

Sociocultural Influences for Password Definition: An AI-based Study

Carlos Ocanto Dávila^{a,*}, Rocío Cabrera Lozoya and Slim Trabelsi

Security Research, SAP Labs France, Mougins, France

Keywords: Password Cracking, Password Security, Probabilistic Context Free Grammars, Semantic Clustering.

Abstract: Most of the research that analyses password security has been developed targeting English-speaking users. In this work, we present a framework for password segmentation, semantic classification, and clustering, in a multilingual context. This research uses natural language processing, statistical and deep learning techniques to obtain and leverage semantic patterns for password definition. Using the methods proposed in this work in password-guessing models produce over a 10% increase with respect to state-of-the-art methods (with a guessing space limited to 500 million predictions) on a dataset of leaked credentials.

1 INTRODUCTION

Passwords have been used since ancient times as a tool in military, commerce, and private life. To date, they remain the primary authentication method despite their known flaws which include human's difficulty for remembering complex patterns, user engagement, storage and encryption and susceptibility to attacks (Morris and Thompson, 1979). They are not likely to be replaced soon as alternatives, such as biometric or device authentication and trust scores, are still not feasible for massive use. Even though passwords have been used for a long time in networked systems, there seems to be little understanding of their composition. The patterns people use to create them are very important to generate adequate security policies, assess security metrics, and suggest best-practice modifications to users (Weir et al., 2009). Currently, the community's understanding of password structures is limited to superficial patterns and a deeper understanding is key to address their security challenges.


In this work, we analyse these issues. Using SAP's SWAM tool, we collected a dataset composed of publicly available leaks from different internet services which contain information about users from several countries. We provide insights into the most relevant patterns that users display when creating passwords and highlight their dependencies to the language of the subject or the source of the leak. Also, to further analyze the presence of certain key categories in the passwords, we created a language-

dependent system that labels the category of a word (based on a set of dictionaries, a lexical database and semantic categories found through word embeddings created using unsupervised learning). Additionally, we propose a Probabilistic Context Free Grammar (PCFG) model that leverages these language and cultural dependent labels to produce password guessing attempts. We compare the results to those from available password strength meters, such as Dropbox's *zxcvbn* (Wheeler, 2016).

To this end, the research objectives of this work are: First, to analyze the language and source-dependent characteristics of password composition in our database. We then construct a pipeline to generate semantic categories relevant to the password sources using specialized lists, lexical tools and unsupervised word embeddings in a multilingual context. Finally, we assess the impact of using semantic categories in password cracking attempts with a PCFG model and compare these results with available password strength meters.

2 RELATED WORK

Password choice is largely dependent on users' cultural background, such as their language and origin (Maoneke et al., 2018) (Wang et al., 2019) (Mori et al., 2019). Most of the body of research in password analysis has been devoted to predominantly English-speaking sources, although recent attention has been given to other languages (Wang et al., 2019) (Mori et al., 2019) (Maoneke et al., 2018). Most work

^a  <https://orcid.org/0000-0001-6606-0784>

*Work done during internship.

in the field focuses on the presence of numeric, alphabetic, or special characters in the passwords but some works have begun to use dictionaries (Wang et al., 2019), or lexical databases (Veras et al., 2014) to generate semantic categories. Still, work remains to be done to generalize these categories to out-of-vocabulary words and to apply these concepts in a multilingual context.

While certain works provide insights into the relationship between password safety and the user's native language (Wang et al., 2019) (Maoneke et al., 2018), there is currently no consensus on how to measure password strength. Most assessments use word entropy or the number letters, digits and symbols but these patterns have proven to be predictable (Golla and Dürmuth, 2018) (Wheeler, 2016). Other tools, like Dropbox's zxcvb (Wheeler, 2016), provide alternative measurements based on a series of heuristics that simulate the number of tries required to crack a password. It also provides feedback on password alterations to improve its security. Other approaches include using predefined dictionaries, password blacklists, or assessments such as Markov models (Ma et al., 2014) and Probabilistic Context Free Grammars (PCFGs) (Weir et al., 2009). These last two require significant computational resources and are state-of-the-art methods for password cracking/guessing (we refer to guessing as trawling guessing). Some works involve deep learning techniques which require more training data and are usually less insightful about the patterns that produce weak passwords (Hitaj et al., 2017) (Melicher et al., 2016). Nonetheless, (Hranický et al., 2020) presented an online method for character-level vulnerability analysis for passwords, thanks to a data-driven approach based on PCFGs and deep learning. On the other hand, PCFGs provide a framework for targeted attacks (Wang et al., 2016) which signifies a security risk in case of leaked credentials.

