https://doi.org/10.2352/ISSN.2470-1173.2019.8.IMAWM-419
© 2019, Society for Imaging Science and Technology

New results for natural language processing applied to an on-line fashion marketplace*

Kendal Norman[a], Zhi Li[a], Gautam Golwala[b], Sathya Sundaram[b], Perry Lee[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

In the context of clothing and wearable products, fashion is prone to volatile trends. In clothing fashion there are different seasons of clothing, which account for some of the changing patterns. There are also trends on the scale of the general public, on both shorter time scales and longer time scales. The volatility of these trends poses an issue to conventional natural language processing techniques as well as machine learning approaches. Due to the frequent and unpredictable changes that can occur in a fashion context, models that cannot adapt eventually fail. Like our prior work [1], the model developed here predicts the category and subcategory of fashion items based on the textual contents of the title. The model developed is also capable of adapting to future changes in fashion including the addition of new terminology and changing popularity and classification for existing items. This paper covers some of the problems conventional natural language processing approaches face when tasked with classifying titles in a fashion context. It then covers a few potential approaches to dealing with the implementation of machine learning approaches for classification purposes, and why they fail in the given situation. Finally, this paper presents a solution in the form of a model utilizing feature hashing and the Passive-Aggressive classifier. The results show this model performs as well as the prior model, with a much better training time. This model also possesses the ability to adapt to future changes in fashion.

Introduction

Conventional machine learning approaches fit to a training set that is ideally representative of the test set or application setting, with the ideal outcome being that fitting to the training set will fully encapsulate the data the algorithm will encounter in the real world. When applied to a context with volatile trends, with fashion being a prime example of volatile trends in the real world, this assumption does not hold because there is no amount of training data that could perfectly encapsulate a constantly-changing fashion environment with its constantly-changing trends. This is where the prior work fails.

As a brief overview of the problem being addressed, Poshmark is an online peer-to-peer marketplace. Clientele buy and sell clothing on the website, with Poshmark setting themselves apart from the competition with more of an emphasis on the social media aspects of the site. As such, Poshmark has a strong interest in minimizing the amount of time a user spends creating a listing for a product they are interested in selling. They are also interested in minimizing the amount of time a user has to search the website in order to find a specific product that they are interested in buying. One of the methods Poshmark uses to streamline the process of finding a product is the category-subcategory system. They organize items according to category. Then, within each category, there are subcategories with which to further organize items. Refer to Figure 1 as an example of the category and subcategory structure of the system. Poshmark, as it is currently, does not require users to fill in the subcategory field, for the sake of streamlining the process of posting an item to sell. If this process could be automated effectively, it would help organize items better without burdening the users. This is the ultimate problem being addressed by both this work and our prior work [1].

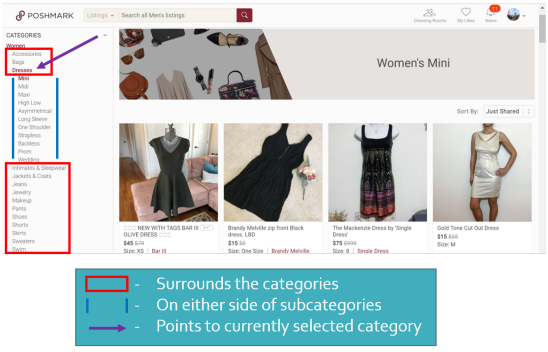


Figure 1. *Screenshot from poshmark.com displaying the organizational structure of their items. [1].*

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

Prior Work

Our prior work on this problem [1], which used the exact same dataset, first involved applying natural language processing techniques to transform the titles into formats usable for classification. Then, the resulting transformed titles and their category and subcategory labels were used as labeled data for the purposes of training supervised learning algorithms to predict the category and subcategory from a given title's textual information. The natural language processing techniques used in said prior work involved tokenizing the words of the title into features. The prior model utilized bigrams, which are pairs of adjacent words, as features. English stopwords were removed for the sake of cleaning the data. There were two supervised learning algorithms applied to the dataset: Support Vector Machine and Naïve Bayesian Classifier. The SVM approach required that not only should the title be transformed into individual words and bigrams, but also required that the words be embedded in a vector space using a bag-of-words approach [1]. This approach involved taking word and bigram frequency for a given title and using that as the value for each dimension in a vector space representing the unique words and bigrams present in the training set. The natural language processing techniques and this idea of a vector space model is reapplied in this work. The results for our prior work are shown in Figure 2 and Figure 3. The SVM approach was more accurate than the Naïve Bayesian approach, and is the standard to which we are comparing our current work.

