https://doi.org/10.2352/ISSN.2470-1173.2018.10.IMAWM-444

Application of natural language processing to an online fashion marketplace*

***Kendal Norman*[a], *Zhi Li*[a], *Young-Taek Oh*[a], *Gautam Golwala*[b], *Sathya Sundaram*[b], *Jan Allebach*[a];**
[a] *School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, 47907, U.S.A;*
[b] *Poshmark Inc., 101 Redwood Shores Pkwy, 3rd Floor, Redwood City, CA 94065*

Abstract

Online fashion marketplaces are experiencing a boost in popularity. People see the appeal of websites where they can sell their products by providing information such as title, price, description, and pictures. With this popular new model for buying and selling fashion products comes a new set of challenges to face. With attention focused on analyzing product titles provided by the user, this paper covers the application of natural language processing techniques and a couple of machine learning algorithms to an online fashion marketplace, with the goal of predicting an item's category or subcategory. The paper begins with an overview of some popular preprocessing techniques in the context of analyzing titles. These preprocessing techniques are vital to the next step, the actual training of the models. This paper covers the development and performance of two models: a model that utilizes a Nave Bayesian learning approach, and a model that utilizes Support Vector Machines as the prediction model. The results from each prediction model are compared and discussed. The results show that the prediction model that utilized the Support Vector Machines was more accurate, and that natural language processing techniques can be effectively applied to an online fashion marketplace to predict an item's category or subcategory.

Introduction

Online fashion marketplaces like the one Poshmark has created are places where users can buy and sell fashion products. Poshmark uses a peer-to-peer business model in which consumers are both the buyers and sellers. Users put their products online to sell, and other users purchase these products. Poshmark serves as the medium through which these sales take place.

For ease of navigation for users, the Poshmark website has an organizational structure for the items consisting of categories and subcategories. Each category contains a unique set of subcategories. The image from the Poshmark website in Figure 1 shows the selected category, the "Dresses" category which is pointed to by the purple arrow, expanded out into its subcategories which are surrounded on the left and right by the blue lines. In total, there are 16 categories on the website, and each category can have anywhere from 4 to 17 subcategories. To point out a couple of categories with the highest and lowest subcategory totals respectively, "Shoes" is a category that has 17 subcategories, and "Swim" is a category that has 4 subcategories.

This dynamic of users serving as both buyers and sellers creates a distinct feature of peer-to-peer marketplaces that is both a perk and drawback for this business model; peer-to-peer marketplaces put a lot of responsibility on the user. When the user wants to put a product up for sale, they must provide accurate information about said product, as well as pictures. The accuracy of the information a user provides is directly related to whether the product gets sold or not. When putting items up for sale online, even the most careful of users are bound to make mistakes and mislabel their items at some point. An example of this mislabeling of an item is show in Figure 2. This image shows, in the red box in the top left, that the item is labeled "Pants". The red underline on the right of the image, however,

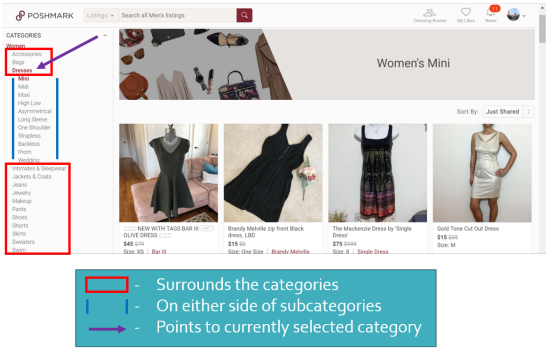

Figure 1. *Screenshot from poshmark.com displaying the organizational structure of their items.*

*Research supported by Poshmark, Inc. Redwood City, CA, 94065

shows that this item should be labeled ”Jeans”. This mislabeling of information causes serious problems because other users will struggle to find said product, which results in the product not being sold promptly. Slowing the rate at which products sell is undesirable for both users and Poshmark themselves. Users want their products to be sold as soon as possible, and Poshmark wants their site to be as attractive as possible. Therefore, there is plenty of incentive to cut down the amount of mislabeled data on the website.

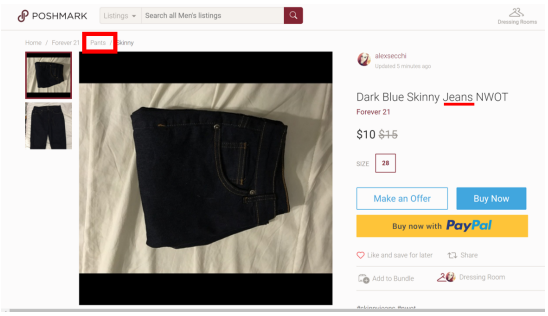

Figure 2. *Screenshot from poshmark.com displaying mislabeled data. The pants shown in the image should be labeled "Jeans" but are instead labeled "Pants".*

Users who are new to the site may not be immediately familiar with the interface of the website or the organizational structure of the items, which is inconvenient and most certainly leads to some data labeling errors. The purpose of this project is to devise a system that will help cut down on the mislabeling errors that users make, and help ease the process of putting items online by providing users with suggested categories and subcategories.