In the remainder of this paper, we will describe our contributions to password security assessment by adapting statistical and deep learning methods.

3 DATASET

A series of leaked credentials were obtained using SAP's SWAM platform, a cyber threat monitoring tool which collects samples from several sources and serves for security research purposes. We collected over 600.000 samples, coming from 17 unique sources, including: *Amazon, Crunchyroll, Ebay, Facebook, Fornite, Gamming site, Hulu, McDonads, Minecraft, Netflix, Spotify, NordVPN, and Wish.com*. Likewise, the dataset contains users from 164 differ-

ent countries. Nonetheless, most of the data did not have any metadata (e.g. leak source, user information). We infer the language of the password either through metadata provided in the leak (e.g. country of origin of the account) or through the suffix of the email linked to the account. We exclude emails with suffixes like ".com", ".net", or ".edu". We focus our study in three main European languages (English, French and Portuguese), which amount to 120.000 samples and we contrast the password characteristics depending on their origin.

Given that these leaks contain personal information about the users, for security and ethical reasons, this dataset cannot be made publicly available for the moment.

4 SEMANTIC CATEGORY GENERATION

In this work, we exploit semantic structures in passwords to aid cracking systems generate better guesses. In this section, we present the different techniques we used to obtain insights of the main passwords patterns chosen by users. We start by introducing password base structures obtained by statistical analysis. We then present a series of created and mined specialized lists that were deemed relevant for this problem. Next, we explore the use of traditional NLP tools, such as lexical dictionaries, and more recent techniques, such as word embeddings. It should be noted that these processes are language dependent; one needs to have a prior of what distribution to use, which we have obtained from the dataset's metadata.

4.1 Password Base Structures

Most of the related works in literature find words in passwords using a naïve strategy in which the vocabulary of the text is found by separating the letters, digits and special characters, e.g. "charles.1994" would have "L6 S1 D4" as base structures. Next, the letter sequence, interpreted as a word, is compared to a predefined dictionary. However, this approach is unable to properly account for multiple words in a sequence, e.g. "goodpassword.1994". We propose a method to optimize the segmentation of a password by including into our vocabulary all the words inside the letter strings in a password.

In the context of password cracking, (Veras et al., 2014) use a set of English corpora as the ground-truth of the language word distribution to then optimize the

coverage of the n-grams¹, according to its frequency information. We follow a similar approach in a multi-language context, selecting the n-grams that maximizes the likelihood of the partition, taking as ground-truth the Zipf's law distribution of a given language.

Zipf's law (Zipf, 1949) is an empirical formulation which uses a power law distribution to model the rank of a word in a language. This law implies that the most frequent word in language will appear roughly twice more than the second most common, and three times the third most common, until a given threshold in which the law breaks for uncommon words (given that the harmonic series diverges). The frequency of a word is inversely proportional to its statistical rank r :

$$P = \frac{1}{r \cdot \ln(1.78R)} \quad (1)$$

Where r is the rank of a word, and R the size of the vocabulary. It is largely agreed this is a good empirical approximation for languages (Yu et al., 2018) (Turner et al., 2014). In our approach, we use it to model the best partition of a given text. We use the data from the Python library *wordfreq* (Speer et al., 2018) as ground-truth on our experiments. This multilingual corpus contains data from sources such as Wikipedia, Twitter, Books, and Movies, which seem to provide a good basis to model the true distribution of a language. We perform analysis in the languages with the largest number of samples in our data and *wordfreq*.