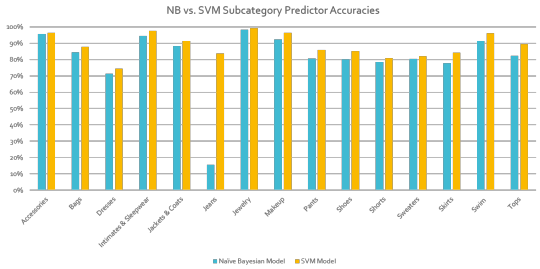


Figure 2. *Subcategory prediction accuracy results from our prior work [1].*

This prior work is also important for highlighting the limitations of these non-online supervised learning algorithms in an actual application. While they work well on the dataset given, non-online supervised learning algorithms have some failure cases in long-term applied settings. One particular failure case involves the introduction of new words. English has a constantly growing and changing vocabulary [2], and as such so does the fashion industry. When new words are introduced, the prior work lacks an method of embedding them properly, so the word has no contribution to the classification. Not only that, the non-online learning algorithms cannot do anything to fix the lack of knowledge or data regarding a specific word, which leads into the next failure case. If the use of a word changes in the applied context, the non-online learning approach cannot adapt to this change. For the Naïve Bayesian model, this is reflected in the fact hat the model does not store new data and adjust priors and posteriors accordingly in an applied setting. For the SVM, this is reflected in the observation that once a line and margin is fit to the training data, it does not change in the applied setting.

Fashion Trends and Conventional Learning Approaches

Variation of fashion trends occurs often in application. There is the cutting-edge of fashion, such as the runway, where new trends are introduced and either adopted by the general public or not. If the trend is adopted by the public, this results in changes in the natural model. There are two general time scales on which fashion trends can vary: short-run acceptance trends and long-run secular trends. Short-run acceptance trends are fashion trends that last on a scale of several months to years. Long-run secular trends can last from decades to cen-

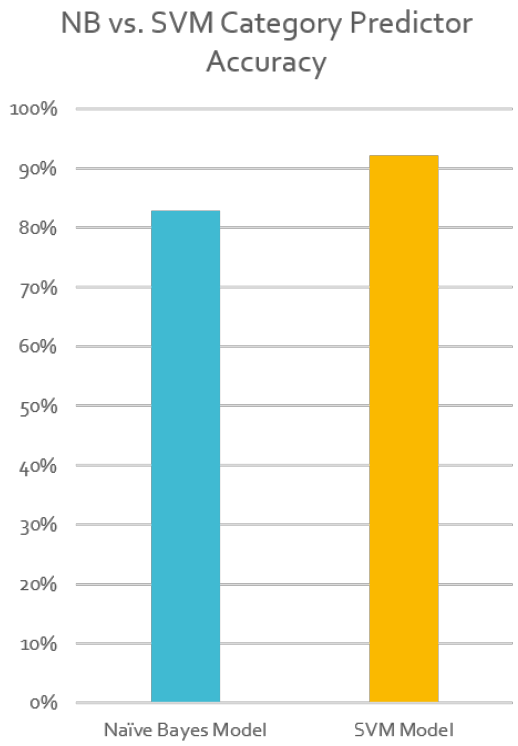


Figure 3. *Category prediction accuracy results from our prior work [1].*

turies [3]. There is a definite need to adapt to short-run acceptance trends, as these trends can crop up and disappear with relative frequency, and failing to adapt to these new trends could result in misclassified data in the application context. The need to adapt to long-run secular trends may seem trivial, but if one of these trends occurs after training and deploying the classification model, much of the models training could prove irrelevant in application. Furthermore, these two definitions for trends give a gauge on the scale at which the learning algorithm needs to adapt. It is desirable that the algorithm does not overfit and is robust to noise, but it needs to fit aggressively enough to predict short-run acceptance trends.

