Automatic Product Categorization for Anonymous Marketplaces

Michael Graczyk, Kevin Kinningham

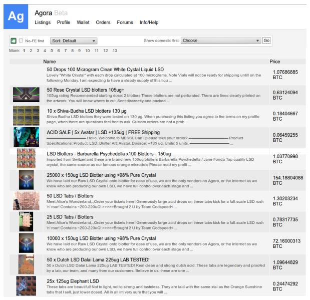
Abstract—In this paper, we present a machine learning algorithm to classify product listings posted to anonymous marketplaces. We classify these listings according to the type of product being sold. The categories are derived from the 12 product categories on a popular anonymous marketplace, Agora. Our algorithm is a combination of TF-IDF for feature extraction, PCA for feature selection, and SVM for classification. We compare our algorithm to simpler models, including multinomial-event naive bayes and a baseline algorithm that uses simple string pattern matching. We achieve an accuracy of 79% on a withheld test set compared to an accuracy of 62% our baseline model.

1 Introduction

Anonymous marketplaces are a rapidly growing segment of online illegal drug sales. Figures 1 and 2 show the mainstream user interface and product listing growth of one such site. However, due to their clandestine nature, it can be difficult to extract information about product listings without manual intervention. Law enforcement officials and interested researchers must have an expert manually tag each listing or rely on error prone ad-hoc methods of catagorizing product listings.

To improve this process, we developed a machine learning algorithm that can automatically categorize listings with high accuracy. The input to our algorithm is the listing text, including it's title and description. We then use TF-IDF to extract features followed by PCA to select features from the text. Finally, we use a SVM to classifiy the features and output a product category.

All data processing and machine learning was executed using tools and algorithms in scikit-learn [1]. Plots were generated using matplotlib [2].



□ No-FE first Sort: Default

Show domestic first: Choose

More: 1 2 3 4 5 6 7 8 9 10 11 12 13

Name	Price
50 Drops 100 Microgram Clean White Cystal Liquid LSD Lovely "White Crystal" with each drop calculated at 100 micrograms. Note Vials will not be ready for shipping untill on the following Monday. I am expecting to have a steady supply of this liqu ...	1.07686885 BTC
50 Rose Crystal LSD blotters 105ug+ 105ug rating Recommended starting dose: 2 blotters These blotters are not perforated. There are lines clearly printed on the artwork. You will know where to cut. Sent discreetly and packed ...	0.63124094 BTC
10 x Shiva-Budha LSD blotters 130 µg Shiva-Budha LSD blotters they were tested on 130 µg. When purchasing this listing you agree to the terms on my profile page, when there are questions feel free to ask. Custom orders are not a prob ...	0.18404667 BTC
ACID SALE | 5x Avatar | LSD +135ug | FREE Shipping ——————— Hello. Welcome to MESSI. Can I please take your order? ——————— Product Specifications: Product: LSD. Blotter Art: Avatar. Dosage: +135 ug. Units: 5 units. ——————— ...	0.06459255 BTC
LSD Blotters - Barbarella Psychedella x100 Blotters - 150ug Imported from Switzerland these are brand new 150ug blotters Barbarella Psychedella / Jane Fonda Top quality LSD crystal, the same source as our famous orange microdots Please read my profil ...	1.03770998 BTC
25000 x 150ug LSD Blotter using >98% Pure Crystal We have laid our Raw LSD Crystal onto blotter for ease of use, we are the only vendors on Agora, or the internet as we know who are producing our own LSD, we have full control over each stage and ...	154.18804088 BTC
50 LSD Tabs / Blotters Meet Alice's Wonderland,.,Order your tickets here! Generously large acid drops on these tabs kick for a full-scale LSD rush 'n' roar! Contains ~200-220uG! >>>>>Brought 2 U by Team Godspeed< ...	1.30203234 BTC
25 LSD Tabs / Blotters Meet Alice's Wonderland,.,Order your tickets here! Generously large acid drops on these tabs kick for a full-scale LSD rush 'n' roar! Contains ~200-220uG! >>>>>Brought 2 U by Team Godspeed< ...	0.78317735 BTC
10000 x 150ug LSD Blotter using >98% Pure Crystal We have laid our Raw LSD Crystal onto blotter for ease of use, we are the only vendors on Agora, or the internet as we know who are producing our own LSD, we have full control over each stage and ...	72.16000313 BTC
50 x Dutch LSD Dalai Lama 225ug LAB TESTED! 50 x Dutch LSD Dalai Lama 225ug LAB TESTED! Real clean and strong dutch acid. These tabs are legendary and proofed by a lab, our team, and many from our customers. Believe in us, these are one ...	1.09644829 BTC
25x 125ug Elephant LSD These tabs are beautiful! Not to light, not to strong and tasteless. They are laid with the same xtal as the Orange Sunshine tabs that I sell, just lower dosed. All in all im very sure that you will ...	0.24474292 BTC

