concepts textual and

visual features [34]. An interesting unsupervised technique showed high performances, correlating abstractness to the context of usage of terms and particular syntactical features of English words [35]. It has been used to produce a dataset of 100 thousands unigrams scored by abstractness. Different studies have been successfully taken benefit from concepts abstractness for different tasks [36; 37] and our work follows a similar line of action, in the specific problem of privacy prediction.

2.3 Encoding Techniques for Textual and Visual Data

Many researches have targeted the task of compressing information and meaning from different type of media into numerical vectors. These type of representation allow the execution of models of very different type on top of these types of features, reducing the dimensionality of the representation. In this section of the related works we will focus on two types of data, which representation is of high interest in our work. We are referring to textual and visual data and we are going to present in detail the works that have been crucial for the execution of our experiments.

2.3.1 Word Embeddings

Word embeddings are a very successful type of representation for textual data. They have been defined as a way to reduce the dimensionality of text representation, by encoding each single term from the chosen vocabulary into a fixed size set of numbers. These values are computed in order to capture the semantic meaning of each word, generally based on the context of its usage in selected large corpora, according to the co-occurrence and similarity of usage with the rest of the words.

Different techniques and architectures have been developed and applied to this task, achieving results that have been largely adopted in researches related to the topic of natural language processing or simply where a representation with semantic meaning is required for text. This vector form of terms is particularly interesting because enabling researchers to easily define comparison metrics between words, able to capture many aspects of their real meaning.

Some of the most used word embeddings are word2vec [38], GloVe [39] and fastText [40]. They differ in many features, especially the architectures used for their extraction and the corpus on which they have taken the information about words usage and context.

In our applications we are going to make use of fastText, one of the latest approaches in word embedding, which showed very good results in several researches akin to ours. Here follows an in depth description of this type of word embedding.

2.3.1.1 fastText

The technique adopted behind this word vectorization has been very successful due to its low resources requirements for execution in the training phase, being able to obtain very good representation in short times. The trade-off that enables this technique, as well as others based on similar approaches, is the necessity of a very large corpus.

The idea at the base of fastText has been applied as used in [41] and it is called Continuous Bag of Words (CBOW) (see Figure 2). As the name suggests, for the representation of each word it takes into account symmetric contexts in the corpus of sentences, composed by the c words preceding and the c words following, where c is one of the parameters of the model. It

does not take into account the order of the words, but weights differently the distance from the the target word.

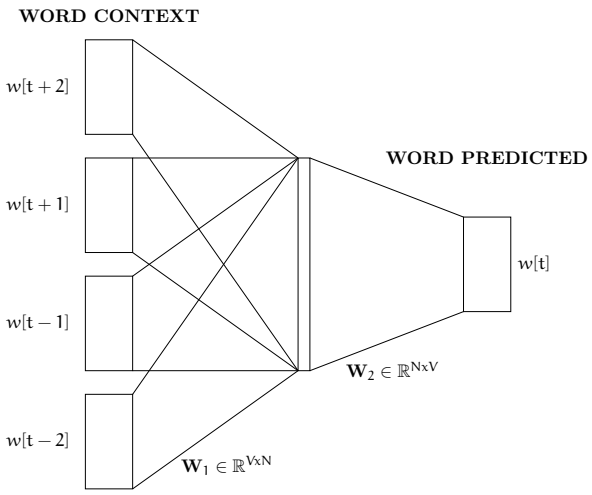

Figure 2: Scheme of the CBOW model.

The CBOW techniques gets the context of a word as input and predicts the words most likely to be associated to that context. It is based on the maximization of the log-likelihood of the probability of the words conditioned by their surrounding, which is represented by the following formula:

$$\sum_{t=1}^{T} \log(p(w_t|C_t)) \tag{2.1}$$

where T is number of terms used from the corpus, w_t for $t \in \{1, ..., T\}$ are the words considered from the corpus, C_t is the context of the term w_t.

The particularity of their approach resides in the formulation adopted for the conditioned probability which is expressed as:

$$p(w|C) = \log(1 + e^{-s(w,C)}) + \sum_{n \in N_C} \log(1 + e^{s(n,C)}) \tag{2.2}$$

where N_C is the set of negative words, which consists in a randomly selected set of words from the corpus never appearing in a context equal to C. This should take into account also the differences between words in the model.

The parametrization of the model is realized by defining the scoring function s, which is based on representing the predicted words through vectors v_w and the set of context's word by the average of the words $w' \in C$ represented by vectors $v_{w'}$. The scoring function is defined as follows:

$$s(w, C) = \frac{1}{|C|} \sum_{w' \in C} u_{w'}^T \cdot v_w \tag{2.3}$$

The parametrization used for predicted words and contexts' words are different.

Additional "tricks " have been applied:

- In order to avoid overfitting the representation on very common words and underfitting on not very common ones, a discard probability has been associated to each word. Each occurrence found in the corpus is used in the training step with the probability defined

by that distribution and these probabilities are depending on the frequency of the word (f_w) and a parameter t as follows:

$$p_{discard} = 1 - \sqrt{t/f_w} \tag{2.4}$$

- Words in the contexts are weighted by their distance to the target word by the definition of a weight vector for each position in the context. Consider $p \in \{-c, ..., -1, 1, ..., c\}$ as the set of positions in the context, d_p as the vector for weighting the word in position p and $u_{t,p}$ as the representation of the word in the context of the term t. The weighted version of the word at position p in the context of the word t is defined as:

$$d_p \odot u_{t,p} \tag{2.5}$$

FastText introduced a peculiar technique to enhance the representation of words that are more rare in the corpora, due to their frequency in the targeted language. It is based on the assumption that words representation can take advantage by the information extracted from the context of usage of the subwords they are formed by. With subwords of a word, they intended all the n-grams, with n maximum value as parameter, that are morphologically part of it. For example the word "word" is composed by the n-grams "wo", "or", "rd", "wor", "ord", etc. They simply applied the model just introduced considering also the n-grams as words. Once the representation of all of them has been computed they added the vector of each word to the vectors representing the related n-grams. This way they proved that this morphology based

semantic information is very useful for general words representation and specifically for rare terms.

They made publicly available the vectors trained with and without subwords, exploiting two particularly large datasets. Precisely they shared:

- 1 million words vectors based on Wikipedia 2017, UMBC webbase corpus and statmt.org news datasets, consisting in a total of 16 billion tokens.
- 2 million words vectors based on Common Crawl dataset, consisting in a total of 600 billion tokens.

The size of the pre-trained vectors available is 300 values.

2.3.2 Features Extraction from Images

For what may regard the extraction of features from images a lot of different techniques have been developed and largely applied. The state of the art in the extraction of such information is represented by the convolutional neural networks architectures applied to image recognition. They are able to identify objects from images very efficiently and with reasonably high accuracy. Their mechanism is based on the extraction of features maps starting from the pixels representing the images, applying subsequent levels of filters on them and producing deeper and deeper features maps. The output is then computed by a set of final fully connected layers mapped on a probability distribution by a soft-max layer. This last steps associate a probability to each of the object categories on which the model has been trained, expressing how it is likely that the related particular item is represented in the image. The training process of

these models is basically tuning the values composing the kernel of several sets of convolutional filters of different dimensions, automatically specializing on the detection of most informative features for the recognition of the desired objects.

The most notable event, that has been a real springboard for the majority of these architectures, has been the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [42]. This contest each year collects and evaluates the efforts of many researchers in applying novel techniques to the classification of images from the ImageNet dataset [43] into the subset of 1000 categories selected for the challenge.

Here we are going to introduce two particular architectures that have been particularly acclaimed by the reviewers of the context and found large use in many applications. They are based on specific expedient for both easing the training phase and making predictions more accurate.

2.3.2.1 Architectures with Residual Module

This architecture have been proposed by He et al. [44] and was the winner of the ILSVRC 2015, introducing a revolutionary module in the literature of convolutional neural networks architectures.

The author of this model addressed a very important and problematic behaviour of deep neural networks, particularly central for the convolutional neural networks because generally composed by high number of layers. The problem is related to the vanishing gradient, and how it influence the results in accuracy along with the increase of the depth of the network. It has been proven that deeper architectures are akin to model more precisely complex problems

such as image recognition, but after a certain threshold a degradation of the performances is appreciated with usual architectures.

They solved the problem by the idea of the "identity shortcut" in the architecture, which propagate the value at one middle layer to a deeper layer, skipping some of them. The idea is to avoid trying to fit the desired $D(x)$ mapping, where x is the input, but instead try to train the architecture to optimize the $F(x) = D(x) - x$ function. This way, in order to get the desired results, it is enough to add at the end of the module trained to model the residual function $F(x)$ the value of the input. A graphical representation of the model is shown in Figure 3.

For situations where the output of the module on which the residual approach is applied, the mere identity propagation of the input x is not possible due to the difference in dimensions between x and $F(x)$, therefore a projection is applied to make the shape of the shortcut equal to the output one.

They proved that this technique is easier to train in terms of how fast the weights update are able to find the right perturbation, and overcome the training degradation problem. Their contribution is very important because they solved a very important problem in gradient based learning, without adding additional complexity to the model: no new parameter to be tuned is in fact added.

The model they proposed that won the challenge is 152 layers deep, which was the deepest at that time. It is composed by two main modules type:

- Residual module where the shortcut is skipping 3 convolutional layers of 3x3 sized filters (*3-layer block*). See Figure 3;

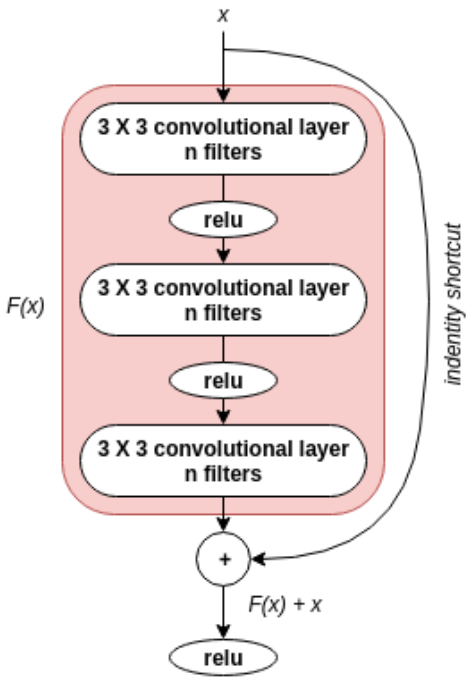

Figure 3: Basic scheme of the residual module used in ResNet152.

- Residual module where the shortcut is skipping 3 convolutional layers of 1x1, 3x3 and 1x1 sized filters in this order (*3-layer bottleneck block*). This technique is reducing the complexity of the model and can be used for channel dimensionality reduction, producing good performances. See Figure 4;

The architecture is shown in Figure 5 and consists in the first 101 layers as a succession of 3-layer bottleneck blocks, followed by the remaining layer consisting in 3-layer blocks. They adopted the projection shortcut for the changes in dimensions, while the number of filters for each convolutional layer are specified in the picture.

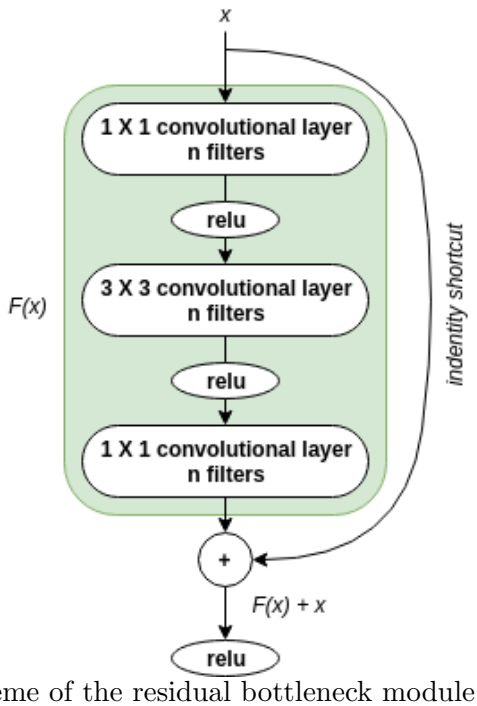

Figure 4: Basic scheme of the residual bottleneck module used in ResNet152.

The final layers of the architecture consist in average pooling,a fully connected layer and a soft-max. This final portion is mapping on 1000 output values, corresponding to the 1000 categories defined by the rules of the challenge.

2.3.2.2 Architectures with Inception Module

The inception module have been introduced in convolutional neural networks by Szegedy at al. in [45]. The idea at the base of this technique aimed at achieving higher performances in images object recognition, avoiding the simple stacking of deeper layers, which has been proven to strongly suffer from vanishing gradient, with subsequent degradation of accuracy. The novelty of the module they introduced resides in the parallelization of the flow in the

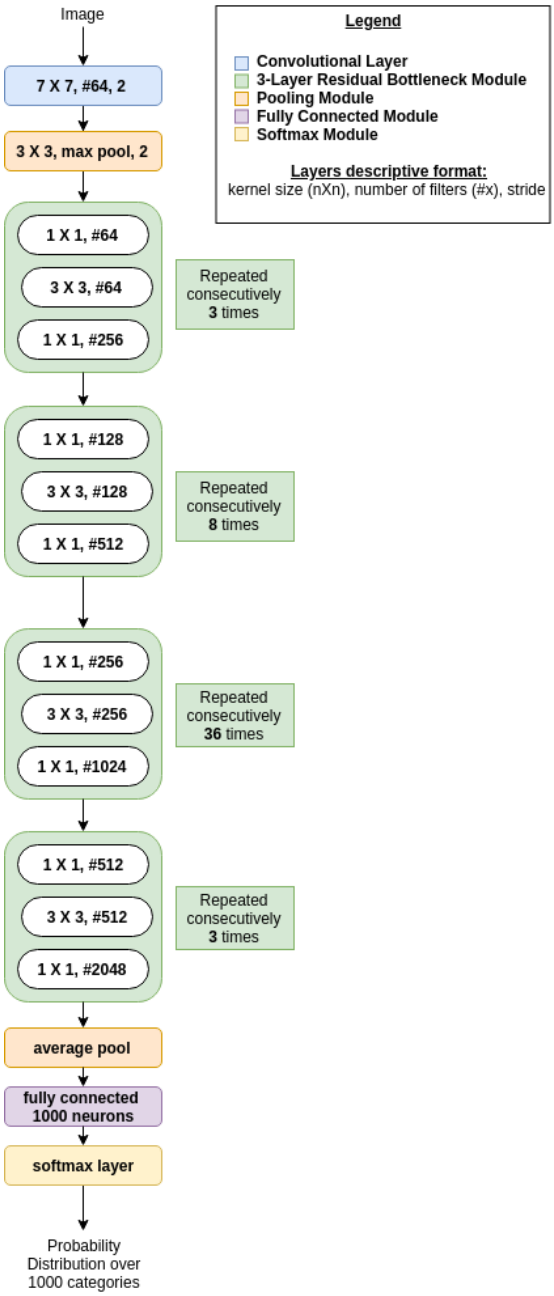

Figure 5: Basic scheme of the ResNet152 architecture.

architecture, applying sets of filters of different dimensions to the same input and concatenating the respective outputs.

The module has been improved adding 1x1 filters before the higher dimensional ones, with the purpose of reducing the number of channels (filters) and allowing cheaper convolutions. The problem of the vanishing gradient was also addressed by the usage of auxiliary classifiers at different depth of the architecture, which outputs are evaluated in the loss function, in order to back-propagate quantities that could diminish the effect of low gradients.

They proposed the architecture called GoogleNet, and based on the module just described. It is composed by an initial set of layers called the "stem", consisting in a sequence of convolutional layer with a max pooling in the middle. The final portion of the network is composed by average pooling, a fully connected layer with output of size 1000 and the final soft-max. Once again this configuration has been adopted to classify in the 1000 categories of ILSVRC.