As an example to highlight the shortcomings of the conventional non-online approach applied to fashion item classification, examine the results from the Google search trend popularity of the term romper [4]. Figure 4 shows that prior to the year 2017, search queries for the term romper occurred infrequently because rompers were not trending until 2017. Then there is a noticeable but short spike in searches of the term frequency, which reflects the trending of the romper. In the application of categorizing fashion items, a non-online model trained on data prior to the rise in popularity of the romper will either have never encountered the term romper or will have encountered it so rarely that it will be prone to large variance and noise and will ultimately be a useless feature. This indicates that there are failure cases for the non-online learning approach, and those failure cases occur when new fashion trends crop up post-training.The non-online approach also fails when terms change meaning, as well as when new terms are introduced. This highlights the need for a model that can handle these shifting trends in an applied setting, with minimal to no retraining required.

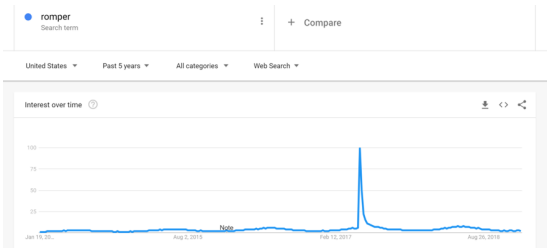


Figure 4. *Romper search term frequency from Google [4].*

Overview of Feature Selection Approaches

One of the biggest shortcomings of the prior work is a lack of ability to adapt to the introduction of new terminology in an applied setting. This is an issue that not only lies with the learning model, but also in how the titles are processed. The model has to be able to either represent new words using an already-known encoding scheme, or it needs to be able to handle newly encountered words as a new encoding entirely. Word-level features are informative. Generally speaking, the use of word-level features requires less data for training than that of other potential approaches such as character-level features. However, storing a scheme for vectorizing words can consume large amounts of memory, as this usually involves storing a vocabulary in order to convert words to dimensions in the vector space model. Without a vector space transformation mechanism, many learning algorithms are off-limits as they cannot deal with data in a non-numeric domain. Support Vector Machines would not have been applicable without a conversion method to turn the titles into a numeric vector. This presents a dilemma with a few potential solutions: try word-level features but only use a popular subset of all the words in the corpus, implement character-level features in order to make new words encode-able and therefore learn-able, or find another approach that uses word-level features without requiring the storage of a vocabulary.

Feature Filtration Approach

The idea of an approach that uses a popular subset of the training vocabulary seemed like a viable option. Prediction accuracy would be maintained if the features that were uninformative could be ignored, and the majority of new words encountered may not prove informative. This would also serve to improve the fit, as overfitting would become less of a problem. Some useful metrics for this removal of uninformative features include: chi squared, mutual information, information gain, term frequency, document frequency, TF-IDF, and corpus-wide term frequency. A combination of information gain and chi-squared, with an inverse relationship with mutual information was the approach proposed in [5]. The idea is that chi-squared and information gain will make sure informative features are selected, while the inverse relationship with mutual information serves to ensure that features do not have overlapping predictive effects on the classes. Features with high mutual information are redundant. It showed very promising results. However, this does not address the issue of encountering new words in application. While doing this could help reduce the vector space size significantly and maybe reduce the need to add new words frequently, there is still the need to handle new trends and words. If a new word or phrase proves highly informative based on the chosen filtering method, it needs to be added to the vocabulary. This means that unless uninformative features are removed, the vocabulary will eventually grow to capacity. Both adding new features and removing features requires that data be stored from the applied setting and analyzed. This is because, due to the volatile trends in the fashion industry, words that are significant for a given class may change. This means that the model must re-evaluate which words are informative as

new data is encountered, which does not meet the needs of the model.

Character-Level Neural Networks

Another approach, explored in [6], involved training a neural network to utilize character level encodings. The network used a combination of convolutional layers and RNN layers to effectively learn from training data. The use of convolutional layers, of varying sizes, helps to segment the character level features into words of meaning that the model can learn. The use of characters as the only type of feature means this model has a method for encoding words that have not yet been encountered. Due to the constraints Poshmark puts on the clientele, there is also a limit to the character length of a title. This means that each data point both in the training set and in the applied setting will have a finite number of features. This all lends the character-level neural net approach very well to the task at hand. However, the issue with this approach is that the number of data points required to train the model is too large for our training set. The amount of data needed for these neural net approaches is on the scale of 30,000 data points per class [7]. Our data per class is significantly less than that for most subcategories, which is why testing with this approach was unsuccessful. This issue also indicates how fast the model can learn and adapt in an online setting. Neural networks are fundamentally online learning algorithms, so neural networks can learn in an applied context. However, the scale of datapoints for a good fit shows the neural network cannot fit to trends fast enough. So, the online nature of the algorithm is wasted as it cannot learn fast enough to predict short-run secular trends. It is due to the slow learning ability of the neural nets and our own limited data that this approach does not work for our purposes.