Assuming that the words are independently distributed, and modeling the language with a Zipf's Law before optimizing the segmentation of the words, we develop an algorithm to obtain a more complete representation of vocabulary in the dataset. First, we separate the string into basic base structures (letters, symbols, and digits). In each of the alphabetic sections, we apply a dynamic programming algorithm that outputs the best partition of the array, according to the Zipf distribution of that array. The most likely sentence is the one that maximizes the product of the probabilities of each candidate word, after evaluating all the possible partitions. We base our method on (Generic Human, 2012), where the logarithm of the inverse of the probabilities is minimized. There can be some ambiguity in the segmentation, which will favor the most common words. The independence assumption limits the capacity of the algorithm to model languages in a more general manner, but it is an approximation that correctly enriches our vocabulary.

¹N-grams are subdivisions of size N in a text, with N being the number of characters in the section. For instance, the N-grams of the word "Password", with N=4, are: "pass", "assw", "sswo", "swor", "word"

This approach is generalizable to multiple languages, which enables us to analyze sources that previous studies have not explored. In the examples of Table 1, the segmentation for a relatively long "password" with multiple words, in different languages, can be properly split into its most likely components, if one has the right prior language.

4.2 Specialized Lists

Like (Veras et al., 2014) we use a set of specialized lists in order to classify some of the words in our corpus. We extend some of the categories they use, for a more extensive categorization, using a multilingual approach. We collect a comprehensive set of lists specific for each language in our corpus (English, French, Portuguese) either through governmental sources, interest-specific online sources or through the authors' personal experience. In our experiments, the words detected after the segmentation step are first fed to a program that detects Keyboard walks (e.g. qwerty, azerty, asdfg), long sequence of repeating n-grams (e.g. lalala, qqqq, asdasd) and specific numeric constructions (e.g. date formats with years in the 1900-2100 range). If the words are not part of these categories, they are labelled using the mentioned set of specialized lists. Some of the collections we used include language-specific stop words from Python's *NLTK*. Because proper names seem to be a recurring source of inspiration for passwords, we include lists with names and terms that could stem from personal relationships (most common given names for males, females and pets) as well as affectionate or love terms (e.g. love, honey, etc), nouns associated to a person (e.g. man, woman, boy, friend, pal) and noble ranks and titles (e.g. lord, queen or knight), all in a multilingual context. We cover pop and media culture with lists including the most popular sports, athletes (active and retired), sports teams and celebrities in social media platforms. We also used the 250 most popular movies of all time according to IMDB and a list of superheroes, anime and video games. Finally, a list of religious and profanity terms as well as a list with regional information (i.e. names of countries, continents and cities with over 200.000 inhabitants) was included. In total, we collect over 17.000 distinct base structures in our dataset.

Nevertheless, this approach has a limited reach, as gathering a set of lists to label all the words in the corpus is unfeasible. Indeed, between 20 and 30% of the vocabulary is composed of alphabetic strings with an undefined category. In the following section, we describe how we use lexical dictionaries and un-

Table 1: Example segmentations found on different languages.

	Input Password	Resulting Segmentation	Cost
EN	Longersentencescanbestudiedtoo	Longer sentences can be studied too	49.42
FR	Desphrasespluslonguespeuventaussiêtréétudiées	Des phrases plus longues peuvent aussi être étudiées	65.58
PT	Frasesmaislongastambempodemserestudadas	Frases mais longas tambem podem ser estudadas	64.84

supervised NLP techniques to address the remaining corpus.

4.3 Lexical Dictionaries

Wordnet is a lexical dictionary originally developed by Princeton (Princeton University, 2010) which ties together word’s concepts in a graph representation. This tool labels words’ meaning under structures called synsets. Synsets are organized into four classes: verb, adverb, adjective and noun. Then, the structure of the words would be composed by a name, a type and a number, therefore having a structure such as *dog.n.01*, for the synset corresponding to the first sense of dog used as a noun. Wordnet groups words from the same category (e.g. car, automobile, auto and autocar) and in a hierarchical structure (i.e. from general to specific concepts). Synsets higher in the hierarchy with respect to a word are called hypernyms, while hyponyms are the synsets located at a lower point in the hierarchy. For instance, the synset *dog.n.01* can have the following hierarchical path: *entity.n.01*, *physical_entity.n.01*, *object.n.01*, *whole.n.02*, *living_thing.n.01*, *organism.n.01*, *animal.n.01*, *chordate.n.01*, *vertebrate.n.01*, *mammal.n.01*, *placental.n.01*, *carnivore.n.01*, *canine.n.02*, *dog.n.01*

Some studies, like (Veras et al., 2014), have used Part-of-Speech tagging (POS-tagging) techniques to disambiguate the syntactical purpose of the word. Nonetheless, we found that POS taggers for the different languages often led to unreliable results. Therefore, we do a semantic classification for Wordnet by taking the most common synset of a word without specifying its POS type.