Fig. 1. Agora Marketplace on January 2015

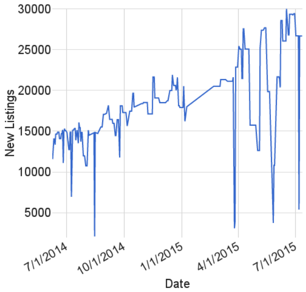


Fig. 2. New Product Listings Over Time

2 Related Work

There have been several attempts to analyze anonymous marketplaces [3] [4]. In [4], the author crawled The Silk Road for eight months, analyzing product listings and the overall distribution of product listings. However, they relied on vendor supplied categorizations, which do not exist for all marketplaces. Additionally, some vendors purposely misclassify their product to appear higher in marketplace search results. Correcting for these problems requires manual labeling of at least a large fraction of the data (as has been used in analysis on other marketplaces, such as RAMP).

Most of the machine learning techniques we used are well documented in the literature as effective building blocks for document classification systems. We extract word features from each listing using Term Frequency-Inverse Document Frequency (TF-IDF), which has proven effective in document categorization. [5, p. 118]. Using SVD for dimensionality reduction is a common technique for reducing the size of the feature space for document classification [6].

3 Dataset and Features

We used an anonymous marketplace product listing dataset created by the researcher known as Gwern [7]. This dataset was created by crawling several marketplaces daily, from June 6th, 2014 to July 7th, 2015. For each product listing, we extracted the posting text, including name and description. The resulting dataset was then cleaned to remove parse errors and duplicate listings. We then tokenized the cleaned listings by extracting all words and numbers seperated by spaces or symbols. The final dataset had about 84,000 unique product listings and was used as input to feature extraction.

Crawl Date	2014-06-28
Category	MDMA
Title	28 Grams of Interways Crystal Clear Molly
Price	1.37853692 BTC
Description	This is 28 grams of Interways crystal clear molly. It will be both rocky and sandy as thats how I received it. I am pre packaging these up accordingly and only cr...

TABLE 1
Example product listing on Agora

To convert the text to features, we used Term Frequecy-Inverse Document Frequence (TF-IDF). TF-IDF converts each token in the listing to a token weight based on how important that token is to the listing, normalized by the number of times the token appears in the whole corpus. This normalization helps to reduce the impact of common token in the corpus. Finally, each listing is then converted to a vector of token weights, similar to word2vec.

$$tf_{t,d} = \text{number of times token } t \text{ appears in document } d$$

$$idf_t = \log \frac{\text{total number of documents}}{\text{number of documents the token } t \text{ appears in}}$$

$$w_{t,d} = (1 + \log(tf_{t,d})) * (1 + idf_t)$$

Additionally, we manually labeled about 500 product listings to train our classifier as well as to use for cross-validation. We classified each product listing into one of twelve categories. These product categories were derived from the product categories on both Agora [8] (one of the largest anonymous markets) and /r/darknetmarkets (a popular forum for advertising and discussing drugs).

4 METHODS

Our learning algorithm combines unsupervised feature selection with a supervised classifier. Our data set includes a large number of uncategorized documents and a small number of manually categorized documents. In both cases, each document's length is only a few dozen words. Because of the the relatively short length of the documents and the small size of the labeled training set, a supervised learning algorithm alone would have insufficient discriminative information to perform robust classification. Instead of using supervised learning in isolation, we extract more discriminative features from our unlabeled data set using a simple dimensionality reduction. These high quality features are then used to train a supervised classifier which is much more likely to generalize well to new examples because the underlying structure of the feature space is more representative of the semantic structure of our entire corpus.