Preprocessing

Tokenization

The first part of the preprocessing step for the models developed is tokenization. Tokenization, in the context of natural language processing, is the process of splitting a string into words which can then potentially be subject to further preprocessing. These words will ultimately serve as features in a training set for the desired machine learning algorithm. Common tokenization techniques involve:

- Removal of punctuation
- Splitting of strings around spaces to form the words
- Splitting of contractions into the two words they represent
- Conversion of entire string to lowercase

Stopword Removal

Stopword removal is a commonly used preprocessing technique. Stopwords are words that are important to human speech patterns, but do not actually provide much meaning to a body of text. As an example, words like ”the”, ”a”, and ”this” are all stopwords. These words tend to occur with high frequency in grammatical bodies of text, and occur very frequently across all documents regardless of the label. This means these words are not descriptive for predictive modeling because the presence or lack of these stopwords does not give any sort of information about what label the data should have. The solution to this problem is to simply remove these stopwords, since in most cases there is no reason to keep them. Removing stopwords can lower the size of the vocabulary needed to be stored by the program.

Word Stemming

Word stemming encapsulates the process of transforming a word from its many different grammatical forms into its root form, as show in Figure 3.

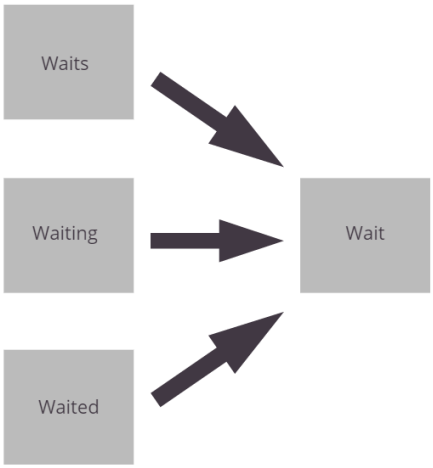

Figure 3. *Visual display of how words get transformed into their root words through stemming.*

There are three types of stemming algorithms: truncating, statistical, and mixed approaches [1]. Truncating algorithms tend to use a heuristic approach where the algorithm designer makes rules that are used to decide what portion of a word to cut off. Statistical algorithms, on the other hand, use some sort of statistical analysis to decide what the root of the word is. Finally, the mixed approach is a mixture of both the heuristic truncating approach and the statistical approach.

Stemming is desirable because it shrinks the overall size of the vocabulary across a corpus. This preprocessing technique both eases the computational load on the machine learning algorithm, as well as making the data potentially less complicated. For example, the words ”waits”, ”waited”, and ”waiting” are ideally all reduced to the word ”wait”, as shown in Figure 3. When these words turn into features for the machine learning algorithm, they will not be three separate words but will be one single word, which in some cases is desirable. The drawback to word stemming is that there may be some loss of meaning when portions of words are cut off. This loss of meaning can mean nothing to some applications of machine learning, but to other applications this extra meaning can be vital.

Bigrams

Bigrams, also called shingles, are all the pairs of two adjacent words in a sentence.

For example, take the sentence:

- ”Shingles can be very informative features.”

The shingles from this sentence are:

- (”Shingles”, ”can”)
- (”can”, ”be”)
- (”be”, ”very”)
- (”very”, ”informative”)
- (”informative”, ”features”)

The use of shingles can help preserve context and word order where utilizing each word in isolation or as part of a bag-of-words model cannot. This extra context can be invaluable to machine learning algorithms because word order can be significant with regards to prediction accuracy.

Preprocessing in Context

While these are all popular preprocessing techniques and can be effective in the proper circumstances, it is important to analyze these techniques to decide whether they seem appropriate for the purpose of textual analysis of titles. Titles, for the most part, tend to be quite short, largely ungrammatical, and not very structured. They also tend to have a bit of a focus on the last word of the title as being one of the most informative words, as the last word tends to be the noun and the words preceding tend to be some sort of descriptor for that noun. Tokenization is a given, and is integral to any sort of natural language processing system because the system must have some sort of token to work with, be that words as tokens or maybe letters as tokens.