Following the idea used to develop the specialized lists in the previous section, we aim to create semantic categories for the labeled words using WordNet. Nonetheless, it is not trivial to find a representation for a word that is not too general or specific. The authors of (Veras et al., 2014), developed a tree-cut model for the synsets found for the English language. Nonetheless, upon manual inspection, we observed that their resulting synsets were not generalizable enough to be considered as semantic categories in the context of this work. We instead perform a clustering of the words found using Wordnet. We propose to use the Wu-Palmer (WUP) similarity metric between the found synsets in Wordnet. The WUP is a normalized metric determined by the depth of two synsets in the

graph, and the depth of their most specific ancestor node (Least Common Subsumer). To create the feature space expressing the synset relations present in Wordnet, we compute the WUP similarity for every combination of synset in the corpus. This results in a $n \times n$ similarity matrix that represents the semantical relations of the words found in WordNet.

4.4 Word Embeddings

Fasttext (Bojanowski et al., 2016) is a framework to represent word embeddings as an aggregation of their n-gram level embeddings instead of computing them from the words itself. Fasttext has pre-trained models are available online in different languages, including English, French, and Portuguese. We also explore the relevance of Bert, a state-of-the-art language model developed by Google (Devlin et al., 2018). In contrast with Fasttext, Bert uses context-dependent embeddings. This means that the vector of a word would be different based on its context, whereas models such as Fasttext obtain static word embeddings; their vector representation will be unique, and context-independent. The use of context-dependent representations can be a great tool to get a richer understanding of the function of a word in a text as language can be ambiguous. For the English language, we make use of the pre-trained, large version of the model, uncased with whole word masking. For the French language, we use the model developed by Inria and Facebook called Camembert (Martin et al., 2020).

4.5 Semantic Categories from Lexical Dictionaries and Word Embeddings

Figure 1 shows the two-dimensional projection after using T-SNE manifold reduction on the word feature spaces obtained from the embedding models (Fasttext and Bert) and the WUP similarity between synsets. Clusters obtained from the lexical similarity feature space (from Wordnet) appear to have larger intra-cluster similarity and inter-cluster separation, with less overlap and noise.

When using a hierarchical clustering model optimized for the Silhouette coefficient (Rousseeuw, 1987) and the Calinski Harabasz (CH) (Caliński and JA, 1974) index metrics (both specific to the unsupervised learning context), we obtain the results in Table

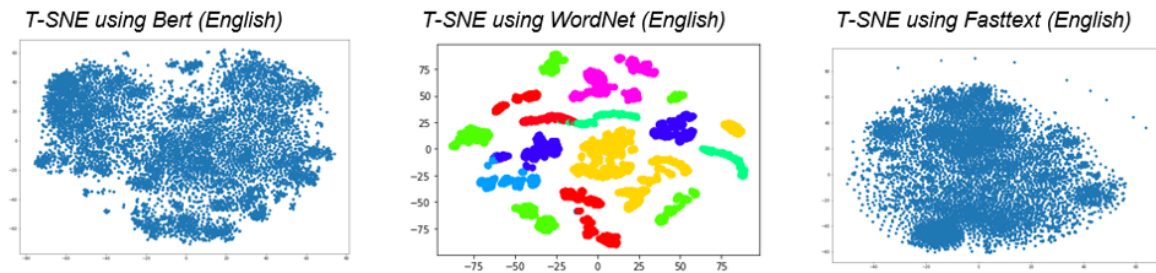


Figure 1: 2-dimensional T-SNE of the Bert, WordNet and Fasttext feature spaces.

Table 2: Metrics for the found clusters of each feature space.