4.1 Unsupervised Feature Selection

We perform feature selection using principal component analysis (PCA). In this case, the principal components are uncorrelated inferred meanings of tokens based on their usage patterns within the training data. This technique is known as Latent Semantic Indexing (LSA) because each document is indexed (described as vector) by a set of inferred statistical (latent) parameters which are chosen using their semantic relationships. This technique has been found to be an effective way to extract structure from unlabeled documents [6].

Let $X_{unlabeled} \in \mathbb{R}^{N \times M}$ be the matrix of N documents and M tf-idf features for which there are no category labels. We compute the truncated singular value decomposition with rank r to produce the transformation matrix $W_r \in \mathbb{R}^{M \times r}$. That is, we find compute

$$\underset{W_r \in \mathbb{R}^{M \times r}}{\text{argmin}} \quad \left| X_{unlabeled} - X_{unlabeled} \begin{bmatrix} Wr & 0 \\ 0 & 0 \end{bmatrix} \right|_{fro}^{2}. \tag{1}$$

We use this transformation to select input features for our supervised classification algorithm by computing $\tilde{X}_{labeled} = X_{labeled} W_r$, where $X_{labeled}$ is a matrix containing our labeled training data.

4.2 Supervised Product Categorization

We categorize product listings using a soft-margin support vector machine (SVM) for each category. We discriminate categories using a one-versus-rest decision rule.

Each support vector machine learns a decision boundary by maximizing the margin between the decision boundary and each training points.

More formally, we will consider a single category c and explain how the SVM determines the decision boundary h_c. For simplicity, we assume each label has value $y_i = 1$ if product i is in category c and -1 otherwise. Let $\mathcal{D} = \{(x_i, y_i), i = 1 \ldots N\}$ be the training data set with features $x_i \in \mathbb{R}^r$ and labels $y_i \in \{-1, 1\}$.

$$\begin{aligned} \underset{w,\xi,b}{\text{argmin}} \quad & \frac{1}{2}||w||^2 + C \sum_{i=1}^{N} \xi_i \\ \text{s. t.} \quad & y_i(w \cdot x_i - b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \end{aligned} \tag{2}$$

The optimization problem can be rewritten into its so-called "dual form". The dual form reveals a simpler optimization problem without explicit slack variables (ξ) and where many of the optimization parameters will be zero.

$$\begin{aligned} \underset{\alpha \in \mathbb{R}^N}{\text{argmax}} \quad & \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j}^{N} \alpha_i \alpha_j y_i y_j x_i^{\intercal} x_j \\ \text{s. t.} \quad & 0 \leq \alpha_i \leq C, i = 1, \ldots, N \\ & \sum_{i=1}^{N} \alpha_i y_i = 0 \end{aligned} \tag{3}$$

5 EXPERIMENTS AND RESULTS

5.1 Algorithm Tuning Procedure

Our algorithm includes four hyperparameters that are not automatically chosen as part of the training process. These hyperparameters are $\max_{\text{df}}$, r, C, and the SVM regularization penalty and loss function.

5.1.1 $\max_{df}$

$\max_{\text{df}}$ is the maximum document frequency allowed for a token to be used a feature. Any token which appears in the data set with frequency greater than $\max_{\text{df}}$ is considered a stop token and is ignored by the classification algorithm. A larger value of $\max_{\text{df}}$ gives more information to the learning algorithm, but increases computational cost and adds potentially useless features to the feature selection process.

In our dataset, some tokens apply to several categories, and thus appear in a lot of documents, even thought they might be useful in discriminating categories. For example, the token "mg" occurs in many listings, but does not usually occur in listings categorized as marijuana or other. Likewies, tokens like "India" or "China" can usually give us some idea of the class of drug, but are sill very common in the overall dataset. For this reason, we expect the optimal value of $\max_{\mathrm{df}}$ to be high, since we do not want to throw away words soley becuase they are common.

5.1.2 PCA output dimensionality (r)

r is the dimensionality of the feature space selected by PCA. The SVM operates on training examples which have been transformed into this feature space. Larger values of r increase computation time but make it easier for the SVM to find high margin, simple, separating surfaces between each class and non-class training examples. SVM hypothesis found for larger r may also generalize better because the SVM was able to use more information to decide on its hypothesis. Smaller values of r simplify computation and make it more difficult for the SVM to find high margin separating surfaces.