The improvement proposed in [46] aimed at avoiding filters of large sizes (e.g. 5x5) because possibly introducing information loss by drastically reducing the size of the input. They realized it by substituting these convolutional layer by a sequence of smaller sized convolutional layers, reducing this way also the complexity of the model. For example they substituted 5x5 layers with two subsequent 3x3 layers (see Figure 6); the resulting inception module will be referred to as *inceptionA*.

Another improvement was introduced by factorizing the convolutions. Any layer involving nxn filters was substitute by two subsequent layers using 1xn and nx1 filters, defining the *inceptionB* module type (see Figure 7). Additionally, instead of using this factorization connecting

the substitute layers one after the other, they decided to connect them in parallel, concatenating all the results at the end of the inception module and defining *inceptionC* module type (see Figure 8).

In terms of grid size reduction, they also substituted the usual max pooling with an efficient grid size reduction that consists in an inception module using stride equal to 2 instead of the usual value 1. This module is referred to as *grid-size-reduction*.

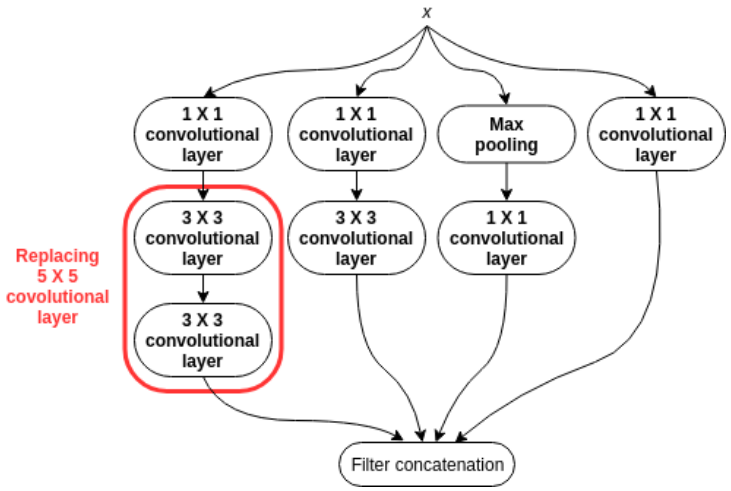

Figure 6: Basic scheme of the *inceptionA* module, with filter sizes used in InceptionV3.

The InceptionV3 architecture implements all the improvements of the inception module described above, largely exploiting the factorized 7x7 filters, and new regularization techniques:

- RMSProp Optimizer;

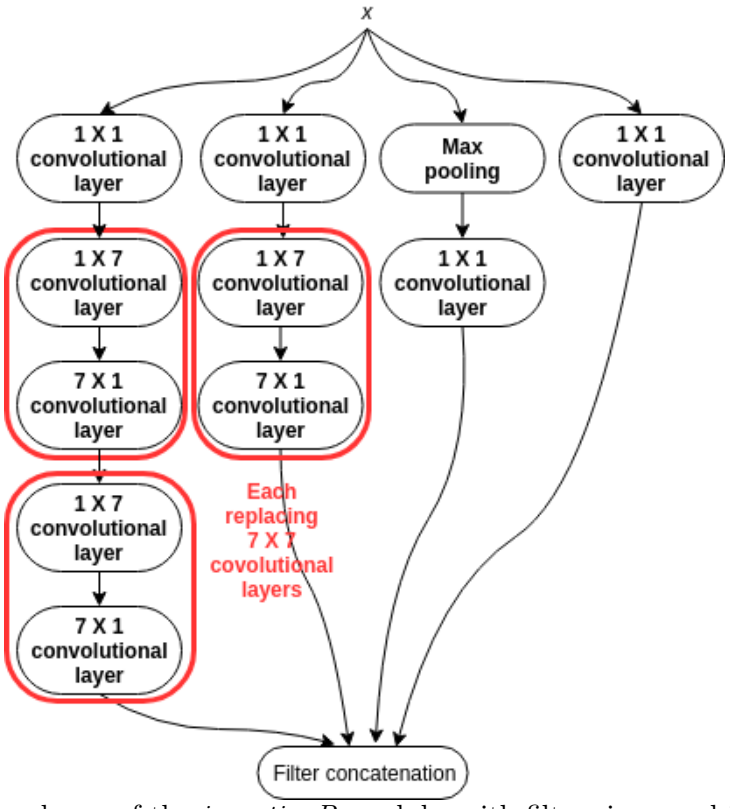

Figure 7: Basic scheme of the *inceptionB* module, with filter sizes used in InceptionV3.

- BatchNorm for the auxiliary classifiers;
- Label smoothing to avoid the model to overfit on specific classes.

Additionally they used only on auxiliary classifier composed by one convolutional layer of size 5x5 and 768 filters, followed by another one with 128 filters and concluded by a fully connected layers mapping on 1024 values. This feature has been used in InceptonV3 architecture as a regularization mechanism rather than a help to achieve deeper models. The complete

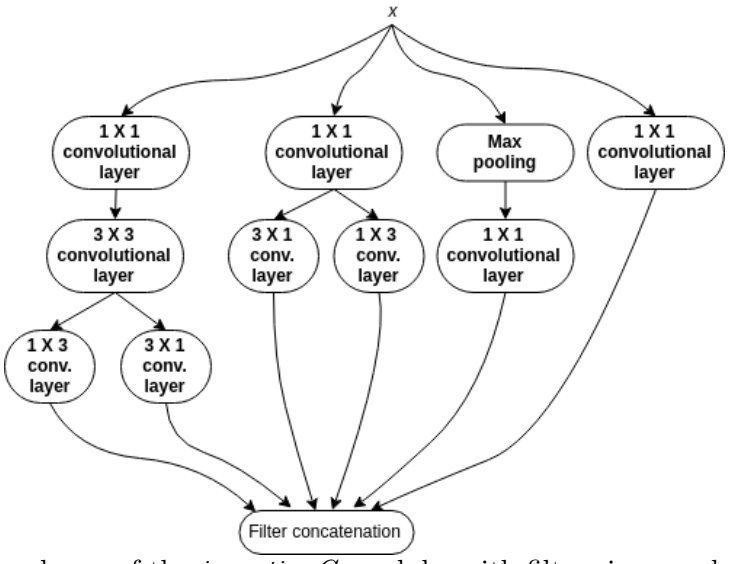

Figure 8: Basic scheme of the *inceptionC* module, with filter sizes used in InceptionV3.

architecture of the InceptionV3 model is composed by the following building blocks, in the order of listing:

- *stem* layers, composed by three subsequent convolutions, a max pooling layer, two convolutions and a final max pooling;
- a series of 5 *inceptionA* modules;
- a *grid-size-reduction* block;
- a series of 4 *inceptionB* modules;
- a *grid-size-reduction* block;
- a series of two *inceptionC* modules
- average pooling;

- final layers composed by dropout, fully connected with 1000 nodes and a final softmax to create the probability distribution on the ILSVRC categories.

Before the second *grid-size-reduction* block it is positioned the auxiliary classifier, which consists in average pooling, tow convolutions and the usual fully connected with softmax layers.

CHAPTER 3

PRIVACY PREDICITON TASK

The main task investigated by our research is the prediction of the privacy class of images posted on social media. In particular, the type of classification we approached is binary, consisting in attributing the *public* or *private* label to a set of selected samples.

An additional level of complexity has been added to this task, because our interest is in the analysis and comparison of the performances of abstract and concrete features in the context of this specific type of classification. For this reason the selection of the dataset and the features for its representation has been crucial in our approach.

Regarding the classification's techniques adopted, we aimed at generalizing as much as possible our results, focusing less on achieving high performances. This is the main reason behind the choice of the models, which have been chosen from the commonly known literature on the classification topic for their diffuse usage and not for the expectation of particularly good performances. In order to provide more interesting results, one particular neural network model has been chosen instead for its applicability to the problem and for the results achieved in similar tasks.

In this chapter we are going to introduce and explain the choices behind the setup adopted in our work for the privacy classification problem. Particular attention is going to be dedicated to the dataset and the models.

3.1 Dataset Preparation

The dataset we are going to adopt in our evaluation is the product of different choices that we are going to introduce here. In particular some datasets available from the results of other researches have been exploited for this purpose.

3.1.1 Online Privacy Reference Dataset

The first step for building our dataset is about defining the type of resource to use, which must be able to provide samples from social media labeled by privacy classes. At this regard there are not many available choices, therefore our attention was directed towards the Picalert dataset [22] largely used in the context of privacy prediction of social media, as introduced earlier in section 2. It consists in a collection of images posted on Flickr and manually annotated by people of different ranges of age as *private* or *public*.

The diversity of the contents available on Flickr in terms of images, in conjunction with the different concepts of privacy that can be obtained from a sample of people with different ages, make this dataset perfectly in line with the purpose of extracting a generalized enough understanding of privacy in social media.

3.1.2 Abstractness Scoring Reference Dataset

Consequently, considering the purpose of our investigation, it is essential to have a metric to evaluate the abstractness of concepts. Different researches appeared to be interesting for this scope, publicly providing large sets of words associated to scores indicating their abstractness or concreteness.

The choice has been made considering the data provided by two works in particular, introduced earlier in chapter 2:

- Brysbaert at al [31] expressed as concrete anything that:

  > "refer to things or actions in reality, which you can experience directly through one of the five senses. We call these words concrete words. Other words refer to meanings that cannot be experienced directly but which we know because the meanings can be defined by other words."

  They put the emphasis on the abstractness as the quality of a concept to be a linguistic or cognitive artifact. They asked a sample of people to label a set of approximately 40 thousands English lemmas with a concreteness score from 1 to 5. We will refer to this dataset as $D_{Brysbaert}$.

- Rabinovich at al. [47] used a weakly supervised model, which selected some concrete and abstract words from Wikipedia entry based on the morphology of the terms themselves. They got these words annotated, by a set of people, with abstractness score from 1 to 7, providing the definition:

  > "Words or phrases may refer to persons, places and things that can be seen, heard, felt, smelled or tasted or to more abstract concepts that cannot be experienced by our senses. [...] concreteness in terms of sense-experience."

Exploiting examples of usage of these terms in sentences, they trained a recurrent neural network model to label 100 thousands unigrams with abstractness scores in the range between 0 and 1. We will refer to this results as $D_{Rabinovich}$.

For the selection of the reference abstractness dataset we decided to manually evaluate the words in their overlap. We could notice many samples where the scores where presenting high discrepancies, characterizing some word with abstract connotation in one and concrete in the other, and vice versa. Manual considerations, directly made by us, brought our decision towards $D_{Rabinovich}$, because better fitting the idea of abstractness that we have. Both datasets presents some objectively recognizable imprecisions, caused by different reasons. For demonstration purposes we are showing in Table I a list of words with high disagreement between $D_{Rabinovich}$ and $D_{Brysbaert}$.

The differences in the scores from the two dataset, in our pinion, are strictly correlated with the slightly different definition adopted for the concrete and abstract concepts. For example Brysbaert et al. emphasized the language related nature of the formulation of abstract idea, which can lead annotators to confusion if it differs from their own idea of abstractness. Also the difference in the scale of scores they adopted is probably influencing the annotation, for both the width of the range, and for the order of the scores: one proposed increasing concreteness scores, the other increasing abstractness scores.

The other important reason for the selection of $D_{Rabinovich}$ has been the better overlapping with the privacy dataset, allowing us to deal with higher amount of samples.

Word	$D_{Rabinovich}$ score	$D_{Brysbaert}$ score
abortion	0.7809	0.3975
accessibility	0.2133	0.675
adult	0.5311	0.15
analytics	0.3294	0.6425
insecurity	0.2124	0.87
availability	0.3784	0.67
brightness	0.6046	0.3925
collaboration	0.3932	0.6025
counting	0.5844	0.3575
dedication	0.4005	0.71
dictator	0.6284	0.1775
extinction	0.3161	0.6425
fascinated	0.0883	0.81
fructose	0.6234	0.1925
funeral	0.6077	0.2925
headache	0.7363	0.275
individual	0.7511	0.37
parent	0.6535	0.11
pregnancy	0.6425	0.1725
preacher	0.66	0.075
organism	0.6294	0.21
pain	0.6507	0.375
raven	0.5791	0.035
smart	0.2073	0.8125
technologic	0.1794	0.6975
universe	0.6597	0.2875

TABLE I: Datasets scoring discrepancies examples.

3.1.2.1 Samples' Features of Interest

The first step for the selection of the samples for our dataset has been to decide the type of features we needed for their representation.

We started considering that the effort applied in research towards the evaluation of abstractness has been mostly strictly related to words, as the most common way to identify concepts and meanings.

Another important consideration, given the visual nature of the posts on social media such as Flickr, is that generally the extraction of meaning from images and video in state-of-the-art researches is generally aiming at information associated to concepts of concrete nature. Some examples are the results obtained in image recognition models, which are usually used to identify physical objects or actions. We haven't been able to find notable approaches targeting the extraction of information of abstract nature from pictures or visual representations in general.

Starting from these considerations, the only type of data we are able to exploit and discriminate as abstract or concrete is of textual type, therefore our classification task will focus on representing the images through the user tags they have been posted with. We believe that the quantity of tags normally associated to the images from the Picalert dataset, and on Flickr in general, is big enough for being able to represent the samples in a satisfying way. Therefore we are going to represents our samples through sets of words rather than visual features of any type.

3.1.2.2 Samples' Selection Criteria

Once both the initial privacy and abstractness datasets have been defined we decided to select a subset of samples with the specific purpose in mind of using them in an experiment setup appositely tailored that we will introduce later in chapter 5. For the purpose of the explanation given in this section, we will simply say that the main factor that interests us in

the experiments is the ability to represent each of the samples using both abstract and concrete information. Considering this aspect, we decided to apply the following constraints for selecting the samples from Picalert to be part of our dataset:

- Presenting at least 2 abstract tags;
- Presenting at least 2 concrete tags;
- Presenting at least 10 tags in total, not counting as valid the unscored words not available in fastText embedding vocabulary.

The first two constraints set a minimum of words for representing the samples from both the abstract and concrete point of view, providing enough descriptive information. The choice of using 2 as threshold followed a superficial analysis of the tags distribution in the original dataset, which showed that using higher values would have reduced the number of samples to a too low quantity in our opinion, possibly causing problems in the execution of a correct classification evaluation.

The third restriction is instead deriving from the second step of our experiments, which involves the automatic abstractness scoring of words not available in $D_{Abstractness}$. We decided to consider samples allowing us to increase the number of abstract and concrete words used for their representation, in order to evaluate how the consequences in the privacy prediction task would be. The limitation on the fastText embedding availability is related to choices we are going to introduce in the chapter related to the abstractness scoring task (4).

3.1.2.3 Abstract and Concrete Discrimination

Some manual analysis of the abstractness dataset showed us some interesting features in the actual words scoring. We noticed that the terms belonging to the central range of values, precisely between 0.4 and 0.6, seem to be not perfectly scored. It is obviously difficult to locate a precise value dividing abstract from concrete because of the continuous nature of the adopted scale. The interesting thing is that there are instances of terms that for many people would be ordered by abstractness in a certain way, but appear distributed in the dataset with different relative positions. The explanation we decided to follow is that there are many words that are not easy to be precisely arranged on a continuous scale of abstractness intensity. Another particularity is that some words carry multiple meanings, and those can be correlated to opposite positions in the abstractness range. Any dataset based in any way on manual annotation, especially if averaging the labels given by different people on the same sample, would be affected by this imprecision, directly derived from the subjective judgement of the people involved in the process.

We considered the option of not using the words in this range of values in our dissertation, but we also noticed that many of the words in it are instead correctly positioned on one of the two extremes of scores and more importantly are very meaningful words for both the abstract and concrete classes. We concluded deciding to keep all the words available in the dataset.

The discrimination of these terms in the abstract and concrete classes has been executed using as a threshold score the central value of 0.5. This choice will surely cause some of the words in the central range of values introduced above to be misclassified, but we believe that

the error would be similar in intensity for both classes and would not involve a considerably big quantity of terms. Also this is the trade-off we had to make in order to keep all the other correctly scored words available in that range.