	# of Clusters	Silhouette Score	CH Index
English			
WordNet	15	0.34	4585.77
FastText	10	-0.04	161.84
Bert	9	0.04	299.35
French			
WordNet	18	0.37	2245.03
FastText	10	-0.05	75.24
Camembert	11	-0.02	121.49
Portuguese			
WordNet	18	0.41	1703.94
FastText	11	-0.07	51.72

2. Here it is clear that the quality of the clusters seems to be superior in the WordNet space.

Nonetheless, WordNet’s drawback is its relatively low coverage for the vocabulary in the dataset. The vocabulary covered by WordNet on the unlabeled data was of 62.19% for English, 51.53% in Portuguese, and 49.79% in French. This poses additional challenges, as the rest of the vocabulary must be labeled in a naïve manner. To account for this limitation, we grouped together strings based on their length to address this issue.

From a more empirical point of view, we observe that the categories obtained using the WordNet space are more likely to be grouped in semantically similar categories. Examples of the clusters found in French include: *animals* (viper, ape, chat, poussin, chien,...), *substances* (whisky, sand, vodka, fur,...) and *regions* (Togo, Cracovie, Acadie, Versailles, ...).

Clusters found using Fasttext and Bert embeddings appear to follow morphological similarities by clustering abbreviations, words in foreign languages, and small affectionate terms: *affectionate nouns* (lolo, coucou, titi, juju, vero, fifi, nath, jojo, caro, zouzou,...), *abbreviations/slang* (ouch, lol, janv, nov, tte, km, jr...), *words in foreign language- English* (Cat, One, Top, Fire, Big, Red, Did, And, Funny, Killer,...), *words in foreign language- Germanic* (der, das sie, wie, wenn, wij, vor, von, ook,...)

In the following section we will assess the influence of these clusters, along with the previously constructed specialized lists, for password security.

5 ASSESSING PASSWORD SECURITY

Inspired by (Weir et al., 2009), we use a Probabilistic Context Free Grammar (PCFG) model to learn the syntactic and semantic characteristics of a password vocabulary exploiting the previously defined semantic categories. Intuitively, the use of semantic categories reduces the search space the guesser needs to explore to find the most probable passwords. PCFGs produce competitive results while using a relatively small number of samples and have shown to benefit from semantic categories. These properties are not found in similar deep learning models, which motivates the use of this statistical learning method. A generic PCFG is composed of:

- A set of terminals Σ formed by the vocabulary of the Corpus: $\Sigma = w_1, w_2, \dots, w_v$
- A set of non-terminals, or variables V , which are the syntactic categories of the grammar.
- A star symbol N_1
- A set of rules, $N_i \rightarrow \zeta_j$, where ζ_j is a sequence of terminals or non-terminals. And follows: $\sum_j P(N_i \rightarrow \zeta_j) = 1, \forall i$

In the context of password security, a PCFG first splits the elements in a string into similar groups. For instance, under the first implementation of this method (Weir et al., 2009) the password “seville!1994” will be divided into the letter segment L “seville” of length 7, a symbol segment D “!” of length 1, and a digit segment D “1994” of length 4. Providing the base structure $\zeta_i = L_7, S_1, D_4$, which will have a probability:

$$P(\zeta_i) = \frac{\text{Occurrences of } \zeta_i}{\text{Occurrences of all base structures } \zeta_n} \quad (2)$$

Then the probability of a guess will be equal to the product of the production (non-terminal base structures and terminals). Thus, for this example the probability of “seville!1994”, which we will denote as $P(\zeta_a)$ is:

$$P(\zeta_a) = P(\zeta_i)P(w_{L_x})P(w_{S_y})P(w_{D_z}) \quad (3)$$

$$P(\zeta_a) = P(L_7, S_1, D_4)P(L_7 \rightarrow "seville")P(S_1 \rightarrow "!")P(D_4 \rightarrow "1994") \quad (4)$$

With $(w_{L_x}) \cup (w_{S_y}) \cup (w_{D_z}) = \zeta_i$

In our approach, instead of assuming the naïve letter, symbol and digit, we use the semantic categories obtained with specialized lists and with unsupervised learning methods, as we described in the previous section. These classes served to build the base structures and compute the terminal probabilities. Moreover, there were some practical considerations and assumptions we considered, some proper of the model, others designed.