However, smaller values of r may improve discriminative quality in the input features. The unsupervised feature selection process uses vastly more data than the SVM training process, so feature selection may reveal structure in the data that cannot be learned by the SVM. Smaller r enables feature selection to make stronger quantitative statements about the semantic structure of the data.

5.1.3 SVM regularization weight (C)

C is the SVM regularization weight. This real number determines the relative importance of regularization as compared to maximizing the margin. We use the same regularization for each category for simplicity. Large values of C imply more complicated decision boundaries in which some input features may be vastly more important than others. Small values lead to simple decision boundaries which consider each feature dimension similar in importance. Since our use of feature preselection serves to make the data somewhat compact, we expect the optimal C to be somewhat small.

5.1.4 SVM regularization function

For the SVM regularization penalty and loss functions, we restricted our choices to L_1 or L_2 regularization penalties and linear or quadratic loss functions. The choice to limit the possible loss functions was made to simplify our implementation.

5.1.5 Hyperparameter Selection

To choose values for our hyperparameters $\max_{\mathrm{df}}$, r, and C, we used course grained grid search with 5-fold cross validation. Grid search operates by exhaustively enumerating every possible combination of hyperparameters and selecting the combination that performs the best on the validation test set.

We searched 100 possible values for C logarithmically spaced from 10^{-4} to 1 and 10 values of r linearly spaced from 200 to 700. We also searched 100 possible values of $\max_{\mathrm{df}}$ logarithmically spaced from 10^{-4} to 1. We also experimented with both L_1 and L_2 for SVM regularization, and decided to use L_1 since it performed better on our dataset.

Hyperparameter	Value
$\max_{\mathrm{df}}$	10^-1
r	300
C	10^-4

TABLE 2
Hyperparameters Used

5.2 Experimental Procedure

We analyzed our algorithm by fitting our complete processing pipeline using all of our training data set, then testing the accuracy of the model using a previously untouched test set. The structure of our processing pipeline can be seen in Figure 5.2.

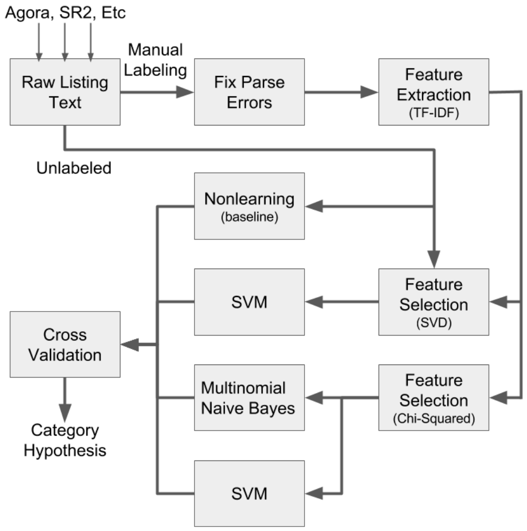


Fig. 3. Data Pipeline

We had $\frac{1}{3}$ of our data at the beginning of our algorithm design process and had not used it for training or cross validation. We used our processing pipeline to compute category predictions for each point the in test set.

5.3 Results

The most important metric for our classification algorithm is categorization accuracy. The accuracy for a test set is the proportion of labels that were correctly assigned. For comparison, we compared the accuracy of our method with three other algorithms.

- A baseline model that used simple substring search, with substrings chosen by an online market expert. For example, if a product listing contained the text ”xanax”, then the listing was classified under ”benzos”.
- The χ^2 test provides a simple way to remove features that are not correlated with any labeling, and is much faster than SVD. However, it is also much less accurate and has a much higher output dimension than SVD.
- Multinomial Naive Bayes, using the same χ^2 features.

The accuracy for each model is shown in Figure 4. The figure shows that our model outperformed several more simple models.