Hashing Trick

The proposed featurization approach utilizes what is called the hashing trick [8]. This technique involves creating a feature vector with significantly more space than necessary, creating a hashing function accordingly, and using the hashing function to index the features into the feature vector. This method does not have a static vocabulary; all that is needed for this method to work is a feature vector and a hashing function. This is the perk of this approach. This approach does not need a vocabulary and therefore does not need to store new words, which makes learning and encoding new words tractable. This approach uses word-level features and has a means of encoding words not yet encountered in the training and applied setting. This is unique among all the other approaches, as they either required storing a vocabulary, or were not word-level feature approaches, and therefore had dataset size needs that could not be met.

In this application, the vector size needs to be large enough to account for new words. The size of the feature vector is important because if the feature vector size selected is too small, features will be likely to collide when hashed into the vector resulting in a loss of information. These collisions could be extremely detrimental in the situation where valuable and frequent words collide when hashed into the vector space. Or, alternatively, the collisions may not prove problematic if the words that end up colliding are not particularly informative or frequent. The safest approach is to choose a vector space size that will not result in too many collisions given the vocabulary size of the training data and the added space for new words. This was calculated using a hashing collision approximation [9], which involves taking the size of the vocabulary, squaring it, and making that the feature vector size. This leaves ample room for new words, as well as reducing the odds of a collision to a small probability.

Overview of Model

The model was implemented and tested using SCIKIT-LEARN [10]. First, the same preprocessing is applied as in the prior work: unigrams, bigrams, and removal of stopwords. After this preprocessing is performed, the term frequency of the features are embedded in the feature vector using the hashing technique, as covered in the prior section. Words and bigrams are indexed into a large vector array according to the given hashing function, resulting in a bag-of-words model where each dimension represents the term frequency of a given feature. The index of each feature is arbitrarily assigned and cannot be reversed. As long as the model is consistent with where a particular feature is represented in the array, the specific index of a feature is arbitrary; it is not necessary to be able to retrieve said index. This hashing processs runtime is gated by the size of the title in characters, which is constant due to the character length constraint placed on the title by Poshmark.

Passive-Aggressive Classifier

Next, an online learning algorithm is selected to do the classification for this task. This algorithm must be online because in an applied context, it must be able to learn in real time in order to adapt to new fashion trends. The selected online learning algorithm is the Passive-Aggressive Classifier [11]. The algorithm works as follows:

1. Predict: $\hat{y}_t = sign(w_t \cdot x_t)$
2. Receive ground truth: $y_t \in \{-1, +1\}$
3. Compute loss: $L_t = max\{0, 1 - y_t(w_t, x_t)\}$
4. Update: $w_{t+1} = w_t + \tau y_t x_t$, Where $\tau = min\left\{C, \frac{L_t}{\|x_t\|^2}\right\}$

How aggressively this learning approach fits to new data is controlled by a parameter C, known as the aggressiveness parameter. With this particular implementation, this parameter represents the maximum value update a model can perform on

weights, given an incorrect prediction. The larger the selected C, the more aggressively the weights can be adjusted based on the current incorrect prediction. If the prediction is correct, the algorithm does not adjust the weights; the model remains passive. If the prediction is incorrect, the weights are adjusted accordingly; the algorithm becomes aggressive. Hence the name of the learning algorithm: the Passive-Aggressive Classifier.

Classification Results

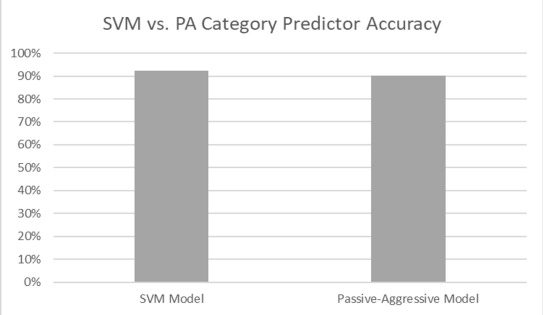


Figure 5. *This graph shows the category prediction accuracy of the SVM model from the prior work and the Passive-Aggressive model.*