Individual Transitions Do Not Impact Each Other.

By using a PCFG, one assumes the probabilities in the base structures are independent. In reality, there can be dependencies between the terminals, but this simplification has proven to be effective while keeping the complexity of the model and the computational resources necessary low.

The Transitions of the Grammar Are Non-ambiguous. Every terminal value will be associated with only one rule. Sometimes repeated guesses are produced because of different evaluations of the two base structures might coincide in its final string. Nonetheless, this accounts for less of 0.5% of the total in our experiments.

Recursion Is Not Allowed. The graph will follow a hierarchical structure with no jumps back (which are allowed in PCFGs). For instance, $rule_a \rightarrow rule_b; rule_b \rightarrow rule_a$ is not allowed.

Rules Are Expressed in Lowercase. Naïve implementations of PCFG interpret “Cat”, “cat”, and “CAT” as different terminals. We group terminal values according to their lower-case version as mangling rules can be trivially added in later stages (Veras et al., 2014).

In order to keep the number of guesses (as well as computational times and memory resources) to a minimum, we opted for an algorithm that parses the generated tree using an efficient strategy, as opposed to producing all possible combinations in each base structure. To generate guesses, we replicate the “next” algorithm developed by Weir (Weir et al., 2009). It uses priority queues to progressively add candidate guesses and achieves the goals of avoiding duplicates by parsing the tree once, generating parses in probability order, and minimizing time and memory re-

Table 3: Percentage of the passwords guessed by the models.

PCFG	English	French	Portuguese
Fasttext-clusters	48.47%	41.07%	43.29%
Wordnet-clusters	48.14%	39.85%	43.07%
Bert-based	45.53%	40.97%	-

quirements. This results in lopsided trees that avoid duplicate nodes.

Despite these optimizations, the process is very resource-intensive: using PCFGs to generate 500 million guesses requires around 30 hours of computations, outputs a guessing file of 5 GBs, and requires to have in memory an increasingly large priority queue (approximately 60 million samples). Further alternatives to optimize this algorithm (e.g. implementing it in a low-level programming language) are out of the scope of this work.

6 RESULTS AND DISCUSSION

We use the categories found in the previous section as non-terminal rules of the model to evaluate their impact and compare it to the naïve implementation of PCFG (taking only character level information about the passwords), and Weir’s publicly available tool (an improved version of his original PCFG proposal) (Weir, 2017). Moreover, we conduct these experiments on the English, Portuguese, and French sources we labeled using WordNet, Fasttext, and Bert-like models. We perform a stratified train-test split of 80-20% by selecting for testing, similarly ranked passwords than in the training set.

In Table 3, the performance of the models trained by our method can be observed. It is seen how most of the models have a guessing rate of over 40% of the test data. Table 4 summarizes the improvements found for the Fasttext Embeddings models, which is common across the worked languages, where it is clear that our proposed semantic categories found using specialized lists and unsupervised learning represent an improvement over other comparable methods.

The available data for English was roughly twice that of French and Portuguese, which is partially responsible for the higher performances achieved. Nonetheless, it is remarkable that the multilingual pipeline to address the different semantic categories have produced a valid PCFG model, that outperforms Weir’s tool, and classic PCFGs.

In Figure 2, we can observe how the different models behave in time, for the English language. Portuguese and French have similar behaviors. At the point of the final evaluation (500 million guesses),

Table 4: Summary of the Improvements of the Fasttext PCFG over the other tested tools.