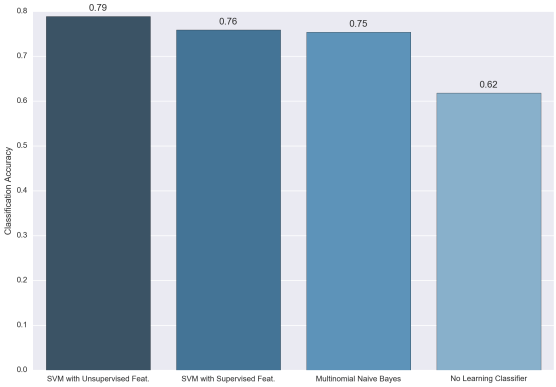


Fig. 4. Comparison of Model Accuracies

benzo	bensedin mot unmarked som bars 2mg diazepam xanax clonazepam zepose
dissociative	purity reputation mxe 1000g methoxetamine chopping lab requested
ecstasy	240 pressed pills mephedrone dutch red ecstasy express crystals 84
misc	camel zolpidem lunesta eszopiclone caffeine phenargan zolab tranax
opiate	4mg 30mg heroine methadone opium codeine fentanyl naloxone tramadol
steroid	10ml boldoject dianabol ml propionate sibutramine sibutramin test testosterone
psychedelic	buddha stand sheet babies mushrooms psilocybe hearts blotter nbome lsd
research chemical	fluoroamphetamine 36794 al 14g lad chiral dichloropane mdai fa apdb
prescription	mobic pseudoephedrine tadalafil generic dexamphetamine medications
stimulant	coke amphetamine check modafinil modalert adderall methamphetamine
marijuana	open grown dream crash kush weed hash wax taste sativa
other	size windows custom ways facebook account kinesiology book dpz guide

TABLE 3
Top Tokens For Each Category

The precision recall curve compares shows the tradeoff between precision (the number of true positives over the total number of positives) vs recall (the number of true positives over the number of true positives plus the false negatives) as we vary our algorithm's parameters. Figure 5 shows the precision-recall curve for our model.

Classification quality was not equal between each category. Figure 6 shows the confusion matrix of our algorithm on our test set. The grid position in row i and column j shows the number of times a listing which is truly in category i was classified by our algorithm as category j. The values in the grid cells show the absolute number of test points, while the colors show that number normalized by the true number of test points in each category.

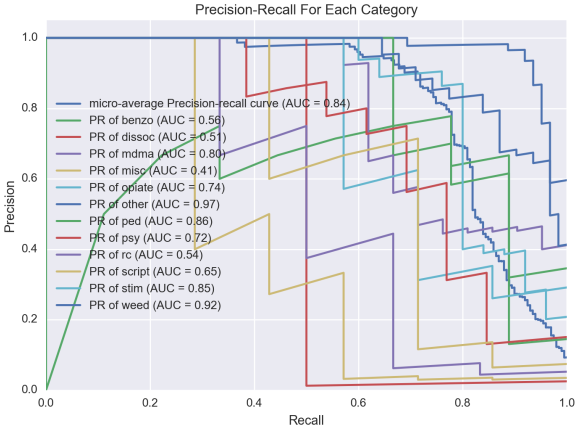


Fig. 5. Normalized Confusion Matrix

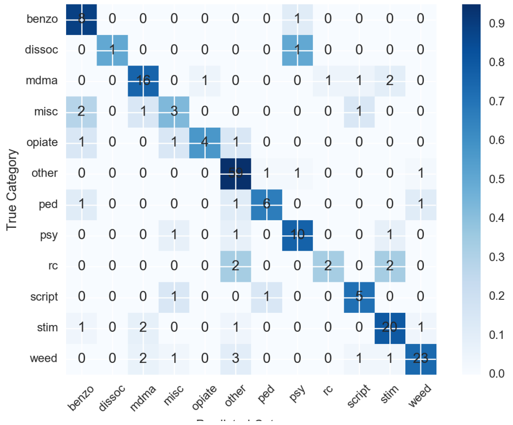


Fig. 6. Normalized Confusion Matrix

The confusion matrix shows that our classifier was particularly bad at classifying the ”other” category. We believe this is because of the large variety of ”other” products available and the somewhat arbitrary definition of the category. For example, we one category of goods we included in ”other” was illegal digital goods, like movie downloads. Because of this, drugs that had similar descriptions to unrelated products got misclassified as ”other”. To highlight this effect, our algorithm misclassified a strain of weed called ”The Big Lebowski” as ”other” because we had included a label for the movie download of ”The Big Lebowski”. See Table 4 for the complete example.

True Category	Marijuana
Predicted Category	Other
Title	1 4 Ounce The Big Lebowski
Description	The Big Lebowksi 2 3 week cure Indica sativa blend. Over the years pot has gotten a lot stronger especially with the advent of indoor hydroponic growing under metal halide and high pressure s...