3.1.3 Selected Privacy Dataset

The dataset resulting from the selection process just introduced has been referred to as the privacy dataset ($D_{Privacy}$). It is composed by approximately 3 thousands samples and will be used in all the experiments involving the privacy classification task. Some statistics about the features of its content have been shown in Table II for informative purposes. It is going to be helpful to keep in mind this type of insights during some of the considerations that are going to be stated in the next sections. In order to provide deeper information on the type of words involved, Table III shows instead the distribution of words according to part-of-speech typologies.

Class	Pics	Abstract Tags		Concrete Tags		Unscored Tags	
		average tags per picture	All	average tags per picture	All	average tags per picture	All
Private	1072	3.28	469	7.11	1658	8.71	3126
Public	1853	3.07	787	8.69	2839	9.90	5754
All	2925	3.15	913	8.11	3322	9.46	7077

TABLE II: $D_{Privacy}$ dataset statistics.

Class	Abstract Tags				Concrete Tags				Unscored Tags			
	Noun	Verb	Adj.	Adv.	Noun	Verb	Adj.	Adv.	Noun	Verb	Adj.	Adv.
Private	445	7	16	0	1604	19	27	7	2724	49	284	52
Public	753	9	23	0	2739	29	55	14	5062	106	476	86
All	874	12	25	0	3206	39	59	16	6258	128	555	110

TABLE III: $D_{Privacy}$ dataset statistics on part-of-speech tagging.

3.2 Classification Models

The classification task itself has been executed adopting and comparing different types of classifiers. This choice has been leaded by mainly two reasons:

- The need of showing relevant results through different types of classification models in order to provide more strength and general agreement to the conclusions we will extract from the results;
- The necessity of trying different representation of the samples, as well as experimenting the diverse features of the classifiers. This would allow us to better understand the results obtained in term of what each classifier is specialized in capturing.

This subsection will provide a summary of the models adopted, the hyper-parameters and variant chosen and the input representation selected. The approaches adopted are all classified as supervised learning, therefore making use of the privacy labeled dataset introduced in the previous subsection.

3.2.1 Naïve Bayes Classifier

This is the simplest approach we explored for the classification. It is based on the naïve assumption that each of the features used for representing the sample are statistically independent with each other. This is the foundation of this method, which classify a data-point by the computation of the conditioned probability of the features it is represented with the different classes, selecting the class showing highest probability value. The implementation variants we considered in our exploration are all provided by scikit-learn library [48].

3.2.1.1 Model's Insights

More specifically it computes the conditioned probability of each feature, for each class, considering the training set as the corpus of reference, then for a new data-point computes the probability of belonging to each class, conditioned by the values of its features.

The mathematical dissertation behind it is based on the following step, making use of the following definitions:

- n and K are respectively the number of features of the samples representation and the number of classes;
- $\mathbf{X}$ represent the features vector of a generic sample and its components are identified by x_i where $i \in \{1, ..., n\}$;
- C_k represent the classes, where $k \in \{1, ..., K\}$;
- $p(\cdot)$ is the probability operator;
- $(\cdot|\cdot)$ is the conditioned probability operator;

- Z is a constant positive scaling factor whose value is not relevant for this context.

The theoretical foundation of this classifier is the following theorem:

$$\textbf{Bayes' Theorem}: \quad p(C_k|\mathbf{X}) = \frac{p(C_k)p(\mathbf{X}|C_k)}{p(\mathbf{X})} \tag{3.1}$$

It is used to compute the conditioned probability for a set of features to belong to a certain class as:

$$p(C_k|x_1, ..., x_n) = Z \cdot p(C_k) \prod_{i=1}^{n} p(x_i|Ck) \tag{3.2}$$

The training process simply computes, based on the content of training set, the values of $p(C_k)$ $\forall k \in \{1, ..., K\}$ and $p(x_i|C_k)$ $\forall i \in \{1, ..., n\}$, $k \in \{1, ..., K\}$. The final class y is predicted by:

$$y = \underset{k \in \{1,...,K\}}{\arg\max}\, p(C_k|x_1, ..., x_n) \tag{3.3}$$

3.2.1.2 Variants Explored

This model is used diffusely for the classification of documents composed by sets of words which order doesn't need to be contemplated, following the bag-of-words assumption.

We tried to use this model in different variants and different input representations. For instance we evaluated the *Gaussian Naïve Bayes Classifier*, which is suited for features with continuous values, adopting a tf-idf sample representation. Preliminary experiments confirmed that this approach wasn't giving us satisfying results, therefore we concentrated on a different type of representation. We decided to adopt the simpler tf representation, in combination with

Complement Multinomial and *Bernoulli Naïve Bayes Classifiers.* Our case is applicable to both variants, because each user tag can be present or not in a sample, excluding the possibility to count it more than once.

In the validation process of this model we considered the two variants just introduced, with tf samples representation and with the add-one smoothing.

3.2.2 Support Vector Machine Classifier

This model is widely used in classification problem. It works by the selection of particular data-points in the training set that are close to element of the other class. It exploits the points to find the hyper-plane maximizing the distances to these support vector points. The resulting plane should be a good separation between the two classes. We exploited the implementation provided by scikit-learn library [48].

3.2.3 Model' Insights

The model is basically representing in a high dimensional space, with same dimensionality as the one of the data-points, all the samples from the training set. It simply aims at solving a quadratic programming problem that practically corresponds to finding the equation of the hyper-plane that separates the points of the two classes, also maximizing the distance between each point for the two classes. The additional soft-margin technique add to the objective function of the quadratic programming problem a penalty factor that gives a non zero contribute only for the data-points that happen to be on the wrong side of the hyper-plane, directly proportional to their distance to it. This feature introduces tolerance to the model, which

allows solutions identifying a separation hyper-plane that doesn't completely separate the two classes.

The quadratic programming problem introduced above can be formalized as follows. Consider the training vectors $x_i \in \mathbb{R}^p$, with $i \in \{1, ..., n\}$, labeled according to two different classes 1 and -1, and the vector $y \in \{1, -1\}^n$ which contains at the position i the value corresponding to the class associated to the training point i. n is the number of training data-points and p is the dimension of each of them. The quadratic programming problem solved by the support vector machine aims at the minimization of the following objective function:

$$\min_{w,b,\zeta} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \zeta_i \tag{3.4}$$

subject to the following constraints

$$y_i(w^T \phi(x_i) + b) \geq 1 - \zeta_i, \tag{3.5}$$

$$\zeta_i \geq 0, \text{ for } i \in \{1, ..., n\}$$

The solution finds the values of w, b and ζ_i for $i \in \{1, ..., n\}$. The first two identify the hyper-plane in the space of the training data-points, while ζ_i are optimized for the definition of the margins. The most important parameter is C which defines the intensity of the penalization given by the soft-margin technique and act like a regularization factor, used to balance between

better fitting the data and avoiding overfitting. Higher values of C are specializing the model on the training set, while lower values allow higher misclassifications.

The ϕ function consists instead in the kernel trick. This technique is simply mapping through a pre-defined function (the kernel) each data-point in a different dimensional space. This trick allows to have the model solving the same exact problem, but creating hyper-planes that are not linear, but mapped on a polynomial, exponential or other type of function, helping the model to better fit the data for some datasets.

3.2.3.1 Hyper-parameters Exploration

The input have been encoded using the tf representation. After a preliminar evaluation of the performances using tf-idf we decided to avoid using it. Each input feature has been scaled, by subtracting the mean and dividing by the standard deviation, in order to obtain better result with this type of model. The model has been exploited using a balanced weighting of classes because of the unbalanced dataset we are using. This option weights the elements of the different classes proportionally to the population size of the classes themselves, providing better results. Other than that, we decided to use the radial basis function kernel concentrating our attention on the commonly used γ parameter set to the inverse of the product of the number of features and the variance of the training set ("scale" value). We adopted for γ the values "scale", "scale"/10 and "scale"·10. The C values explored are the following: 0.01, 0.1, 1, 10, 100, 1000.

3.2.4 Random Forest Classifier

This type of classifier is based on an ensemble of *Decision Tree* models all trained on the same training set, but with some randomness in their tuning. The variant we decided to use is provided by scikit-learn library [48] merges together the results obtained by each tree by averaging the probabilistic predictions.

3.2.4.1 Model's Insights

The randomness of the models considered in the ensemble is realized by training them on different subsamples of the same training set and using only a subset of the total features available in the input representation. This allow each tree to capture particular features of the dataset with high variance. The combination of the trees into the forest has the effect of reducing the overall variance, decreasing this way the chances of overfitting.

The variant of the model we decided to use instill randomness in each tree by controlling three main parameters:

- The number of trees generated in the ensemble.
- The maximum number of features evaluated in the creation of new nodes in the trees, which allow each of them to randomly focus their search on specific subsets of the input representation.
- Our variant of the random forest uses for each tree a training set with the same size of the complete one, but creates it by randomly sampling from it. The choice of adopting the sampling with or without replacement in this step is a parameter influencing the randomness of the trees.

The rest of the parameters that we can select in our application are instead referred to the techniques adopted in the creation of each of the decision trees. Decision trees consist in a branched map of choices, guiding the classification of a data-point. Each choice splits the decision branch in multiple subbranches, based on the value of a specific feature associated to the particular decision node. When the path reaches a leaf node of the tree, it defines the class of the sample considered.

The basic mechanism behind the creation of a tree concerns the choice of a features on which defining a threshold value for the creation of a branch. This choice is influenced by different factors such as the metric used to evaluate the goodness of a branching and the number of features considered for the branching. The other factor influencing the creation of trees is whether to try to generate a new node or not, creating therefore a leaf node. This choice is instead influenced by constraints defined on the minimum number of training samples to allow the branching, minimum number of training samples allowed in a leaf node, the depth-level of the node and maximum number of leaf nodes.

The choice of the parameters related to the creation of the trees is tuning how each of them can be trained specifically on the subset of features and samples assigned. We are balancing the variance of the singular trees through these choices, influencing the chances of overfitting on each subset of the training set.

3.2.4.2 Hyper-parameters Exploration

Once again preliminary evaluation on the dataset revealed better performances representing the data points through tf encoding, rather than tf-idf.

The hyper-parameters we decided to experiment for this model are several and we are going to list them here. We have tried:

- both variant with an without sampling with replacement;
- number of estimators equal to 50 and 250;
- limiting the maximum number of leaf nodes to 500 and leaving it unlimited;
- minimum number of samples to allow splitting equal to 2 and 5;
- minimum samples per leaf equal to 1, 2 and 5;
- maximum number of features evaluated in the generation of nodes equal to the squared root of the total number of features;
- entropy as the metric used for choosing the best spitting criteria;
- no constrains on the tree depth.

3.2.5 Text Convolutional Neural Network Classifier

This particular model has been proposed by Yoon Kim [49] and consists in applying the idea of convolutional neural network to input representing sentences. This approach is completely different if compared with the ones previously introduced because it takes in consideration the order of words, exploiting a totally different input representation.

3.2.5.1 Model's Insights

The architecture proposed here has been developed following the idea of training sets of filters of different dimensions on the representation of an ordered set of words. The filters have

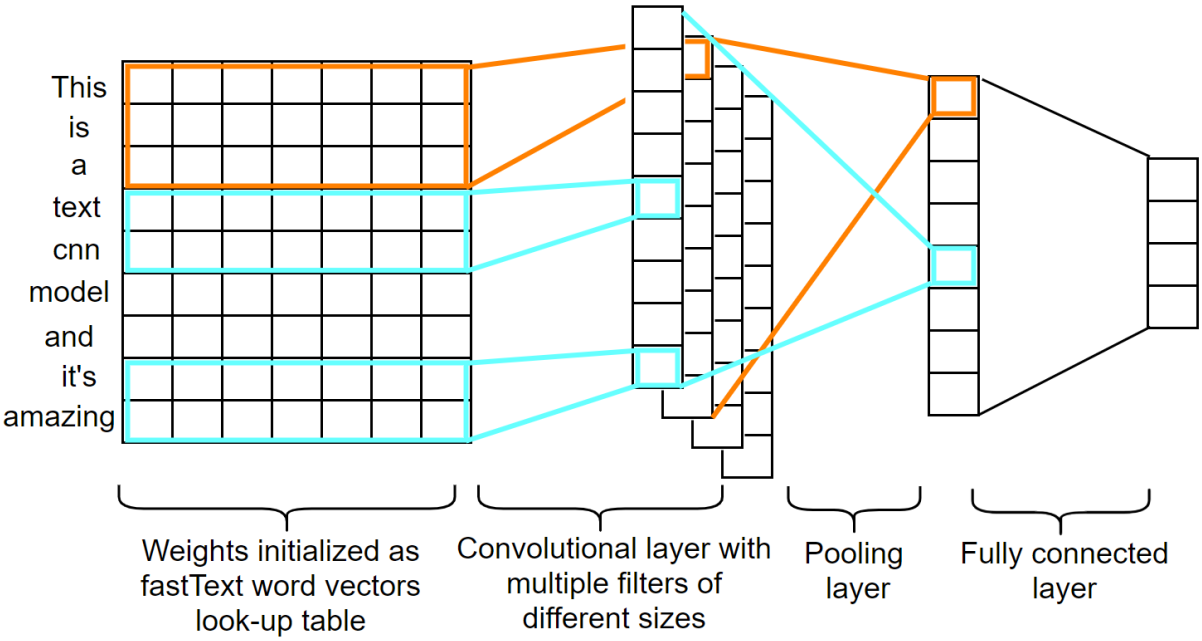

Figure 9: Visual representation of text convolutional neural network model.

a shape which is mono-dimensional and correspond to their width, or in other word the number of words included in the sliding window that is passed along the sentence. These filters are used to extract different features from the sentence and then fed in a fully connected layer for the prediction of the class.

More in detail, each filter is computing a value for each position of the sliding window on the sentence. If the positions of the sliding window are $j \in \{1, ..., n - h + 1\}$, with n as the length of the sentence and h is the filter size, we have:

$$x_{j:j+h-1} = x_j \oplus ... \oplus x_{j+h-1} \tag{3.6}$$

$$c_j = f(\mathbf{w} \cdot \mathbf{x}_{j:j+h-1} + b) \tag{3.7}$$

Where x_i for $i \in \{1, ..., n\}$ is the representation of the i^{th} word in the sentence, k is the length of words representation, $\mathbf{w} \in \mathbb{R}^{kh}$ are the weights of the filter, $b \in \mathbb{R}$ is the bias term and f is the activation function. The concatenation operator is $\oplus$ and $\mathbf{x}_{a:b}$ indicates the concatenation of the words' representations from x_a to x_b.

The set of $n-h+1$ features extracted by each filter is called feature map and the maximum value is selected from it. This layer is called the max-over-time pooling and produce a total number of values equal to the number of filters. A fully connected soft-max layer is applied on this set of values, extracting a probability distribution over the number of classes desired: two in our case.

It is important to specify that the value of n, in the training process on a specific training set of sentences, is equal to the maximum sentence's length available. Any phrase that is shorter than that has been filled with trailing padding values.

3.2.5.2 Model's Training and Features Exploration

First of all, the hyper-parameters of the variant of this architecture that we decided to use have been mostly set with the author's suggested values, we just explored some values for regularization purposes, in order to avoid overfitting. We also decided to represent samples by sorting in alphabetical order the tags associated, so that if two tags are both present they are always in the same, or similar reciprocal positions.

In terms of architecture we used filters with of 3, 4 and 5 width, training 128 filters for each size. Furthermore the input words have been encoded adopting 300-values sized fastText word

embedding vectors, allowing the training updates also in the first layer, resulting in a fine-tuning of the embedding values. This last design choice has been selected because performing better than both the non-trainable version and the one using trainable random initial weights. The activation function used for the computation of the features extracted by the filters is the *relu*: $f(x) = max(0, x)$.

Regarding the training process of the model the *softmax cross entropy loss* $L(T, Y)$ has been used on the values obtained from the output of the fully connected layer:

$$L(T, Y) = -\sum_{i=1}^{n} \sum_{c \in C} t_{ic} \cdot log(y_{ic}) \tag{3.8}$$

where T is the set of flags t_{ic} associating each sample $i \in \{1, ..., n\}$ to the classes $c \in C$, n is the number of element of the batch and C is the set of classes. t_{ic} flags values are 1 for the class associated to sample i and 0 otherwise. $y_{ic} \in Y$ are instead the output value produced by the model for sample i for the class c.