Language	Model	Improvement (%)
English	original PCFG	34.03
English	Weir’s tool	10.53
French	original PCFG	30.08
French	Weir’s tool	2.3
Portuguese	original PCFG	71.49
Portuguese	Weir’s tool	10.11

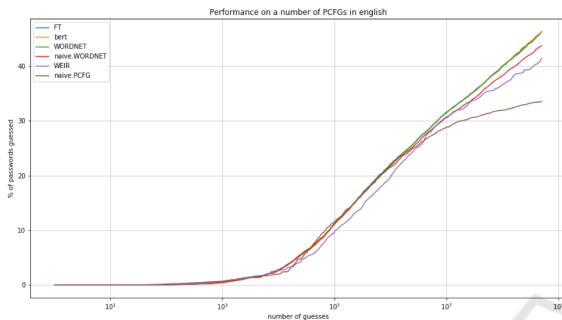


Figure 2: Performance of the tested models for the English language.

the proposed models do not seem to reach the saturation point. Nonetheless, continuing with the guesses until a higher dimensional space was not possible due to time and memory constraints. It can also be seen in Figure 2 how the use of simple synsets as semantic categories produce results that also outperform the other PCFG tools. Grouping these clusters into optimized semantic categories proves to be beneficial for the overall performance of the model. However, the simpler PCFG version has a performance comparable to those of the semantic guesser, up to 1.000.000 guesses. This could indicate their validity for a quicker, less resource-intensive cracking session, where the hacker is satisfied with a lower guessing accuracy.

It is interesting to compare these findings with Dropbox’s *zxcvbn* tool for measuring password strength. We found that the tool was “pessimistic”, as it indicated that the number of guesses that it would take a model to guess a password was inferior to what we found in our experiments. The number of times it overestimated the strength of the password was smaller than the number of times that it underestimated it. Excluding the passwords that were not guessed by our model, we found that the number of attempts required to crack a password was smaller 37.48% for English, 18.84% for French, and 30.05% for Portuguese. These assessments might be more accurate under a context where there is more vocabulary to learn from. In Table 5, it can be observed that the number of tries required by our algorithm and Drop-

Table 5: Correlation between the log10 of the required guesses for the passwords in French, English and Portuguese, and the log10 estimated by Dropbox’s *zxcvbn* tool.

Model	English	French	Portuguese
Fasttext	0.68	0.48	0.22
WordNet	0.67	0.46	0.22
Weir	0.57	0.32	0.3
Naive PCFG	0.66	0.6	0.49

box’s tool are correlated.

7 CONCLUSIONS

We performed an empirical analysis over a set of different languages and sources that helps characterize passwords. We analyzed some of the main structural and semantic patterns proper of the origin of the leak. Factors such as user language and interests, as implied from the sources of the leaks (e.g. gaming or anime sites), influence the choice of passwords. In our experiments, we reaffirm the remarks in (Veras et al., 2014) that the usage of semantic categories improves password guessing algorithms.

We successfully propose a new method for semantic classification in different languages that provide insights into the intricacies of password composition, while highlighting some of the challenges of obtaining these categories. We extend PCFG structures to include syntactical and semantical categories not used before (like pop culture semantical categories) as well as structures derived from lexical tools (Wordnet) and unsupervised NLP techniques resulting in word embeddings. Our new method for password cracking outperforms existing architectures and could serve researchers and forensic experts by using it for proactive password checking and password strength estimation. Nonetheless, we would suggest training these models on a bigger dataset to leverage the extended vocabulary for a more precise assessment of the security of a password.

We show that with our architecture, we can leverage a relatively small number of samples to outperform other comparable tools. Moreover, the quality of the feature space using the Silhouette score and the Calinski Harabasz Index, seem to be outperformed by the fact that vocabulary can be fully exploited in a feature space. This could explain the difference in performance between the WordNet and the embeddings feature space. Even though, WordNet and the WUP metric output a space of characteristics that is more suitable for clustering, it does not account for all the words in the vocabulary, which probably accounts for the decline in performance (one has to

label the rest of the data with a more naïve approach). Nonetheless, the difference is very small and shows improvements concerning the other methods tested.

In future works, we suggest exploring the security implications of integrating personal information into the PCFG models. As we have highlighted in this work, language and culture are shapers of password structures. Thus, websites, and other stakeholders that use passwords as an authentication method, should consider the cultural patterns proper of their users.