TABLE 4
Text of Typical Misprediction

6 Future Work

In this project, our training and testing used data from only a single market. The algorithm could be made more robust by including data from more sources. Test data drawn from a broader source would also provide a better generalization estimate.

Additionally, our algorithm could have considered features other than the product listing text when classifing products. For example, we could have used product price as a feature. However, as the amount of product is not listed in a consistant format, normalizing the prices would require extracting the amount of product being sold.

7 Conclusion

In this project, we developed a machine learning algorithm that can classify product listings according to the type of product being sold. Our algorithm consisted of a pipeline using TF-IDF for feature extraction, SVD for feature selection, and SVM for final classification. We then used cross-validation to select hyperparameters and tune our algorithm.

We also evaluated our algorithm in comparison to several other models:

- A baseline model that used simple string search
- SVM with a Chi-Squared test for feature selection
- Multinomial Naive Bayes

Our algorithm had an accuracy of 77% compared to the baseline accuracy of 62%. Our algorithm also outperformed the alternate SVM and multinomial models.

Our algorithm was able to outperform the alternate models because it was able to take advantage of structure in the large unlabeled dataset. In this project, we had a small amount of labeled data, and a very large amount of unlabeled data. The SVD we use is performed on the unlabeled data, which helps to expose structure not captured by the labeled data. We then project our labeled data onto the vector space chosen by the SVD and train our SVM on the result. This allows to take advantage of both our unlabeled and labeled data in selecting our decision boundry.

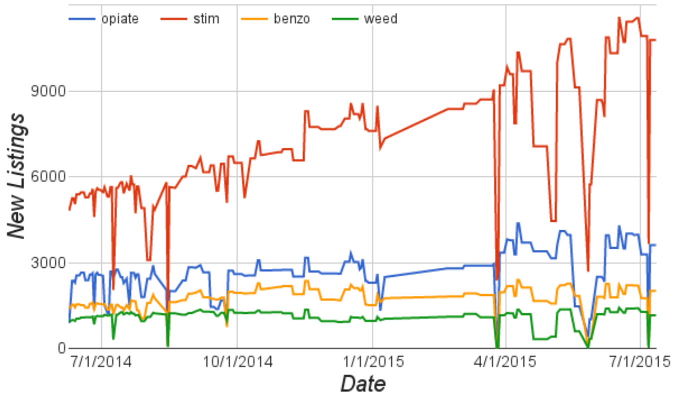


Fig. 7. New Listings By Category On Agora

We also used our classifier to measure the number of new products by category over time. We found that by far the most common product listed on Agora is stimulants. Interestingly, this was significantly different than an earlier anonymous market, The Silk Road, where marijuana was the most common listed product, with stimulants a distant fourth [4].

This also matches reports from user and vendor forums, many of whom have complained that the legalization of marijuana has reduced the profitability of selling online. Conversely, amphetamine demand has dramatically risen, while production costs have dropped. This has resulted in an apparent increase in product listings. As far as we are aware, we are the first to rigorously document this switch. This shows that our algorithm is extremely useful in practice, particularly for law enforcement and researchers.

REFERENCES

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

[2] J. D. Hunter, “Matplotlib: A 2d graphics environment,” *Computing In Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[3] G. Branwen, “Silk Road: Theory And Practice,” http://www.gwern.net/Silk%20Road, Augest 2015, Accessed: 2015-12-09.

[4] N. Christin, “Traveling the silk road: A measurement analysis of a large anonymous online marketplace,” in *Proceedings of the 22nd international conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2013, pp. 213–224.

[5] P. R. Christopher D. Manning and H. Schtze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[6] J.-T. Sun, Z. Chen, H.-J. Zeng, Y.-C. Lu, C.-Y. Shi, and W.-Y. Ma, “Supervised latent semantic indexing for document categorization,” in *Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on*. IEEE, 2004, pp. 535–538.

[7] G. Branwen, “Dark Net Market archives, 2011-2015,” http://www.gwern.net/Black-market%20archives, July 2015, Accessed: 2015-12-09.

[8] A. Greenberg, “Drug market agora replaces the silk road as king of the dark net,” http://www.wired.com/2014/09/agora-bigger-than-silk-road, Sept. 2014, Accessed: 2015-12-09.