We adopted the *Adam optimizer* for weights update, which consists in an adaptive momentum estimation through moving average on the batch results, tuning the learning rate accordingly. The only parameter we fine tuned is the initial learning rate, adopting values from within the following set: 0.001, 0.00025, 0.0001.

For avoiding overfitting we applied 0.5 drop-out probability and tried l2-regularization with γ equal to 1, 0.1 and 0.01. The training termination criteria adopted is the early stopping,

interrupting the weights update once the validation set loss function starts to increase for 3 consecutive epochs.

CHAPTER 4

ABSTRACTNESS SCORING TASK

A secondary task faced by this research is the scoring of words by abstractness. This task has been approached by different researches and there are many resources available of scored terms, from which take advantage for labeling new samples.

Our investigation here is strictly related to the primary privacy prediction task. In fact the final purpose of the evaluation of scoring techniques is to enable us to enlarge the set of available words labeled by abstractness and evaluate them in the final experiment setup.

The main steps of this task consist in selecting reference resource for scored words, some scoring methodologies and an evaluation technique. In the process of choosing the methodologies and the dataset to use, different considerations have been made and new approaches evaluated. In this section we are going to explain in depth this procedure.

In conclusion we are going to discuss the scoring of the unlabeled words we will need in the privacy classification dataset.

We would like to anticipate that the set of results of the evaluation of the scoring techniques is going to be shown and intensively analyzed here. We decided to introduce them here and not in 6 because we believe them to be partial evaluation, not directly related to the final purpose of this dissertation. It is important though to mention them because representing an interesting example of both regression models evaluation and their application to an unlabeled dataset

4.1 Dataset Preparation

Firstly, for this portion of the dissertation it is necessary to introduce the steps behind the selection of the resources involved in the dataset we exploited for this analysis. We will start introducing the reasons behind the choices of the features for the input data representation, following with the actual dataset selection.

4.1.1 Features Representation

We decided to experiment two different types of representation of our data samples.

We occurred to consider that the abstractness of concepts is correlated to the semantic of the associated words and in particular to the way they are used in the context of sentences. The first type of representation we adopted is in fact completely related to the usage of the meanings in textual contexts.

Additionally, we took inspiration from the work of Bhaskar et al. [34], who faced the inverse problem of scoring words by concreteness. They explored the challenge evaluating the performances of scoring techniques encoding the input words used by their models, through features extracted by image recognition architectures from pictures correlated to the terms involved.

4.1.1.1 Textual Features

After extensive searches, looking for the best distributional model for the representation of words by they textual usage, we decided to exploit a particular word embedding representation. This type of words encoding techniques is generally based on the training of a neural network

architecture, focused on natural language processing, capturing in a numerical representation the usage of selected words in the sentences of a selected corpus.

At this regard the options available are several, and considering the purpose of this research we avoided to extensively try all of them to find the best performing ones. Actually we decided to follow the results obtained by several other researches to guide our choice:

- Rabinovich et al. [47] compared, for the automatic scoring of words, the usage of different word embeddings encoding such as Google word2vec [38], Glove [39] and fastText [50]. They observed the best performances in this particular ranking task using fastText embedding.
- Charbonier and Wartena [51] also targeted the task of concreteness scoring of words, achieving better results using fastText [40] rather than GoogleNews [41]. They found as particularly performing the fastText embedding version trained on the Common Crawl corpus and without subword information.

Given the analysis offered by the results of these works, we decided to adopt the fastText word embedding trained on Common Crawl and without subword information. This type of encoding resulted in very good performances generally, due to the innovative approaches used by it in the application of the CBOW model (see section 2.3).

The flavor of fastText embedding we decided to use is composed by a vocabulary of 2 million words and each of them is encoded by an array of 300 numerical values.

4.1.1.2 Concatenation of Textual and Visual Features

The alternative representation we explored for our data points consists in visual features related to each word. As introduced above we are following the ideas proposed by Bhaskar et al. [34].

In their work they explored the comparison of performances using words co-occurrences counts, word embeddings, features extracted from image recognition models and their concatenation in the task of words' concreteness ranking.

Regarding the specific features used in their work, they involved word2vec [38] for the text based representation and the features extracted by the AlexNet [52] and GoogleNet [45] as the visual counterpart. For what concerns the results with the regression model, they showed better performances with GoogleNet features and using the co-occurrences count. In general the results obtained with the concatenation of visual features and word2vec were slightly worse.

Another aspect that emerged from their experiments is that the best approach, for applying regression models on a dataset of concreteness scored words, is to exploit the samples on the whole distribution in the range of values. The results obtained exploiting only samples in the extremes of the range or in the central portion reflected worse performances, probably due to the low number of available samples.

Features wise, We decided to follow a similar approach applying some changes:

- The visual features we are going to use are extracted from newer architectures that achieved better results in image recognition with respect to AlexNet or GoogleNet;

- The images we are going to use for the visual features extraction will be strictly related to the context of social networks, in particular they are consisting of images from Flickr. The relationship between an image and a word is realized when the word appears in the set of user tags associated to the picture;
- We will only explore the concatenation of visual features and word embeddings variant. It is in our believes that in the bigger dimensions of our context, which aims at the final scoring of 7077 unlabeled words, using word embeddings instead of co-occurrences count vectors would provide a better distribution for our samples. In addition the particularly good results obtained by fastText embedding in previous researches (as introduced above) lead us to this decision.

As announced above, we decided to exploit and evaluate two specific convolutional neural networks for the extraction of the visual features, acknowledged to be the current state-of-the-art in object recognition from picture: Details about both of them have been introduced already in chapter 2:

- ResNet152 [44] is the particular variant of the convolutional neural network architecture implementing the residual module. This particular architectural feature performed particularly well in the ILSVRC 2015 challenge, especially the deeper variant we have chosen with 152 layers.
- Inception-V3 [46] is instead an improved version of GoogleNet that achieved very good results also in the ILSVRC 2015 challenge.

From both the architectures we decided to extract the features from the versions trained on the 1000 objects categories from ImgeNet dataset [43] defined in the ILSVRC challenge itself. In relation to the choice of the layers elected to be the ones providing the features desired, we decided to select the last layers before the final fully connected and soft-max ones, after the average pooling. The output of this layer, present in both the architectures, is characterized by output's dimension independent from the number of classes used in the training phase and consists in 2048 values for both ResNet152 and InceptionV3. This choice will allow us to execute a fair comparison of the two configurations, based only on the extracted values and not influenced by differences in the dimensions.

4.1.2 Abstractness Scoring Reference Datasets

In regards to the abstractness scoring dataset we are going to use for the evaluation of the different techniques we dedicated our attention again to the two works [31; 47] already presented in chapter 3: $D_{Brysbaert}$ and $D_{Rabinovich}$.

A couple consideration about their scores distribution and our necessities are needed here to explain the choices made by us.

First of all, considering the words representation, in term of features, introduced above, we need all the element of our abstractness dataset to be represented by both textual and visual features. At this regard the limitation imposed by the usage of fastText word embedding is the most stringent constraint, because forcing us to filter out a portion of any reference dataset we choose. This consideration has been done under the assumption that the images necessary for the visual features are going to be available for most of the words considered. After the

filtering, we obtained exactly 35425 words from $D_{Brysbaert}$ and 36277 from $D_{Rabinovich}$. Figure 10 and Figure 11 provide respectively a graphical representation of the distribution of words over the whole range of scores. The scoring system from $D_{Brysbaert}$ has been converted from the increasing scale of concreteness from 1 to 5 to the increasing abstractness one from 0 to 1. Each bar in the graph is as wide as 1/100 of the whole score range.

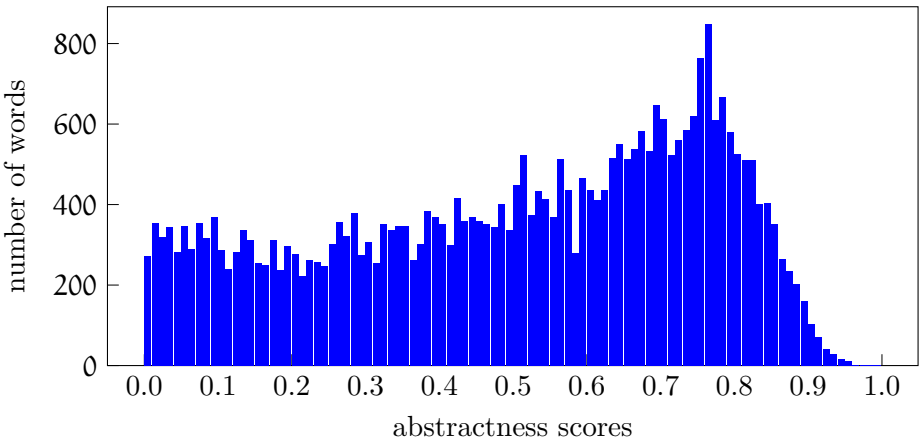

Figure 10: Distribution of the words from $D_{Brysbaert}$ filtered by fastText embedding.

From the resulting distributions, it is possible to notice how each of the dataset is composed by a predominant presence of abstract or concrete words, reflecting pseudo-symmetric arrangements of terms in the score range if compared. Therefore involving both of them would automatically balance the presence of abstract and concrete samples, creating a suitable set of data-points for the tuning of scoring models.

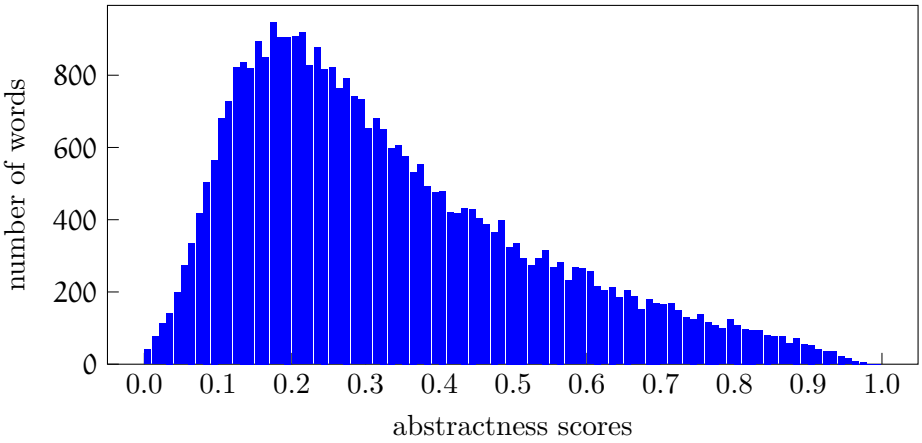

Figure 11: Distribution of the words from $D_{Rabinovich}$ filtered by fastText embedding

The other important factor to keep in mind is that we are going to exploit the dataset selected at this step, not only for the evaluation of several scoring techniques, but most importantly to score the unlabeled words from $D_{Privacy}$. A total of 7077 words is present in the samples selected in the privacy dataset and we want to maximize the chances of providing scores as accurate as possible and to achieve this for both the halves of the scoring range. In order to do so, the more samples are given in the training set, the more are the chances for a new word to present features similar to one of them, achieving this way precise results. Also, using a balanced distribution would avoid to overfit the models towards scoring mostly on one side rather than the other.

After these observations, we expect that using only one of the two dataset would not be satisfying enough in term of general accuracy in the scoring techniques applied to the labeling of our unscored words. We decided therefore to make use of both of them merged.

4.1.3 Selected Abstractness Dataset

The technique we decided to use for the merging of $D_{Brysbaert}$ and $D_{Rabinovich}$ simply consists in:

- the conversion of both scoring schemes to the range of values from 0 to 1 of increasing abstractness;
- assigning to any word belonging to both sets the average of the respective scores.

The choice of using both was necessary here due to the observation brought to the attention of the reader in the preceding subsection. We haven't followed the same choice in the context of the privacy prediction task because of the possible incompatibility of some scores, signaled by the presence of conflicting values associated to terms belonging to both dataset (see chapter 3). In addition to the reasons just mentioned, we believe that one basic aspect justifies the different choices: the use that we are going to make of those scores. In the context of abstractness scoring of words introduced here, the scores from the dataset have been used to train a model tuning a mapping of these numerical values on top of a set of features characterizing words by semantically important features. Therefore, the final result expected from such approach would add a deeper level of complexity to the definition of the scores for the unlabeled word, mitigating the possible errors present in the original datasets. In the privacy task the values would have been used as they are, without tuning on additional type of information. We think that the two dataset would "collaborate" achieving better results if merged, allowing the scoring techniques to take advantage from both the scoring systems, guided by the semantically characterized representation of the input.

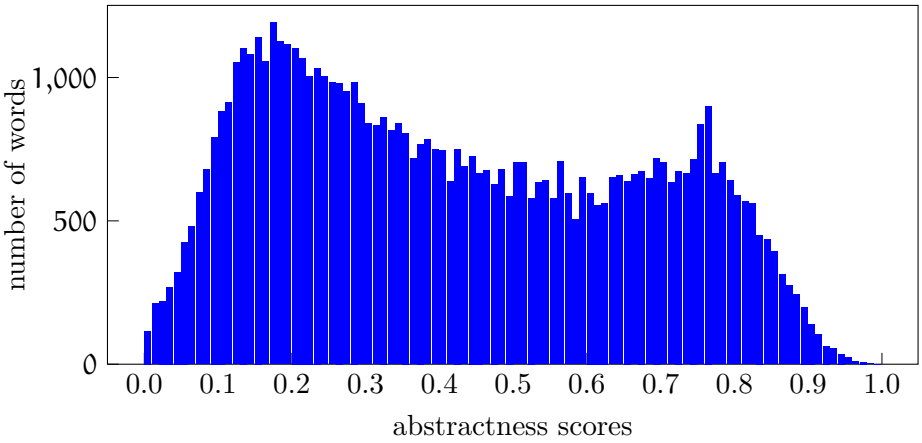

Figure 12: Distribution of the words from $D_{Abstractness}$

We are going to refer from now on to the result of the merging procedure as the abstractness dataset ($D_{Abstractness}$. It is composed by exactly 63877 terms and their distribution is represented in Figure 12.

4.1.3.1 Textual Dataset Resources

As anticipated previously the resource exploited for the representation of words through textual features consists in the vocabulary offered by fastText word embedding [50]. Any reference to textual features in this context is considering fastText encoding.

4.1.3.2 Visual Dataset Resources

Each word has been represented also by visual features. We already defined the modalities we followed for the extraction of the above-mentioned features from single images, but here we will explain how we select them and associate them to the terms involved.

Again, we took inspiration from the approach adopted by [34], where each word have been associated to the averaging of the features extracted by up to 25 images collected through Google images search, using the term itself as keyword for the query.

Our approach is appositely focusing on the specific environment we selected: Flickr social media. In fact we decided to adopt the same averaging technique for the features extracted, but we differed by querying pictures through Flickr API. Specifically for each word we collected a minimum of 5 and a maximum of 25 pictures having the term within their user tags.

We executed the crawling of pictures for both the words in $D_{Abstractness}$ and the unlabeled terms in $D_{Privacy}$, in order to be able to provide a valid representation for each of them. Unfortunately not all the words involved in these two dataset were available as user tags in Flickr in such quantity to respect the constraint of 5 pictures per term. We have been forced, for the visual features approach, to use reduced version of both the reference dataset and the unlabeled set of words, respectively consisting in 47130 and 6689 samples. We are referring to the visual features oriented reduced version of $D_{Abstractness}$ as $D^{Visual}_{Abstractness}$, the distribution of its samples in the range of scores is shown in Figure 13. We can state that the distribution has not been remarkably unbalanced with respect to the original one, keeping the overall disposition.