For our application, word stemming is not a particularly valuable preprocessing method. Due to the nature of the data set, titles often contain proper nouns. Some word stemming algorithms, especially truncating algorithms, can potentially reduce proper nouns to a root word that it should not have. Proper nouns should not be touched by stemmers because preserving the entirety of the proper noun is important. For example, changing the name of a brand is not desirable because it is not supposed to have a root, and is not supposed to be connected to other words that share this root. This reducing of a brand name to a non-existent root word removes much more meaning from the corpus than is desirable.

The implementation of bigrams, on the other hands, is very valuable in this context because titles are considered ungrammatical and structured only in the vaguest sense. Across grammatical text documents, there will occur many frequent pairs of words just for the sake of maintaining grammatical correctness. With regards to the ungrammatical nature of titles, if two titles share a common bigram it can be quite informative as to whether they share the same category or subcategory. The drawback to using bigrams is that it is going to generate more features, which means utilizing them is more computationally expensive. However, titles tend to be quite short compared to most other natural language processing applications, where each document can be a full length article or other text document. The short length of titles means that there cannot be many bigrams generated from the data set.

As for stopword removal, titles do not frequently contain stopwords as titles do not tend to be grammatical, and therefore do not need them. However, this does not mean they should not be removed when they do occur. When titles contain stopwords, the stopwords still do not contribute much to the meaning of the title, and while removal of stopwords will not help reduce the size of the vocabulary across the corpus by much, it will at least help reduce the size of the vocabulary a little.

Models Developed

Preprocessing is merely the first step in the prediction process. Preprocessing generates the feature sets for training and testing the classifiers. Next, these feature sets and labels need to be utilized to train a model for prediction. Two models have been developed: a Naïve Bayesian model and a Support Vector Machine model. The Naïve Bayesian model is a simple probabilistic implementation, which utilizes multiple Naïve Bayesian Classifiers working within to make predictions. The Support Vector Machine model is a more complicated approach that utilizes a vector space representation of the data and Support Vector Machines to draw decision boundaries to categorize the vectorized titles.

Both these models are organized into subcategory classifiers and a single category classifier. The category classifier is a classifier that predicts the category from the input features. This is in contrast to the subcategory classifier, which predicts the subcategory instead of the category, based on the input features.

There are multiple subcategory classifiers trained, one for each category. This is with the intent that each subcategory classifier be trained to predict an item's subcategory within a given category. The purpose behind each category having a subcategory classifier stems from the fact that the Poshmark website requires that the user put down a category for the item they want to sell, but the user is not required to put a subcategory. This means, because the category is given, that the classifier need only be able to predict a subcategory from the set of subcategories that belong to the given item's category. The reason that a category classifier is desirable, though the category is given, is because a category classifier could be implemented with the intent of providing a user with category suggestions or providing category correction.

Naïve Bayesian Model

The Nave Bayesian model is implemented using the Natural Language Toolkit (NLTK) [2]. This model uses the Naïve Bayesian Classifier, which is a supervised learning method that works with the fundamental assumption that the features are independent of each other. This particular classifier utilizes the Conditional Bayesian Probability Theorem:

$$P(label|F) = \frac{P(label) \times \prod_{n=1}^{N} P(f_i|label)}{P(F)} \quad (1)$$

Where F is the set of features, N is the number of features, $P(label)$ is the prior probability representing the distribution of the data into the labels, and finally $P(f_i|label)$ is the conditional probability of a specific feature (a word or bigram) being in the title, given a certain label.

The input features for this model consist of each word in the tokenized title, each bigram formed from the title, a first word marker, and a last word marker.

Support Vector Machine Model

This model was implemented using scikit learn [3]. The Support Vector Machine model utilizes Support Vector Machines as the classifier of choice. They form decision boundaries in a vector space filled with points, with the intent of dividing one label from another as accurately as possible. Support vector machines are capable of forming decision boundaries that can accurately label the data with differences in only one dimension. This is significant for text classification because it is entirely possible that two titles will only have one word of difference, but different labels. For example, "black suede shoes" compared to "black suede jacket". They both share only one word of difference and two similar words, but are of different categories. Support Vector Machines are equipped to handle situations like this.

One of the complications of this model is that Support Vector Machines require all features to be in a numeric vector format, which means across the corpus of titles collected, each title must be converted to a vector. There are a couple of ways to form vector space representations of text corpora: one way consists of giving each unique word across the whole corpus a dimension in the vector space, and the other way involves taking a chosen subset of the vocabulary from the data set and using each word in that subset as a dimension of the vector space. Encoding a document into the vector space involves finding the term frequency, the number of times a word occurs in a document. This term frequency must be collected for each term in the document. Each word's term frequency becomes the value for that term's dimension of the titles vector representation. This model's implementation not only generates a dimension for each unique word in the corpus, it also generates a dimension for each unique bigram across the corpus of titles, with each bigram's dimension populated by the term frequency of each bigram within the title being encoded. The resulting dimensionality of the vector space used for this classification model is quite large, which can create computational problems. Without a sparse vector implementation, which scikit learn has built-in, the high dimensionality of this vector space would be unmanageable [3].