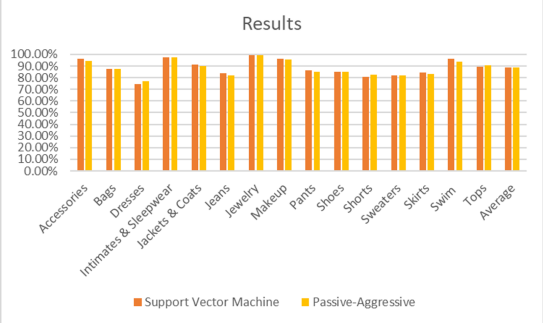


Figure 6. *This graph shows the prediction accuracy of the SVM model and the Passive-Aggressive model on each of the subcategory predictors.*

Figures 5 and 6 highlight that the Passive-Aggressive classifier performs just as well as the support vector machine classifier from the prior work. There are some subcategories where the SVM approach worked better, and some subcategories where the Passive-Aggressive approach worked better, but the differences are largely trivial. This means that the Passive-Aggressive classifier in a non-online setting can compete with the SVM, while also having the invaluable perk of being able to fit to data in the applied setting.

Training (s):	PA	SVM
Accessories	0.91	10.92
Bags	0.60	49.39
Dresses	0.42	89.64
I & S	0.18	3.58
J & C	0.23	10.04
Jeans	0.21	25.57
Jewelry	0.09	0.34
Makeup	0.95	107.10
Pants	0.31	35.05
Shoes	0.97	125.98
Shorts	0.07	5.31
Sweaters	0.10	15.54
Skirts	0.28	32.84
Swim	0.06	1.46
Tops	0.33	28.91
Category	0.13	0.85
Sum	5.85	542.51
Average	0.37	33.91

Table of training times in seconds for each predictor between the SVM and the Passive-Aggressive classifier.

Table 1 shows the training time of the model for the category predictor, each subcategory predictor, and the sum and average across all the predictors. The table shows that the Passive-Aggressive classifier trains much faster than the SVM. This means that if a situation arrises where the hashing function and feature vector size need to be updated, retraining the model can be very inexpensive and fast. This also shows the model is fast for learning from new data in an applied setting.

Discussion

The results presented show the power of the hashing trick in conjunction with the Passive-Aggressive classifier. This model performed just as well as the SVM model presented in the prior work in terms of accuracy, and substantially outperformed the prior model in terms of runtime. In this testing setting, however, the hashing trick is ultimately equivalent to the vector space transformation technique used in the prior work. This is because the introduction of new words is hard to emulate given the dataset we have to work with. It is also difficult to emulate changing fashion trends given the dataset we have to work with. Even without factoring in the model's ability to learn from incorrect predictions on new data, the approach is competitive with the prior work in a fully offline setting and outperforms the prior work in terms of runtime.

Optimizing for Fashion Trends

In order to optimize this model for online learning in a fashion context, some form of study or analysis needs to be done

to optimize the algorithm for prediction of fashion trends. The C parameter, or aggressiveness, of the algorithm is what will mainly control for how aggressively the model will fit to misclassified data. This is the parameter that ultimately needs to be analyzed for optimizing the rate at which the model fits to newly encountered data. If this parameter is too large, the model will be prone to fitting to noise that is not actually a trend. If this parameter is too small, the model may fail to fit to short-run acceptance trends. An optimal choice for this aggressiveness parameter could be determined using temporal studies where data from Poshmark is collected over a long period of time, with time-stamps to indicate when a datapoint was collected. The data could then be partitioned according to date, and prediction performance could be assessed for each time period. This would require a large amount of resources regarding data collection and time. Mathematical approaches for formalizing fashion trends could help to find the optimal aggressiveness parameter to fit to new data [12], and are worth further exploration. However, yet again, in order to test whether the mathematical formalization of fashion trends resulted in an optimal C parameter would require testing on time-stamped data for verification. The issues with time and data collection are not entirely averted.

Conclusion

The Passive-Aggressive classifier model presented shows competitive performance in terms of accuracy with the prior SVM-based model. The current model heavily out-performs the prior work in terms of runtime, and also benefits from utilizing the hashing trick to reduce the amount of space consumed. This means the model has a larger variety of feasible implementation environments. This model also has the ability to adapt to new words and trends in the applied setting because the hashing trick is capable of encoding words that have yet to be encountered; and the Passive-Aggressive classifier utilizes an online learning approach. This model successfully addresses the problem of fitting to volatile fashion trends in the applied setting without manual intervention. Further work should be done to determine the optimal aggressiveness for the model so that the it does not overfit and predict noise, but also successfully fits to short-run acceptance trends.