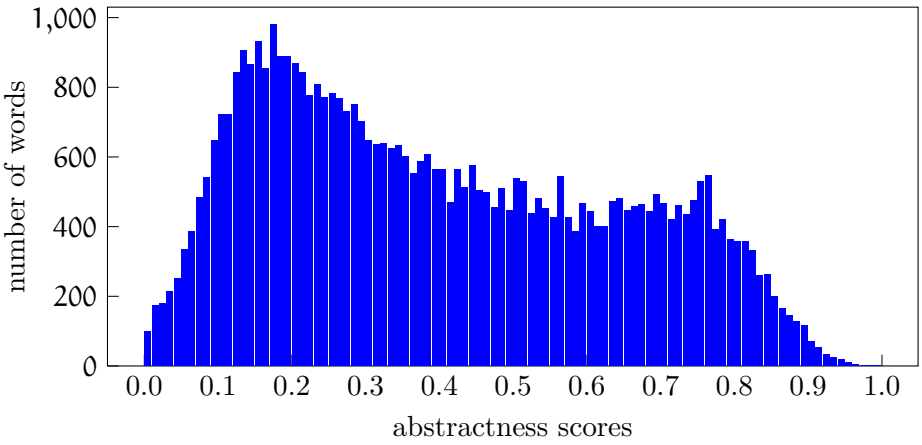

Figure 13: Distribution of the words from $D^{Visual}_{Abstractness}$.

4.2 Scoring Techniques

Now that we have defined the dataset and the features we are going involve in this abstractness prediction task, we are describing the different approaches explored. We have divided them in two categories: distance based and regression approaches

4.2.1 Distance Based Scoring

Two simple typology are included in this category. The main feature of these approaches is the fact that they exploit a metric of distance between the representation of samples. We decided to adopt the cosine distance as measurement of how two data-points are close to each other.

Given two vectors x and y, sharing the same size n, the cosine distance between them is computed as:

$$cosine\_distance = 1 - \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} y_i^2} \tag{4.1}$$

4.2.1.1 Minimum Distance Scoring

This approach is the simplest and it consists in assigning to an unscored sample the same score associated to the closest labeled data-point. We will refer to this technique as *MinDist*.

4.2.1.2 K Nearest Neighbors

This widely known clustering technique has been applied here, adopting different values of K, to select a neighborhood for each unscored word and assigning to it the average of the scores of the neighbors. The neighborhoods are defined as the samples with the smallest cosine distance from the target word. This approach is going to be referred to as *KNN*.

4.2.2 Regression Models

These models are instead widely used methodologies for regression. The typologies experimented are described in the following subsections. For each of them we used the available implementation offered by the Python scikit-learn library [48].

4.2.2.1 Linear Regression

This model is the simplest one for regression. It looks for the linear dependency of the input features and the output value of the training data-points.

It realizes this by minimizing the sum of the squares of the residuals of each sample. More specifically the model define a linear dependence of the output variable and the input features

x. We can define $x_i \in \mathbb{R}^k$ as the set of features of a generic data-point, with $i \in \{1, ..., n\}$, where n is the number of training samples and k the number of features. The linear dependence is expressed by the equation:

$$f(x, \beta) = \beta_0 + \beta_1 x \tag{4.2}$$

where $\beta_0 \in \mathbb{R}$, $\beta_1 \in \mathbb{R}^k$ and $x \in \mathbb{R}^k$ is the set of independent variables. This is the linear function that the linear regression tunes by selecting the best values of β.

The model converges when the linear function produce the minimum value of sum of squared residuals S. The residual for the i^{th} training element is defined as follows for $i \in \{1, ..., n\}$, where y_i is the labeled value for x_i features:

$$r_i = y_i - f(x_i, \beta) \tag{4.3}$$

While the sum of squared residuals as:

$$S = \sum_{i=0}^{n} r_i^2 \tag{4.4}$$

The minimization of S is achieved by gradient descent approach, aiming for a null value for the gradient for each of the input variables.

4.2.2.2 Ridge Regression

The ridge regression model, in particular the version exploiting the kernel trick that we adopted, has a mathematical formulation very similar to the linear regressor just introduced.

The kernel introduce a small difference in the function f and an l2-regularization term is added to the loss function S.

In particular a kernel function $\sigma(x)$ is introduced for mapping the input variables to a new dimensional space, modifying Equation 4.2 into:

$$f(x, \beta) = \beta_0 + \beta_1 \phi(x) \tag{4.5}$$

The formulation of Equation 4.4 for ridge regression is expressed as:

$$S = \sum_{i=0}^{n} r_i^2 + \alpha(\beta_0^2 + \beta_1^T \beta_1) \tag{4.6}$$

The values we evaluated for the regularization parameter α are: 0.001, 0.01, 0.1, 1, 10, 100, 1000. For what concerns the kernels, we tried three variants:

- the linear variant, that results in linear regression with l2-regularization;
- polynomial kernel, evaluating for 2^{nd}, 3^{rd}, 4^{th}, 5^{th}, 6^{th} and 7^{th} degree ;
- radial basis function kernel with γ equal to the inverse of the product of the number of features and the variance of the training set (called "scale" value), "scale"/10 and "scale"·10.

4.2.2.3 Support Vector Regression

This regressor model is based on the same exact mechanism introduced in section 3.2 concerning classification task. The idea behind applying this technique to regression is based on

the goal of finding a hyper-plane that deviates from the training values by a distance no greater than ϵ, for each training point, and at the same time is as flat as possible.

In order to do so the quadratic programming problem is modified as explained below. Consider the training vectors $x_i \in \mathbb{R}^p$, with $i \in \{1, ..., n\}$ and the vector $y \in \mathbb{R}^n$ which contains at the position i the label of training point i. n is the number of training data-points and p is the dimension of each of them. The formulation of the problem is

$$\min_{w,b,\zeta} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \zeta_i + \zeta_i^* \tag{4.7}$$

subject to the following constraints

$$y_i - w^T \phi(x_i) - b \leq \epsilon - \zeta_i, \tag{4.8}$$

$$w^T \phi(x_i) + b - y_i \leq \epsilon - \zeta_i^*, \tag{4.9}$$

$$\zeta_i, \zeta_i^* \geq 0, \text{ for } i \in \{1, ..., n\}$$

The solution finds the values of w, b, ζ_i and ζ_i^* for $i \in \{1, ..., n\}$. The regularization introduced by this model is of l1 type.

The values we evaluated for the C parameter are: 0.001, 0.01, 0.1, 1, 10, 100, 1000. We decided to adopt the suggested 0.1 value for the ϵ parameter instead. For what concerns the kernels, we tried three variants:

- the linear variant, that results in linear regression with l2-regularization;

- polynomial kernel, evaluating for 2^{nd}, 3^{rd}, 4^{th}, 5^{th}, 6^{th} and 7^{th} degree ;
- radial basis function kernel with γ equal to the inverse of the product of the number of features and the variance of the training set (called "scale" value), "scale"/10 and "scale"·10.

4.3 Techniques Evaluation

This section describe the methodologies adopted for the evaluation of the different techniques, concentrating on the metrics used. Then the resulting performances are shown and discussed briefly.

4.3.1 Evaluation Metrics

The metrics adopted for the performance evaluation for the regression models are explained above. In all these definition $\mathbf{x}$ is used to refer to the array of samples' original annotated abstractness score, $\mathbf{y}$ is for the array of samples' predicted scores and D indicated the set of samples that have a score in $\mathbf{x}$ and $\mathbf{y}$. the notation $\bar{\cdot}$ stands for the *mean* operator.

- **Mean Absolute Error** MAE

  Metric for computing the average discrepancy of the value predicted for a sample with respect to the correct one.

$$MAE_{\mathbf{x},\mathbf{y}} = \frac{1}{|D|}\sum_{i=0}^{|D|}|y_i - x_i| \tag{4.10}$$

- **Mean Squared Error** MSE

  Metric for computing the average squared discrepancy of the values predicted with respect to the correct ones.

$$\mathsf{MSE}_{\mathbf{x},\mathbf{y}} = \frac{1}{|D|}\sum_{i=0}^{|D|} |y_i - x_i|^2 \tag{4.11}$$

- **Coefficient of Determination** R^2

  This metric is defined exploiting the following definitions:

$$\mathsf{TSS}_{\mathbf{x}} \text{ (Total Sum of Squares)} = \sum_{i=0}^{|D|} (x_i - \bar{x})^2 \tag{4.12}$$

$$\mathsf{SSE}_{\mathbf{x},\mathbf{y}} \text{ (Sum of Squared Errors)} = \sum_{i=0}^{|D|} (x_i - y_i)^2 \tag{4.13}$$

  TSS value expresses how the data values are far away from the mean value, while SSE indicates how intensely the predictions are different from the original values. We define the coefficient of determination as:

$$R^2_{\mathbf{x},\mathbf{y}} = \frac{\mathsf{TSS}_{\mathbf{x}} - \mathsf{SSE}_{\mathbf{x},\mathbf{y}}}{\mathsf{TSS}_{\mathbf{x}}} \tag{4.14}$$

  The difference at the numerator can be interpreted as the improvement in the prediction given by the regression model, in comparison to a model scoring everything with the mean value. Dividing this quantity by the TSS should simply return a metric of the same improvement, but proportioned. Values close to 1 indicate predictions very close to

reality, while close to 0 indicates a model with performances similar to the mean model. An alternative way of representing this measure in function of the MSE is the following:

$$R^2_{\mathbf{x},\mathbf{y}} = 1 - \frac{MSE_{\mathbf{x},\mathbf{y}}}{MSE_{\bar{\mathbf{x}},\mathbf{x}}} \tag{4.15}$$

Where $\bar{\mathbf{x}}$ is an array of element with the same dimension as $\mathbf{x}$ and $\mathbf{y}$ but having each element as the average value form $\mathbf{x}$. It corresponds to the simplest prediction possible, scoring everything with the average value.

- **Pearson Correlation Coefficient** r

  This metric is used to evaluate intensity and direction for the linear correlation between a pair of variables. Let us see the mathematical formulation:

$$r_{x,y} = \frac{cov(x,y)}{\sigma_x \sigma_y} \tag{4.16}$$

  Where:

  - $cov(x,y) = \sum_{i=0}^{|D|} \frac{(x_i - \bar{x})(y_i - \bar{y})}{|D|}$ is the co-variance between x and y variables;
  - $\sigma_x = \sum_{i=0}^{|D|} \frac{(x_i - \bar{x})^2}{|D|}$ is the variance of variable x, analogously for y

  Rearranging the formula we obtain the following:

$$r_{x,y} = \frac{\sum_{i=0}^{|D|}(x_i - \bar{x})(y_1 - \bar{y})}{\sqrt{\sum_{i=0}^{|D|}(x_i - \bar{x})^2}\sqrt{\sum_{i=0}^{|D|}(y_i - \bar{y})^2}} \tag{4.17}$$

This score can have a value in the range between 1 and -1. The extremes of this scope indicate respectively perfect positive and negative linear correlation between the two variables. On the other hand a r value of 0 stands for the complete absence of linear relationship.

- **Spearman Correlation Coefficient** ρ

  This metric evaluate the rank correlation between two variable. In other words it asses how much the relationship between the variables can be expressed by a monotonic function.

  In order to express it mathematically we need to define rank variables: $rank_x$ is the rank variable of variable x, which simply represents each element of the latter by their rank in the whole set of values.

  We can define the ρ metric as the Pearson Correlation Coefficient of the rank version of the variables:

$$\rho_{x,y} = r_{rank_x, rank_y} = \frac{cov(rank_x, rank_y)}{\sigma_{rank_x}\sigma_{rank_y}} \tag{4.18}$$

  This metric also have values in the range between -1 and 1 and expresses intensity and direction of the monotonicity of their correlation.

We made some consideration regarding the metric to use for the choice of the best performing hyper-parameters of the scoring techniques. Our interest is to train models able to score with acceptable precision on the whole scale of values. Considering the lower amount of training samples in the extremes of the scores range, we expect to obtain a general tendency towards predicting values oriented to the central portion of range. We aim at increasing as possible

the chance to achieve satisfying precision in predicting scores also in the extremes of the scores range, especially for the abstract portion.

The latter consideration lead us to expect higher errors for the fewer elements in the extremes of the range, we need to select a metric that allow us to highly penalize this behaviour during the parameters validation step. We started excluding *MAE* because it assign the same weight to errors, hiding low occurrences of high errors in the averaging process. Regarding $\mathtt{r}$ and ρ, they are respectively evaluating the correlation in terms of linearity and the monotonicity between the prediction and the expected values, but again, high errors for few samples are not penalizing enough these metrics in our opinion. The behaviour of the *MSE* metric instead is exactly targeting the penalization weights we desire, amplifying the value of errors with high discrepancy through squaring the values. The only counterpart is that it is dependant on the range of error values we are dealing with, not giving an absolute definition of what is a good value and what is not.

Our finally decision is to elect the $\mathtt{R}^2$ metric as method to evaluate performances in validation. Referring to its formulation in function of *MSE*, we can obtain the same weighting effect offered by the latter, but evaluated in a proportional way, with respect to the performance of the mean value scoring, which correspond to the simplest model we could think of.

4.3.2 Evaluation Setup

The evaluation of the techniques adopted has been executed through 5-fold cross validation for testing, with hold-out for the validation of the hyper-parameters. It results in:

- 20% for testing;

- 20% for validation of hyper-parameters;
- 60% for training.

As introduced above, the datasets used for this evaluation are $D_{Abstractness}$ and $D^{Visual}_{Abstractness}$ respectively when textual features only and the concatenation of textual and visual features are used.

In the execution of this evaluation all the hyper-parameters introduced with the techniques in the previous section have been experimented and fine-tuned to obtain the best results. This validation step has been evaluated through the R^2 metric. Each regression based scoring method have been applied to both the typology of features introduced.

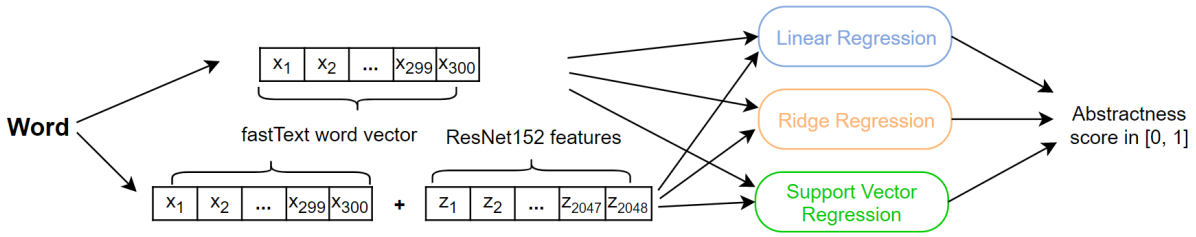

Figure 14: Visual representation of the evaluation of the different features and scoring models.

4.3.3 Performance Results

We are showing here the complete results obtained by the different approaches. In order to avoid useless extensive tables of values we decided avoid presenting any partial results associated

to the validation of most hyper-parameters. On the contrary, in the specific case of regression models exploiting the kernel trick, we opted for showing the different results obtained for each kernel type. We believe this information can be interesting in terms of what type of feature mapping is best for fitting the specific type of representation.

Table IV shows the results obtained using textual features form fastText word embedding. Table V depicts the performances obtained adopting the representation concatenating textual to visual features extracted respectively from ResNet152 and InceptionV3 architectures. The performances using the textual features only has been recomputed and shown here, using the same exact splits, in order to provide a proper comparison.

In the comparison with the usage of visual features, results are showing best performances with the regression models when visual features are involved, in particular the one extracted from ResNet152, while they are degrading when the concatenation is with InceptionV3. Anyways the differences in the scores are very small, changing only after the second decimal digit. For this reason, and for the reduce training sat available for the visual features, we will adopt the fastText representation for the scoring of the unlabeled words.

The most important observation is that, independently from the input representation, any linear model is not performing well, meaning that mapping values onto a new dimensional space is useful. Ridge regression with rbf kernel seem to be the best performing for when using visual features, while the polynomial one achieves best results for textual features. Furthermore the results obtained exploiting residual architecture are better than the one from InceptionV3.