Support Vector Machine classification is relatively simple for binary prediction situations, but multi-class prediction, as in this application, is more complicated. There are two general approaches to non-binary classification using Support Vector Machines: one-vs.-one and one-vs.-rest. One-vs.-one attempts to create a decision boundary for every possible pair of prediction labels, while one-vs.-rest attempts to create a decision boundary dividing one prediction label from the rest of the prediction labels, and does this for each label. Historically speaking, one-vs.-one has been more accurate, but had a much longer runtime. However, a more sophisticated method of one-vs.-rest SVM modeling has achieved accuracy results rivaling that of one-vs.-one SVM modeling. It is called the Crammer-Singer method [4], and our model uses this method.

Classification Results

As shown in Figures 4 and 5, the Support Vector Machine model has higher accuracy than the Naïve Bayesian model for every classifier, especially for the "Jeans" category. The Naïve Bayesian model struggles to predict the correct subcategory for the "Jeans" category, which is because the composition of the titles in each subcategory of the "Jeans" category tend to all contain the same words or very similar words. The Naïve Bayesian model is a probabilistic approach, and so fails to generate significant probabilistic differences between titles that share largely similar words. The Support Vector Machine classifier, on the other hand, is designed for forming decision boundaries that can

properly label a title, even if only one word is different.

Conclusion

Based on the results, natural language processing techniques can be applied very effectively to online fashion marketplaces, and in particular these models can help streamline the process of putting items online to sell. Support Vector Machines are an especially good classifier for this application, but at the cost of more expensive computation due to the immense dimensionality of the vector space model involved.

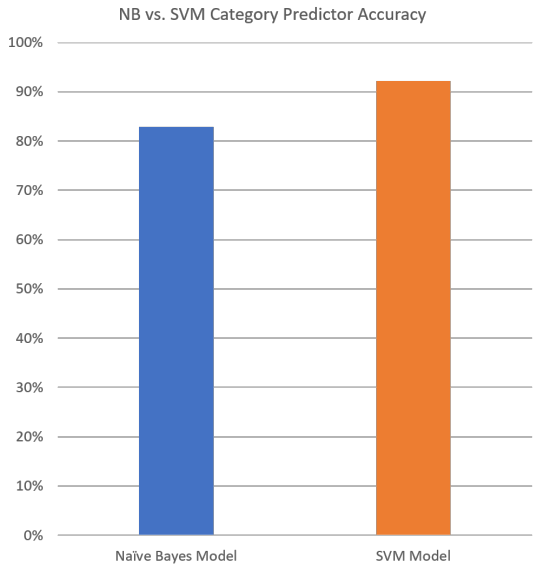

Figure 4. *This graph displays the accuracy with which the category classifier predicts categories, for both models.*

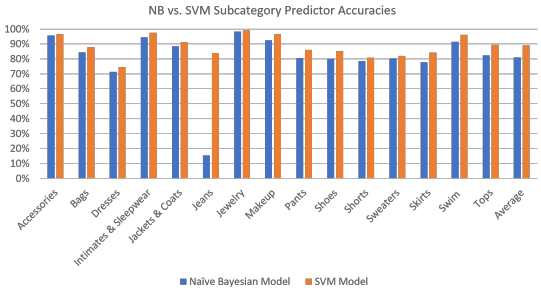

Figure 5. *This graph compares the accuracy of each subcategory classifier for the Naïve Bayesian and SVM models, and includes an average of each models' performance in the furthest right column.*

Acknowledgments

We thank Poshmark Inc. for their continued support of our research project, and the following people for their contributions and suggestions regarding this work: Zhi Li, Young-Taek Oh, and Professor Jan Allebach.

References

[1] A. Jivani, "A comparative study of stemming algorithms," *I. S. S. N.*, November 2011.

[2] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. O'Reilly Media Inc., 2009.

[3] 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.

[4] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *J. M. L. R.*, 2001. [Online]. Available: http://jmlr.csail.mit.edu/papers/volume2/crammer01a/crammer01a.pdf

Author Biography

Kendal Norman is a Junior in his undergraduate program studying Computer Science with a focus on Machine Intelligence at Purdue University, West Lafayette. He is an undergraduate researcher for Professor Allebach working under Zhi Li. His research focus is on applying natural language processing techniques to textual data in an online marketplace. He has been a member of the Purdue IEEE ROV team since 2016, as well as an active member of University Choir in Purdue Musical Organizations. He plans to acquire his Masters degree and PhD studying A.I. Theory.