REFERENCES

- Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching word vectors with subword information. *CoRR*, abs/1607.04606.
- Caliński, T. and JA, H. (1974). A dendrite method for cluster analysis. *Communications in Statistics - Theory and Methods*, 3:1–27.
- Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.
- Generic Human (2012). Complexity dictionary. <https://stackoverflow.com/questions/8870261/how-to-split-text-without-spaces-into-list-of-words>. Accessed: 2020-03-06.
- Golla, M. and Dürmuth, M. (2018). On the accuracy of password strength meters. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS '18*, page 1567–1582, New York, NY, USA. Association for Computing Machinery.
- Hitaj, B., Gasti, P., Ateniese, G., and Pérez-Cruz, F. (2017). Passgan: A deep learning approach for password guessing. *CoRR*, abs/1709.00440.
- Hranický, R., Zabal, L., Ryšavý, O., Kolář, D., and Mikuš, D. (2020). Distributed pcfg password cracking. In Chen, L., Li, N., Liang, K., and Schneider, S., editors, *Computer Security – ESORICS 2020*, pages 701–719, Cham. Springer International Publishing.
- Ma, J., Yang, W., Luo, M., and Li, N. (2014). A study of probabilistic password models. In *2014 IEEE Symposium on Security and Privacy*, pages 689–704.
- Maoneke, P. B., Flowerday, S., and Isabirye, N. (2018). The Influence of Native Language on Password Composition and Security: A Socioculture Theoretical View. In Janczewski, L. J. and Kutylowski, M., editors, *33th IFIP International Conference on ICT Systems Security and Privacy Protection (SEC)*, volume AICT-529 of *ICT Systems Security and Privacy Protection*, pages 33–46, Poznan, Poland. Springer International Publishing. Part 1: Authentication.
- Martin, L., Muller, B., Ortiz Suárez, P. J., Dupont, Y., Romary, L., de la Clergerie, É. V., Seddah, D., and Sagot, B. (2020). Camembert: a tasty french language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.
- Melicher, W., Ur, B., Segreti, S. M., Komanduri, S., Bauer, L., Christin, N., and Cranor, L. F. (2016). Fast, lean, and accurate: Modeling password guessability using neural networks. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 175–191, Austin, TX. USENIX Association.
- Mori, K., Watanabe, T., Zhou, Y., Akiyama Hasegawa, A., Akiyama, M., and Mori, T. (2019). Comparative analysis of three language spheres: Are linguistic and cultural differences reflected in password selection habits? In *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS PW)*, pages 159–171.
- Morris, R. and Thompson, K. (1979). Password security: A case history. *Commun. ACM*, 22(11):594–597.
- Princeton University (2010). About WordNet. <https://wordnet.princeton.edu/>.
- Rousseeuw, P. (1987). Rousseeuw, p.j.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *comput. appl. math.* 20, 53-65. *Journal of Computational and Applied Mathematics*, 20:53–65.
- Speer, R., Chin, J., Lin, A., Jewett, S., and Nathan, L. (2018). Luminosinsight/wordfreq: v2.2.
- Turner, S., Hanel, R., and Corominas-Murtra, B. (2014). Understanding zipf’s law of word frequencies through sample-space collapse in sentence formation. *Journal of the Royal Society, Interface / the Royal Society*, 12.
- Veras, R., Collins, C., and Thorpe, J. (2014). On the semantic patterns of passwords and their security impact.
- Wang, D., Wang, P., He, D., and Tian, Y. (2019). Birthday, name and bifacial-security: Understanding passwords of chinese web users. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 1537–1555, Santa Clara, CA. USENIX Association.
- Wang, D., Zhang, Z., Wang, P., Yan, J., and Huang, X. (2016). Targeted online password guessing: An underestimated threat.
- Weir, M. (2017). Pcfg_cracker. https://github.com/lakiw/pcfg_cracker. Accessed: 2020-03-03.
- Weir, M., Aggarwal, S., de Medeiros, B., and Glodek, B. (2009). Password cracking using probabilistic context-free grammars. pages 391–405.
- Wheeler, D. L. (2016). zxcvbn: Low-budget password strength estimation. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 157–173, Austin, TX. USENIX Association.
- Yu, S., Xu, C., and Liu, H. (2018). Zipf’s law in 50 languages: its structural pattern, linguistic interpretation, and cognitive motivation.
- Zipf, G. (1949). *Human behavior and the principle of least effort*. Addison-Wesley, Cambridge.