We can highlight that the best performing models are showing high Pearson and Spearman correlation, reflecting respectively a good linearity and monotonicity in the correlation.

Technique	R^2	MAE	MSE	r	ρ
Minimum Distance	53.58	0.1215	0.027	76.13	74.67
K Nearest Neighbors ($k = 12$)	71.28	0.0998	0.0167	85.84	84.28
Linear Regressor	0.6509	0.113	0.0203	0.8196	0.8051
Linear Ridge Regressor	0.6509	0.113	0.0203	0.8196	0.8051
Kernel Ridge Regressor (polynomial kernel)	**0.7406**	**0.0949**	**0.0151**	**0.8677**	**0.8553**
Kernel Ridge Regressor (rbf kernel)	0.7367	0.0956	0.0153	0.8623	0.8503
Linear Support Vector Regressor	0.651	0.1130	0.0203	0.8196	0.8051
Kernel Support Vector Regressor (polynomial kernel)	0.6719	0.1095	0.0191	0.8372	0.8313
Kernel Support Vector Regressor (rbf kernel)	0.717	0.1011	0.0165	0.864	0.8518

TABLE IV: Performances of abstractness scoring techniques using textual features.

4.4 Unlabeled Words Scoring and Results Discussion

From the consideration presented above regarding the performances of the different scoring techniques we selected the best performing hyper-parameters for each technique in relation to both the type of input representations. For what concerns the involvement of the visual features, we decided to analyze here the results obtained by the best performing convolutional neural network architecture, which is the ResNet152 one.

In this portion of the dissertation we are going to discuss the results from both the quantitative point of view of the performances and the qualitative one of the application of the techniques to a real scoring scenario.

Technique	R^2	MAE	MSE	r	ρ
Regression on Text & Visual features (ResNet152)					
Linear Regression	0.5979	0.1179	0.022	0.7910	0.7747
Linear Ridge Regression	0.6211	0.1147	0.0208	0.8069	0.7892
Kernel Ridge Regression (polynomial kernel)	0.7122	0.0977	0.0158	0.8565	0.8404
Kernel Ridge Regression (rbf kernel)	**0.7244**	**0.0956**	**0.0151**	**0.8639**	**0.8473**
Linear Support Vector Regression	0.6233	0.114	0.0206	0.7944	0.7777
Kernel Support Vector Regression (polynomial kernel)	0.6991	0.1009	0.0165	0.8541	0.8372
Kernel Support Vector Regression (rbf kernel)	0.7031	0.1004	0.0163	0.859	0.8418
Regression on Text & Visual features (InceptionV3)					
Linear Regression	0.6041	0.1168	0.0217	0.7949	0.7789
Linear Ridge Regression	0.6241	0.1145	0.0206	0.8115	0.7945
Kernel Ridge Regression (polynomial kernel)	0.6670	**0.1054**	**0.0182**	**0.8297**	**0.8157**
Kernel Ridge Regression (rbf kernel)	**0.6686**	0.1059	**0.0182**	0.8281	0.8156
Linear Support Vector Regression	0.6101	0.1159	0.0214	0.7956	0.7795
Kernel Support Vector Regression (polynomial kernel)	0.6439	0.1104	0.0195	0.8191	0.8074
Kernel Support Vector Regression (rbf kernel)	0.6361	0.1131	0.0199	0.8283	0.8140
Regression on Text features (fastText)					
Linear Regression	0.6311	0.1133	0.0202	0.8151	0.7961
Kernel Ridge Regression (polynomial kernel)	**0.7207**	**0.0961**	**0.0153**	**0.8615**	**0.8452**
Kernel Ridge Regression (rbf kernel)	0.717	0.097	0.0155	0.8545	0.8381
Kernel Support Vector Regression (polynomial kernel)	0.6611	0.1083	0.0186	0.8334	0.8188
Kernel Support Vector Regression (rbf kernel)	0.6857	0.1044	0.0172	0.8563	0.8401

TABLE V: Performances of abstractness scoring techniques using the concatenation of textual and visual features from ResNet152 and InceptionV3 architecture. The performances using textual features only have been reevaluated using the same splits division, for comparison

4.4.1 Results

We are providing here some information about the resulting scoring of the 7077 unlabeled words from $D_{Privacy}$. We scored them using the whole set of words from $D_{Abstractness}$ using the following techniques and respective hyper-parameters. For textual features from word embeddings:

- Minimum distance scoring: no hyper-parameter is needed;

- K nearest neighbors: best performing with $k = 12$;
- Regression model: ridge regressor with polynomial kernel of 5^{th} degree and $\alpha = 0.1$.

For concatenation of textual features and visual ones from ResNet architecture:

- Regression model: ridge regressor with $\alpha = 0.1$, rbf kernel, γ equal to the inverse of the product of the number of features and the variance of the training set.

The resulting distribution of words in the scoring range are shown in Figure 15, while some basic statistics are provided in Table VI.

Extension	Average Score	Concrete Words		Abstract Words	
		< 0.5	< 0.4	≥ 0.5	> 0.6
Minimum Distance	0.3995	4985	4045	2092	1348
K Nearest Neighbors	0.4134	5140	4160	1937	1222
Regression on word embedding	0.3960	5322	4178	1755	972
Regression on concat. with visual features	0.4123	4912	3689	1777	1003

TABLE VI: Statistics on the new scores

From these visual representation of the distributions we can infer the following observations:

- The results from the maximum similarity technique appear to be the most spread on the whole range, counting several samples for almost each of the sub-ranges of values. This behaviour is expected, considering that this technique assign to words the score of the most similar scored one. This approach is increasing the chance of error as we can state

from the performances collected in Table IV, but it is statistically able to target most of the score values available.

- The k-nearest neighbors approach shows a distribution concentrated in the central portion of the scores range. We expected this behaviour, because the fact that each new score has been computed as the average of a set of values influence the distribution to be more dense around the central range. For the same exact reason this method is not good in predicting scores belonging to the outer ranges of values: it is unlikely that the whole neighborhood of a data-point present only extreme scores. This method would work well for the extreme scores decreasing the value of k, causing lower precision for the words belonging to the central portion and therefore an overall worse performance.
- The results from the regression method are also showing a low number of samples with scores in the farthest extremes of the range, but not as low as the previous approach. We believe it is representing an improved interpretation of the correlation between the encoded representation of samples and the scores, supporting this observation through the quantitative better results obtained.
- Analogue observation can be applied to the results using the concatenation with visual features. They present a distribution very similar to the regression applied to word embedding. The quantitative results are showing lower metrics for this case probably because it has been trained with slightly less samples, due to the problem faced in the retrieving of enough images for all words.

Considering the overall pattern in the distribution obtained from the different methods, we can state that they are consistent with each other, all showing a high peak of samples with scores around 0.3, an almost flat profile in the range between 0.5 and 0.7 and few occurrences in the extremes of the whole range. This observation provide some support to the trustworthiness of our results.

4.4.2 Final Conclusion

Our final decision is to adopt the new scores obtained through the regression technique on the textual features. We decided not to use the one produced with visual features because the number of words it can score is limited, providing a lower number of extension samples, but mainly because the real limitation is in the available training set. The latter could strongly effect the performances. The similarity in the distribution obtained by all approaches can be interpreted as a good reason to believe that the results are trustworthy enough for being exploited in the next task.

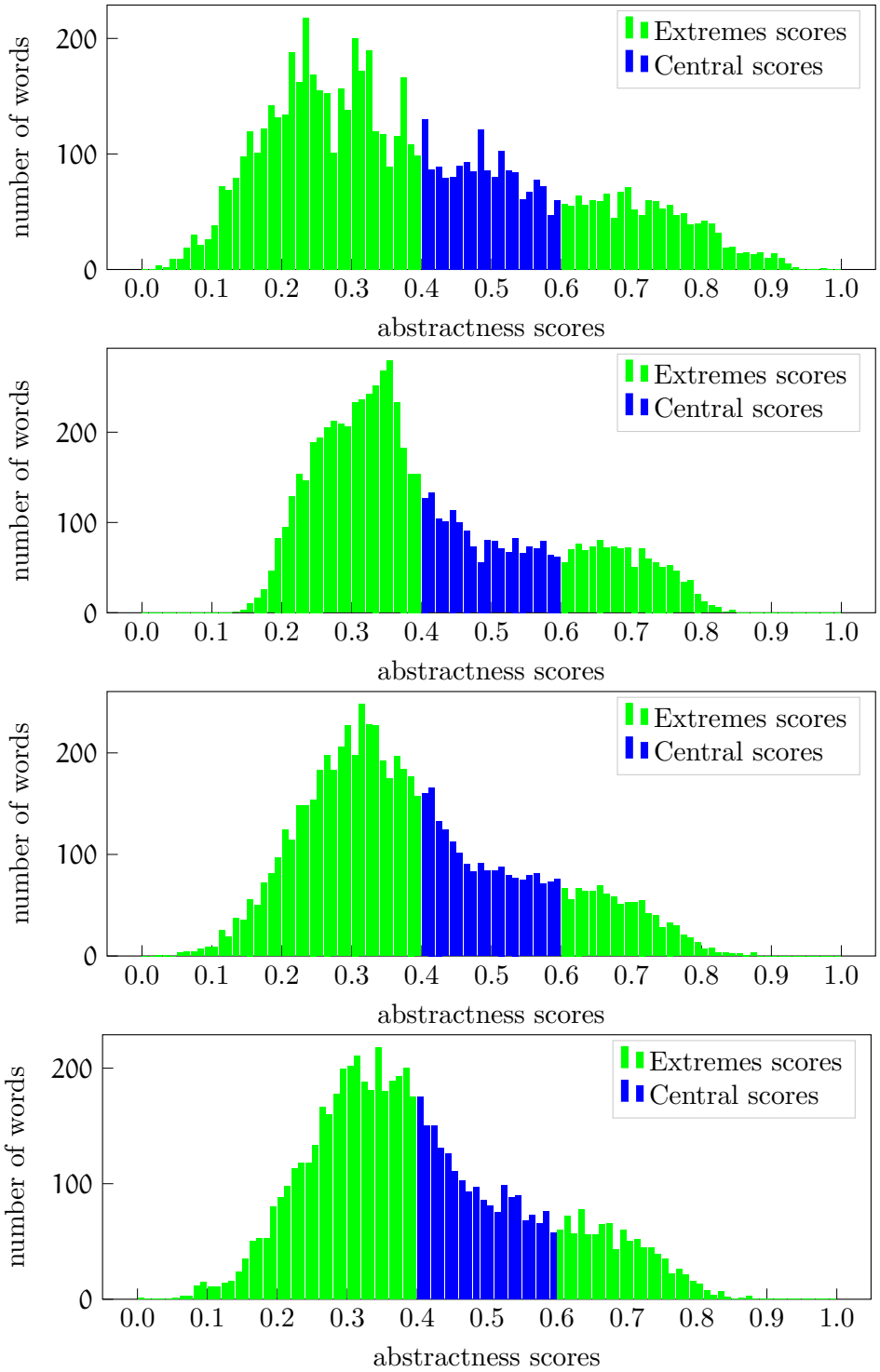

Figure 15: Distribution of new scores predicted for the unlabeled words from $D_{Privacy}$. The four images refer respectively from top to bottom the minimum distance technique, the k nearest neighbors and the regression applied to the textual features, the last one is the result of regression on textual and visual features concatenation.

CHAPTER 5

EXPERIMENTS

This chapter of our thesis work focuses on the description of the reasons, the methodologies and the execution of the different experiments tailored appositely to test our thesis.

As introduced several times in the previous chapter, the final purpose of this dissertation is to investigate how the abstractness of words, user tags associated to pictures in social media, is correlated to the private nature of the content shared. We would like to show that the usage of abstract concepts is better in representing the samples in a classification task such as the binary privacy prediction one, especially in the comparison with terms of concrete disposition.

We are presenting firstly the evaluation setup we decided to adopt for the evaluation of the singular classification models, describing precisely the dataset used and the validation process.

Secondly we introduce a detailed review of the approaches adopted for the experiments. The whole experimental setup is structured in a twofold manner and has been executed in two main steps. Here we are going to describe its organization in detail, providing the reasons behind each choice.

5.1 Evaluation Setup

We are going to apply the same exact setup for the evaluation of each of the experiments that we are going to introduce.

First of all, as anticipated previously we are going to use $D_{Privacy}$ as the online privacy labeled dataset. For evaluating the abstractness nature of the words we will exploit $D_{Abstractness}$ scores, expanding it to cover also the unscored words from $D_{Privacy}$ in the second part of the experimentation.

Each experimental setup involves the evaluation of the four different classification models largely described in section 3.2. In particular we applied a 5-fold stratified cross-validation technique for testing, with hold-out for validating the hyper-parameters. This approach is applied to each model, evaluating the specific hyper-parameters listed in their description (section 3.2). The resulting proportions of the sizes of the different sets used for training, test and validation are respectively 60%, 20% and 20%. Each model has been evaluated using the same exact splits subdivision.

In order to avoid skewed results, due to the unbalance in the population of the private and public classes, the private samples of each training set have been oversampled, so that the same number of data-points per class, even if repeated, is perfectly equal.

The metrics we are going to evaluate in terms of performances are the ones commonly adopted in the task of binary classification. In particular we adopted the accuracy measure for validating the best performing hyper-parameters, and also the f1-score of both the private and public class for the evaluation of the results on the test sets.

5.2 The Experimental Setup

In this section all the details related to the dual characterization of the experiments' setup are explained.

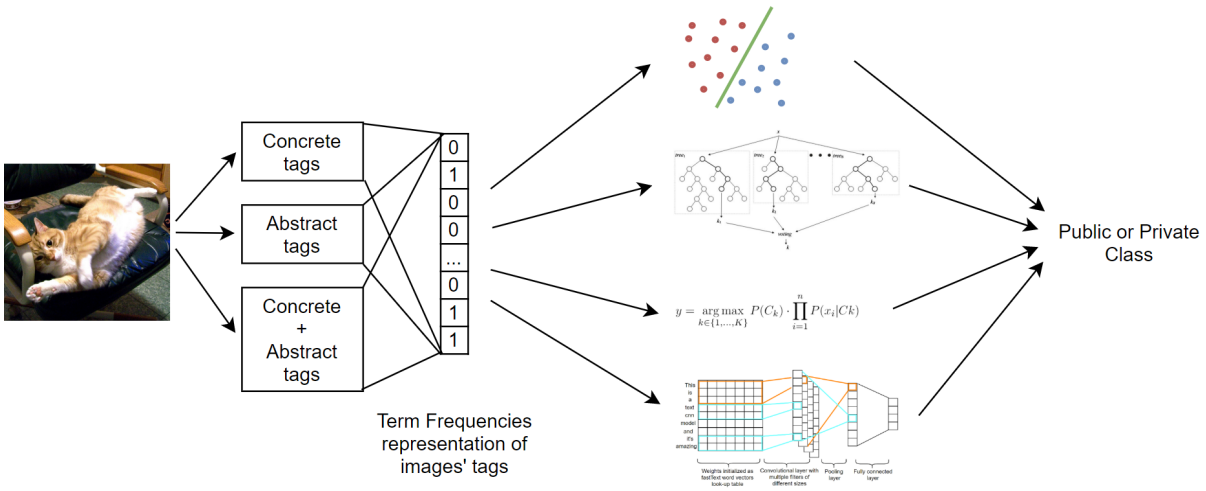

Figure 16: Visual representation the evaluation of the different classifiers for privacy.

5.2.1 Abstract vs Concrete

The first basic information we need to extract from our results comes from the comparison of the results obtained by abstract and concrete words. For this reason each of the experiments described is going to be evaluated by the performances obtained separately by each of the two types of terms, es well as the ones gathered by using both together.

Basically each sample has been used three times in each experiment: represented by abstract words only, by concrete words only and by both typology together. The constraints imposed in the selection of $D_{Privacy}$ have been selected for this exact reason, allowing this way to have a minimum of two abstract terms and two concrete ones representing each data-point.

We believe that this lower bound to the quantity of words of each type is able to provide us with a diverse enough set of samples, that will enable us to capture the difference in privacy discriminating power of the two categories of concepts.

5.2.2 Manipulation of User Tags Distribution Over Samples

The secondary aspect of the investigation is instead focusing on evaluating the abstract and concrete categories, keeping in consideration the intense diversity and width of their distribution among the privacy dataset.

5.2.2.1 Natural Tags Distribution

The first experiment type is simply keeping the distribution of tags over the samples unchanged. Both the type and quantity of the words associated to each sample has been left as it is originally.

This setup aims at observing the differences in performances using concrete or abstract words, reflecting the original configuration and distribution of the words. Therefore any unbalance in the quantity of the tags in each sample or in the overall dataset has been maintained.

The statistics gathered from the dataset we are using, show a largely higher presence of concrete tags, thus we ideally expect the classification model employed to perform better using that type, due to the low presence of the abstract counterpart. We are referring to this experiment setup as *Natural-Distribution*.

5.2.2.2 Tags Presence Equally Balanced by Category

The second setup instead aims at balancing the presence of abstract and concrete tags in the context of each singular sample. In other words we tried to obtain an equal amount of abstract and concrete tags associated to each data-point.

The approach followed here consisted in selecting a specific integer value N, and randomly sampling the tags of each sample in the following way, where A and C are respectively the sets of abstract and concrete word of the specimen sample:

- A number of tags equal to the minimum between N, $|\mathsf{A}|$ and $|\mathsf{C}|$ is randomly selected from A and used as the new set of abstract tags;
- A number of tags equal to the minimum between N, $|\mathsf{A}|$ and $|\mathsf{C}|$ is randomly selected from C and used as the new set of concrete tags.

The balance created for each sample enables this setup to show the actual discriminating power of each tag category. It creates the condition where the descriptive potential of the two types for each sample is completely equal. Another interesting aspect, that this type of experiment will show us, is the evolution of the performances along the increase of the N parameter. We will refer to this setup as the N-*Sample-Balanced*.

A slightly modified version of this setup has been also explored. In this variant we have simply avoided the parameter dependency of the result, removing N in the definition. Therefore for each sample:

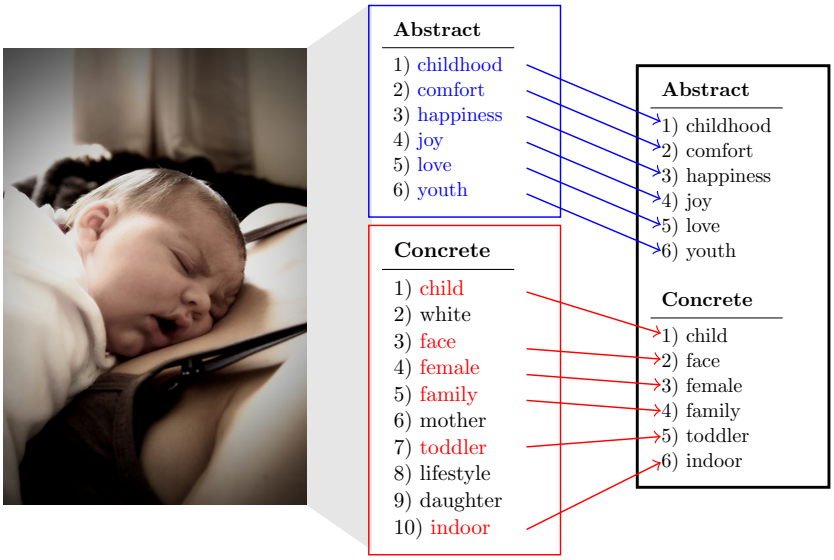

Figure 17: Visual representation of how tags selection is performed in *Max-Sample-Balanced*

- A number of tags equal to the minimum between $|A|$ and $|C|$ is randomly selected from A and used as the new set of abstract tags;
- A number of tags equal to the minimum between $|A|$ and $|C|$ is randomly selected from C and used as the new set of concrete tags.

The result here is the maximization of the equal contribution of information from both abstract and concrete categories. We expect to extract the most insightful results through this setup, because not only it creates the balance just introduced, but it also exploits the tags availability to the fullest. We are referring to this specific type of experiment as *Max-Sample-Balanced*.

5.3 Extending the Abstractness Scored Dataset

This second portion of the experimentation involves the extension of the set of abstractness scored word by labeling the 7077 unscored tags present in $D_{Privacy}$. Specific experiment setups have been executed exploiting this extension, to confirm the insights obtained using the original abstractness dataset and to get further insights.

5.3.1 Abstractness Scoring Extension Choice

The extension of the scored words dataset has been done using the best scoring technique resulted from the evaluation described in chapter 4. The discussion expressed at the end of the above-mentioned chapter concluded that the best performing approach is the ridge regression model, with polynomial kernel of 5^{th} degree, using the textual features from fastText word embedding for the samples representation. This decision has been confirmed by the quantitative considerations about the metrics values of the results. Moreover, the fact that using textual features, rather than the concatenation with visual ones, allows us to score more terms and more precisely, due to the low availability of images on Flickr for some specific terms (see chapter 4).

The results obtained in the scoring process have been largely described in the previous chapter about the abstractness scoring task. Particularly helpful are the information gathered in Table VI and Figure 15.

A notable insight about the resulting distribution is that concrete tags are present in higher quantity with respect to the abstract ones. Specifically we are dealing with three times more concrete words than abstract. Considering only the terms in the extremes of the scoring range,

Class	Pics	Abstract Tags		Concrete Tags	
		per picture	Tot	per picture	Tot
Private	1072	5.95	1279	13.16	3974
Public	1853	6.17	2270	15.50	7110
All	2925	6.09	2668	14.64	8644

TABLE VII: $D_{Privacy}$ dataset statistics using the complete extension.

which corresponds to values above 0.6 and below 0.4, the unbalance is higher, reaching the ratio of four to one.

5.3.2 Exploiting the Complete Extension

Initially we decided to execute the same exact experiment setups adopted for the original abstractness scoring dataset, but extending it with the complete set of 7077 newly labeled terms. Specifically the two *Natural-Distribution* and *Max-Sample-Balanced* experiments have been evaluated.

The usage of the extension allows us to evaluate our hypothesis without the influence of removing from the consideration a large portion of words. We had in fact to ignore the set of 7077 unscored terms, because unable to define their nature in term of abstractness. The automatic scoring technique allow us to use all of them, excluding from the evaluation only the unscored tags not representable through fastText word embedding.

Some statistics about the resulting privacy dataset, using the complete extension, are shown in Table VII. The interesting insights that we can get from this setup concern the fact that the

difference in the amount of concrete and abstract has increased, as well as the average amount of concrete words per sample.

5.3.3 Exploiting the Extremes of the Extension

The second approach we experimented evaluate the performances exploiting only a portion of the automatically scored words. The selection of the samples to use aims at exploiting a subset of terms for which we have higher confidence about their nature.

More specifically, we started considering that in the context of the privacy classification experiments we are using words in a binary way, defining them as abstract or concrete on the base of the scores associated. Therefore we decided to keep in our extension only the tags which have been strongly characterized towards one of the two extremes of the scoring range. Specifically we defined as belonging the extremes of the range any word with a score below 0.4 and above 0.6. This choice has been tailored in order to keep a large enough amount of words from both sides.

Class	Pics	Abstract Tags		Concrete Tags	
		per picture	Tot	per picture	Tot
Private	1072	4.61	917	11.34	3423
Public	1853	4.50	1597	13.71	6177
All	2925	4.54	1885	12.84	7500

TABLE VIII: $D_{Privacy}$ dataset statistics using the extremes of the extension.

The words selected from the extension has been added in the consideration of the experiments and the *Natural-Distribution* and *Max-Sample-Balanced* setups have been executed with the new asset. The statistics related to the version of $D_{Privacy}$ extended this way are shown in Table VIII.

CHAPTER 6

RESULTS AND DISCUSSION

This chapter is dedicated to the presentation of the performance results obtained from the execution of the experimental setup and adopting the evaluation modalities as discussed in chapter 5.

The dissertation is divided in two portion, one related to the outcomes of the experiments using the original $D_{Abstractness}$ and the following regarding the abstractness dataset extension through automatic scoring.

In conclusion we propose a discussion of the results in general, focusing on the comparison of the overall performances.

We are going to analyze the results by the point of view of three metrics: accuracy and f1-measure for both the privacy classes. The accuracy should give us a broader idea about the general proportion of correct predictions among the complete set of samples. F1-scores will provide us some insights about the general performances of the models in relation to each class, its value represent a unique evaluation of both precision and recall for the class targeted, allowing us to understand if the model is skewed towards one of the two, or if it is labeling samples in a balanced way.

6.1 Results with Original Abstractness Dataset

The performances using $D_{Privacy}$, with $D_{Abstractness}$ for terms abstractness evaluation and adopting the evaluation procedures and metrics introduced in 5.1, are shown in Table IX.

We would start commenting these results by highlighting a general performance pattern, easily noticeable by a quick review of the table. Precisely, considering the comparison between the metrics values obtained using abstract only tags and concrete ones, all models performed generally better in the former case. We decided to start from this consideration because it represents an important achievement, consisting by itself in a strong proof in support of our thesis.

Here follows a list of observation we can derive from the resulting performances:

- *Natural-Distribution* setup unexpectedly is showing slightly better results using abstract words rather than concrete. The accuracy metric is showing an average discrepancy oscillating around 1.5% between the usage of the two types of tags.

  It is important to notice that even though the quantity of different concrete tags in each single sample and in the overall dataset is certainly higher with respect to the abstract counterpart, the latter is still performing better. The higher variety and presence of concrete words should suggest higher probabilities, for this category of terms, to be better suited for characterizing the samples in the privacy prediction task. On the contrary the evidence emerged from this initial result are proving the opposite.

- *Max-Sample-Balanced* experiment further supports our thesis showing an important increase in the performances gap compared with the natural distribution. In this case each

of the metric, in the evaluation of each of the models, is stating that abstract words are better performing. More specifically they are reaching an average 5% more in accuracy and a minimum of 3% more in f1-scores, for both classes, when compared with concrete words performances.

It is important to observe from the values reported by the different metrics, how it is emerging a general trend of achieving similar performances when using abstract tags only or in combination with concrete ones. Therefore the contribution of information brought by concrete concepts, when available in the same quantity as the abstract counterpart, seems to be not very meaningful for privacy prediction.

From this setup we can derive that the discriminating power of concrete words here is certainly enhanced by the extremely high presence they have in each single sample. This experiment reduced their presence, keeping it at the same level as for abstract words, consequently showing important decreases in performances.

- N-*Sample-Balanced* experiments, with the different values of N from 1 to 4, have been evaluated in order to understand if the considerations derived from previous results are still valid when the information available for each samples is extremely limited. These setup aim at evaluating the comparison of performances of concrete and abstract concepts at the core of their essence to verify if the discrepancies in their discriminating power is still present.

  With a gradual increase in the value of N we have been able to monitor the performances step by step. Ensuring the balance in the number of abstract and concrete tags for each

sample, we started from the setup with the minimum possible tags, to the one maximizing it. The differences in performances between concrete and abstract have been verified by each model and for all values of N.

The fact itself that this pattern is repeated for each of the four values of the parameter suggests us that it is not a behaviour influenced by the particular randomly sampled terms. Therefore we can use it to prove that the better performance appreciated with abstract concepts in *Max-Sample-Balanced* is not caused by the sub-sampling action itself, but is related to actual distribution of words among samples and their abstract nature

Experiment Setup	Tags Type	SVM			NB			RF			CNN		
		Acc. (%)	Pri. F1 (%)	Pub. F1 (%)	Acc. (%)	Pri. F1 (%)	Pub. F1 (%)	Acc. (%)	Pri. F1 (%)	Pub. F1 (%)	Acc. (%)	Pri. F1 (%)	Pub. F1 (%)
Natural-distribution	A	68.0	73.75	73.83	69.26	64.06	73.17	72.55	77.42	77.44	72.82	63.77	78.22
	C	66.84	73.97	74.0	67.52	60.86	72.26	69.06	73.76	73.90	73.06	62.24	79.04
	A+C	**71.15**	**77.6**	**77.63**	**73.43**	**67.72**	**77.46**	**75.79**	**80.27**	**80.36**	**76.55**	**67.05**	**81.74**
Max-Sample-Balanced	A	**67.22**	73.01	73.04	69.23	63.96	73.17	71.93	76.49	76.54	72.51	63.53	77.91
	C	62.29	70.11	70.18	62.05	55.37	67.00	64.99	72.46	72.53	66.46	53.96	73.62
	A+C	66.74	**73.83**	**73.86**	**70.19**	**64.82**	**74.15**	**74.05**	**78.86**	**78.93**	**74.09**	**64.52**	**79.57**
1-Sample-Balanced	A	59.14	62.57	62.64	57.81	54.97	60.32	67.14	74.3	74.34	65.84	56.44	71.87
	C	51.21	49.24	49.37	52.76	51.37	53.82	62.19	71.92	71.98	60.48	51.65	66.56
	A+C	**60.51**	**66.94**	**67.10**	**61.51**	**56.47**	**65.50**	**67.86**	**74.91**	**74.92**	**67.93**	**57.75**	**74.12**
2-Sample-Balanced	A	**65.88**	71.25	71.29	67.66	62.53	71.56	70.6	76.08	76.11	70.32	62.59	75.4
	C	57.71	64.22	64.36	56.31	51.91	59.97	62.77	72.06	72.08	63.25	52.24	70.11
	A+C	64.68	**71.83**	**71.84**	**68.79**	**63.14**	**72.94**	**71.18**	**76.92**	**76.93**	**72.89**	**63.18**	**78.53**
3-Sample-Balanced	A	**66.70**	72.23	72.24	68.96	63.39	73.06	71.11	75.92	75.95	72.27	64.59	77.2
	C	59.08	66.39	66.53	61.71	54.3	67.05	63.62	71.47	71.48	65.81	54.82	72.45
	A+C	65.54	**72.76**	**72.77**	**70.32**	**64.84**	**74.34**	**72.92**	**77.83**	**77.86**	**74.02**	**65.1**	**79.29**
4-Sample-Balanced	A	66.80	72.63	72.67	68.78	63.03	73.01	71.62	76.63	76.68	71.86	63.51	77.07
	C	60.96	68.3	68.35	62.8	56.77	67.36	65.13	72.69	72.71	67.35	57.16	73.59
	A+C	**67.25**	**74.13**	**74.15**	**71.49**	**66.18**	**75.37**	**73.91**	**78.71**	**78.75**	**73.91**	**64.51**	**79.32**

TABLE IX: Results of the privacy classification experiments on $D_{Privacy}$ using abstractness scores from $D_{Abstractness}$

Experiment Setup	Tags Type	SVM			NB			RF			CNN		
		Acc. (%)	Pri. F1 (%)	Pub. F1 (%)	Acc. (%)	Pri. F1 (%)	Pub. F1 (%)	Acc. (%)	Pri. F1 (%)	Pub. F1 (%)	Acc. (%)	Pri. F1 (%)	Pub. F1 (%)
Complete Extension Natural-Distribution	A	68.24	74.69	74.71	69.16	63.25	73.44	73.06	78.18	78.19	74.02	63.22	79.90
	C	70.63	76.66	76.71	71.62	65.05	76.13	73.78	79.45	79.50	75.73	64.93	81.41
	A+C	**70.67**	**76.67**	**76.70**	**74.6**	**68.45**	**78.76**	**76.65**	**81.24**	**81.31**	**76.92**	**67.45**	**82.11**
Complete Extension Max-Sample-Balanced	A	67.76	74.26	74.28	68.62	62.93	72.79	72.48	77.62	77.64	73.4	63.01	79.21
	C	64.55	71.28	71.33	64.75	58.98	69.11	68.75	75.79	75.84	69.13	56.89	75.93
	A+C	**69.54**	**76.29**	**76.32**	**71.08**	**65.61**	**75.05**	**74.46**	**79.47**	**79.50**	**75.86**	**65.70**	**81.36**
Extension Extremes Natural-Distribution	A	68.96	75.32	75.37	68.68	62.98	72.87	72.31	77.35	77.39	73.98	64.64	79.41
	C	69.09	75.76	75.81	70.25	63.28	75.02	72.34	77.83	77.89	74.26	63.26	80.16
	A+C	**70.22**	**76.42**	**76.47**	**73.67**	**67.34**	**77.97**	**76.17**	**80.94**	**81.00**	**77.13**	**67.10**	**82.45**
Extension Extremes Max-Sample-Balanced	A	68.34	74.80	74.83	68.89	63.33	72.99	72.48	77.33	77.36	73.61	64.34	79.04
	C	62.77	70.82	70.94	62.77	57.15	67.09	66.74	74.15	74.18	67.86	54.49	75.16
	A+C	**68.61**	**75.42**	**75.44**	**71.35**	**65.75**	**75.39**	**73.91**	**78.86**	**78.88**	**75.52**	**65.57**	**80.99**

TABLE X: Results of the privacy classification experiments on $D_{Privacy}$ using abstractness scores from $D_{Abstractness}$ and the extension of automatically scored ones.

6.2 Results with Abstractness Dataset Extension

We are now focusing our attention toward the outcomes observed in the experiment setups exploiting the extension of the abstractness scored dataset. Once again the evaluation has been performed according to the methodologies described in chapter 5.1. The performances obtained are shown in Table X.

Here follows a list of observation on the performances achieved:

- The results obtained from the *Natural-Distribution* experiment using the *complete extension* is reflecting very similar results from the usage of abstract and concrete only words. In general concrete tags are performing slightly better. The main consideration that we can derive from this setup is related to the amount of tags added to the consideration by the extension. The new distribution of words among samples has kept the same ratio

between the number of concrete and abstract terms, also in the two privacy classes, but the amounts have been tripled. This is surely increasing the performances for both categories of words, as verified, but the concrete one is gaining the most advantages from it. This behaviour can be justified by the fact that the large variety of tags added enables the concrete ones to be descriptive enough for the samples, to the point of performing better than the abstract counterpart. This hypothesis is also supported by the fact that the average amount of tags per data-point is increased by almost 6 for the concrete class and only 3 for the abstract one.

- The *Max-Sample-Balanced* setup with *complete extension* is showing better performances with abstract tags. With respect to the results obtained without the extension, it is possible to notice a slight decrease of the gap in accuracy, reduced now to 4%, while the f1-scores have kept on the average the same discrepancy. These results seem to confirm the observation derived from the previous experiment, demonstrating that even with the greater amount of concrete results from the extension, once the number of words per category is equal, the samples are better classified through the abstract tags.
- The setup with *Natural-Distribution* using the *extension extremes* is also providing interesting insights. In comparison to the setup using the complete set of newly scored words, it is increasing a little the ratio of quantity of concrete words with respect to abstract. Considering the average number of words per sample of each type, it also seems to give a slightly higher contribute to the concrete category increasing the statistic by around 3.5 for the latter and only 1.3 for the abstract. The effect in performances is very similar to

the other experiment, but the small gap in the metrics value appear to be even smaller. The main observation that we can derive from this behaviour is that the increased precision in the abstractness scores added by this setup, in spite of creating again an even higher unbalance in the distribution of tags in favor of the concrete category, is decreasing the performance discrepancy. Our hypothesis is that this behaviour is caused by the higher precision of the extension's scores, but we can hardly confirm that due to the very subtle performance's changes with respect to the setup using the whole extension

- The last setup, with *Max-Sample-Balanced* using the *extension extremes*, is proving the previous hypothesis in regards of the scores precision. In fact, comparing the performances with the same setup using the complete extension, we can detect a slight, but shared, performances increase for abstract words and decrease for concrete ones. This behaviour is increasing the gap between the results of the two categories by adding a lower amount of words, but with higher confidence about their abstract nature.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

From the results obtained by this investigation we are now able to infer that abstractness is closely related to privacy, when we talk about characterizing the images through their user tags. The experiments allowed us to evaluate how abstract words are better representing images for privacy, if compared with concrete, especially when the representation of the samples is reduced to one or two terms. We also tested how extending the quantity of abstract and concrete information associated to the images influences the privacy prediction, once again verifying that the abstract terms, even if considerably lower in amount, are obtaining performances comparable to the concrete counterpart. Lastly we also noticed that it is very important to use precise scoring for abstractness, evaluating how this is influencing the results of the privacy prediction.

With regards of the task of scoring words by abstractness, we have been able to test the performances of representing them through word vectors or a concatenation of textual and visual features. We concluded that they perform very similarly in this task, adopting the models chosen, and it is important to mention that the features extracted by ResNet152 architecture are better performing than the InceptionV3 ones. We couldn't compare the results with other works, because the dataset used by us has never been tried before. Through the qualitative analysis of the usage of the scoring models on unlabeled words from our privacy dataset we have been able to appreciate the importance of using a symmetrically distributed dataset. This

feature of the training set allowed us to obtain scores not too unbalanced on one specific side of the range, resulting in plausible values.

One important statement, regarding the abstractness related task in general, is that it is important to enlarge the available datasets on the topic. Another issue regards the preciseness of these data resources. The problem of judging abstractness resulted, by itself, difficult to be faced objectively, often producing consideration that are not shared my everybody. Particularly hard is to score concepts that are not extremely characterized toward one of the two extremes of the scoring range. Even using a precise definition, both dataset we analyzed showed discrepancies, suggesting that the ability of precisely recognizing the abstractness of concepts is very hard to achieve, and could require the choice of a particular annotation setup or a new definition, focusing on slightly different aspects of the idea of abstract.

This research offers a good starting point for further analysis about abstractness and privacy correlation. Future works should investigate the results using other type of textual data associated to online images, such as titles, descriptions and comments. A particularly interesting direction of this research would be to explore new ways to extract abstract information from images, task that has been hardly approached by researchers. This investigation could reach fascinating insights about how complex model would be able to detach from the mere level of the images' pixels, extracting concepts of elevated nature from it.

CITED LITERATURE

1. Xu, H., Wang, H., and Stavrou, A.: Privacy risk assessment on online photos. In RAID, pages 427–447, 2015.

2. Henne, B., Szongott, C., and Smith, M.: Snapme if you can: Privacy threats of other peoples' geo-tagged media and what we can do about it. WiSec '13, 2013.

3. Simpson, A.: On the need for user-defined fine-grained access control policies for social networking applications. In Proceedings of the Workshop on Security in Opportunistic and SOCial Networks, SOSOC '08, pages 1:1–1:8. ACM, 2008.

4. Ghazinour, K., Matwin, S., and Sokolova, M.: Monitoring and recommending privacy settings in social networks. In Proceedings of the Joint EDBT/ICDT 2013 Workshops, EDBT '13, pages 164–168, New York, NY, USA, 2013. ACM.

5. Ilia, P., Polakis, I., Athanasopoulos, E., Maggi, F., and Ioannidis, S.: Face/off: Preventing privacy leakage from photos in social networks. In Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security, CCS '15, pages 781–792, New York, NY, USA, 2015. ACM.

6. Song, X., Wang, X., Nie, L., He, X., Chen, Z., and Liu, W.: A personal privacy preserving framework: I let you know who can see what. In SIGIR, pages 295–304. ACM, 2018.

7. Gross, R. and Acquisti, A.: Information revelation and privacy in online social networks. In Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society, pages 71–80, 2005.

8. Jones, S. and O'Neill, E.: Contextual dynamics of group-based sharing decisions. CHI '11, 2011.

9. Besmer, A. and Lipford, H.: Tagged photos: concerns, perceptions, and protections. In CHI '09, 2009.

CITED LITERATURE (continued)

10. Ahern, S., Eckles, D., Good, N. S., King, S., Naaman, M., and Nair, R.: Over-exposed?: privacy patterns and considerations in online and mobile photo sharing. In CHI '07, 2007.

11. Tran, L., Kong, D., Jin, H., and Liu, J.: Privacy-cnh: A framework to detect photo privacy with convolutional neural network using hierarchical features. In AAAI '16, 2016.

12. Tonge, A. and Caragea, C.: Image privacy prediction using deep features. In AAAI, 2016.

13. Tonge, A. and Caragea, C.: On the use of "deep" features for online image sharing. In Companion Proceedings of The Web Conf., pages 1317–1321, 2018.

14. Kuang, Z., Li, Z., Lin, D., and Fan, J.: Automatic privacy prediction to accelerate social image sharing. In Third IEEE International Conference on Multimedia Big Data, BigMM, pages 197–200, 2017.

15. Yu, J., Kuang, Z., Yu, Z., Lin, D., and Fan, J.: Privacy setting recommendation for image sharing. In 16th IEEE International Conference on Machine Learning and Applications, ICMLA 2017, pages 726–730, 2017.

16. Yu, J., Kuang, Z., Zhang, B., Zhang, W., Lin, D., and Fan, J.: Leveraging content sensitiveness and user trustworthiness to recommend fine-grained privacy settings for social image sharing. IEEE Trans. Information Forensics and Security, 13(5):1317–1332, 2018.

17. Spyromitros-Xioufis, E., Papadopoulos, S., Popescu, A., and Kompatsiaris, Y.: Personalized privacy-aware image classification. In ICMR '16, pages 71–78, New York, NY, USA, 2016. ACM.

18. Zhong, H., Squicciarini, A., Miller, D., and Caragea, C.: A group-based personalized model for image privacy classification and labeling. In Proceedings of the 26th International Joint Conf. on Artificial Intelligence, pages 3952–3958, 2017.

19. Orekondy, T., Schiele, B., and Fritz, M.: Towards a visual privacy advisor: Understanding and predicting privacy risks in images. In IEEE International Conference on Computer Vision, ICCV 2017, pages 3706–3715, 2017.

20. Squicciarini, A., Novelli, A., Lin, D., Caragea, C., and Zhong, H.: From tag to protect: A tag-driven policyrecommender system for image sharing. In PST '17, 2017.

CITED LITERATURE (continued)

21. Squicciarini, A., Lin, D., Karumanchi, S., and DeSisto, N.: Automatic social group organization and privacy management. In 8th International Conference on Collaborative Computing: Networking, Applications and Worksharing, pages 89–96, Oct 2012.

22. Zerr, S., Siersdorfer, S., Hare, J., and Demidova, E.: Privacy-aware image classification and search. In Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '12, pages 35–44, New York, NY, USA, 2012. ACM.

23. Vyas, N., Squicciarini, A., Chang, C.-C., and Yao, D.: Towards automatic privacy management in web 2.0 with semantic analysis on annotations. In CollaborateCom, pages 1–10, 2009.

24. Sundaram, H., Xie, L., De Choudhury, M., Lin, Y.-R., and Natsev, A.: Multimedia semantics: Interactions between content and community. Proceedings of the IEEE, 100(9):2737–2758, 2012.

25. Paivio, A.: Mental imagery in associative learning and memory. 1969.

26. Paivio, A.: Dual coding theory, word abstractness, and emotion: a critical review of kousta et al. (2011). Journal of experimental psychology. General, 142 1:282–7, 2013.

27. Schwanenflugel, P. J. and Noyes, C. R.: Context availability and the development of word reading skill. Journal of Literacy Research, 28(1):35–54, 1996.

28. Paivio, A., Yuille, J. C., and Madigan, S. A.: Concreteness, imagery, and meaningfulness values for 925 nouns. Journal of Experimental Psychology, 76(1, Pt.2):1–25, 1968.

29. Spreen, O. and Schulz, R. W.: Parameters of abstraction, meaningfulness, and pronunciability for 329 nouns. Journal of Verbal Learning and Verbal Behavior, 5(5):459 – 468, 1966.

30. Coltheart, M.: The mrc psycholinguistic database. The Quarterly Journal of Experimental Psychology Section A, 33(4):497–505, 1981.

31. Brysbaert, M., Beth Warriner, A., and Kuperman, V.: Concreteness ratings for 40 thousand generally known english word lemmas. Behavior research methods, 46, 10 2013.

32. Paetzold, G. and Specia, L.: Inferring psycholinguistic properties of words. In Proceedings of the 2016 Conference of the North American Chapter of the Association for

CITED LITERATURE (continued)

Computational Linguistics: Human Language Technologies, pages 435–440, San Diego, California, June 2016. Association for Computational Linguistics.

33. Feng, S., Cai, Z., Crossley, S. A., and McNamara, D. S.: Simulating human ratings on word concreteness. In FLAIRS Conference, 2011.

34. Bhaskar, S. A., Köper, M., Schulte Im Walde, S., and Frassinelli, D.: Exploring multi-modal Text+Image models to distinguish between abstract and concrete nouns. In Proceedings of the IWCS workshop on Foundations of Situated and Multimodal Communication, 2017.

35. Rabinovich, E., Sznajder, B., Spector, A., Shnayderman, I., Aharonov, R., Konopnicki, D., and Slonim, N.: Learning concept abstractness using weak supervision. CoRR, abs/1809.01285, 2018.

36. Tsvetkov, Y., Boytsov, L., Gershman, A., Nyberg, E., and Dyer, C.: Metaphor detection with cross-lingual model transfer. In ACL, 2014.

37. Turney, P., Neuman, Y., Assaf, D., and Cohen, Y.: Literal and metaphorical sense identification through concrete and abstract context. pages 680–690, 01 2011.

38. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J.: Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems 26, eds. C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, pages 3111–3119. Curran Associates, Inc., 2013.

39. Pennington, J., Socher, R., and Manning, C. D.: Glove: Global vectors for word representation. In In EMNLP, 2014.

40. Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., and Joulin, A.: Advances in pre-training distributed word representations. In Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018), 2018.

41. Mikolov, T., Chen, K., Corrado, G. S., and Dean, J.: Efficient estimation of word representations in vector space. CoRR, abs/1301.3781, 2013.

42. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L.: ImageNet Large Scale

CITED LITERATURE (continued)

Visual Recognition Challenge. International Journal of Computer Vision (IJCV), 115(3):211–252, 2015.

43. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In CVPR09, 2009.

44. He, K., Zhang, X., Ren, S., and Sun, J.: Deep residual learning for image recognition. CoRR, abs/1512.03385, 2015.

45. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. E., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A.: Going deeper with convolutions. CoRR, abs/1409.4842, 2014.

46. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z.: Rethinking the inception architecture for computer vision. CoRR, abs/1512.00567, 2015.

47. Rabinovich, E., Sznajder, B., Spector, A., Shnayderman, I., Aharonov, R., Konopnicki, D., and Slonim, N.: Learning concept abstractness using weak supervision. In EMNLP, 2018.

48. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12:2825–2830, 2011.

49. Kim, Y.: Convolutional neural networks for sentence classification. CoRR, abs/1408.5882, 2014.

50. Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T.: Bag of tricks for efficient text classification. CoRR, abs/1607.01759, 2016.

51. Charbonnier, J. and Wartena, C.: Predicting word concreteness and imagery. In IWCS, 2019.

52. Krizhevsky, A., Sutskever, I., and Hinton, G. E.: Imagenet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems 25, eds. F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, pages 1097–1105. Curran Associates, Inc., 2012.

VITA

NAME Gabriele Galfre'

EDUCATION

Bachelor of Science in Computer Engineering
Politecnico di Torino, Turin, Italy.
2017

Master of Science in Software Engineering (current)
Politecnico di Torino, Torino, Italy.
2019

Master of Science in Computer Science (current)
University of Illinois at Chicago, Chicago, IL, U.S.A.
2019

EXPERIENCES

Research Assistant in Data Science
January 2019 - September 2019
University of Illinois at Chicago, Chicago, IL, U.S.A.
Worked on the research of the master's thesis on the topic of privacy classification in social media and text abstractness scoring models.

Teaching Assistant
March 2017 - July 2017
Politecnico di Torino, Turin, Italy
Assistant to the professor in a "linear controls" course.

Research Intern
March 2017 - May 2017
Istituto Superiore Mario Boella, Turin, Italy
Worked on the initial stages of a European project about a framework for the optimization of the behavior of groups of cyber physical systems.