Acknowledgments

We thank Poshmark Inc. for their continued support of our research project.

References

[1] K. Norman, Z. Li, Y.-T. Oh, G. Golwala, S. Sundaram, and J. Allebach, "Application of natural language processing to an online fashion marketplace," *Imaging and Multimedia Analytics in a Web and Mobile World 2018*, (Part of IS&T Electronic Imaging 2018), J. Allebach, Z. Fan, and Q. Lin, Eds., San Francisco, CA, 28 January -2 February 2018.

[2] L. Bromham, X. Hua, T. G. Fitzpatrick, and S. J. Greenhill, "Rate of language evolution is affected by population size," *Proceedings of the National Academy of Sciences*, vol. 112, no. 7, pp. 2097–2102, 2015. [Online]. Available: https://www.pnas.org/content/112/7/2097

[3] G. Sproles, "Analyzing fashion life cycles - principles and perspectives," *Journal of Marketing*, vol. 45, no. 4, 1981. [Online]. Available: http://search.proquest.com/docview/1296563457/

[4] trends.google.com. (2012) Google trends. [Online]. Available: http://trends.google.com/trends

[5] W. Zong, F. Wu, L.-K. Chu, and D. Sculli, "A discriminative and semantic feature selection method for text categorization," *International Journal of Production Economics*, vol. 165, pp. 215–222, 2015.

[6] J. Liu, F. Meng, Y. Zhou, and B. Liu, "Character-level neural networks for short text classification," in *2017 International Smart Cities Conference (ISC2)*. IEEE, 2017, pp. 1–7.

[7] Q. Hua, S. Qundong, J. Dingchao, G. Lei, Z. Yanpeng, and L. Pengkang, "A character-level method for text classification," in *2018 2nd IEEE Advanced Information Management,Communicates,Electronic and Automation Control Conference (IMCEC)*. IEEE, 2018, pp. 402–406.

[8] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," *CoRR*, vol. abs/1607.01759, 2016. [Online]. Available: http://arxiv.org/abs/1607.01759

[9] K. Suzuki, D. Tonien, K. Kurosawa, and K. Toyota, "Birthday paradox for multi-collisions," in *Information Security and Cryptology – ICISC 2006*, M. S. Rhee and B. Lee, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 29–40.

[10] 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.

[11] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," *J. Mach. Learn. Res.*, vol. 7, pp. 551–585, Dec. 2006. [Online]. Available: http://dl.acm.org/citation.cfm?id=1248547.1248566

[12] C. Miller, S. Mcintyre, and M. Mantrala, "Toward formalizing fashion theory," *Journal of Marketing Research*, vol. 30, no. 2, 1993. [Online]. Available: http://search.proquest.com/docview/1297380096/

Author Biography

Kendal Norman is a Senior in his undergraduate program studying Computer Science with a focus on Machine Intelligence at Purdue University, West Lafayette. he is also working on a minor in Mathematics. 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 an active member of University Choir in Purdue Musical Organizations since Fall 2015. He will be attending Purdue University in Fall 2019 to pursue his Masters degree. He plans to pursue a PhD and ultimately become a Professor of A.I. Theory.

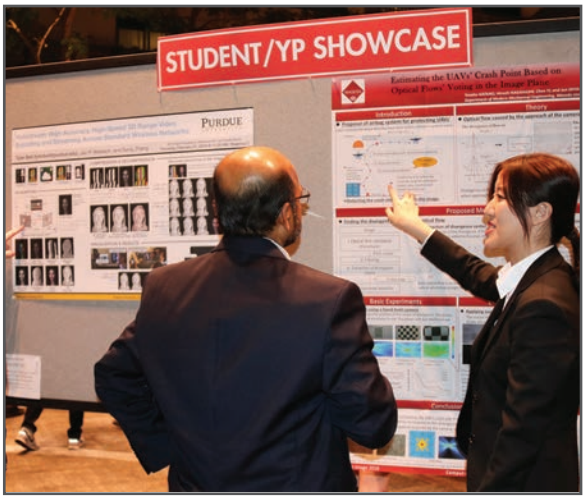
JOIN US AT THE NEXT EI!

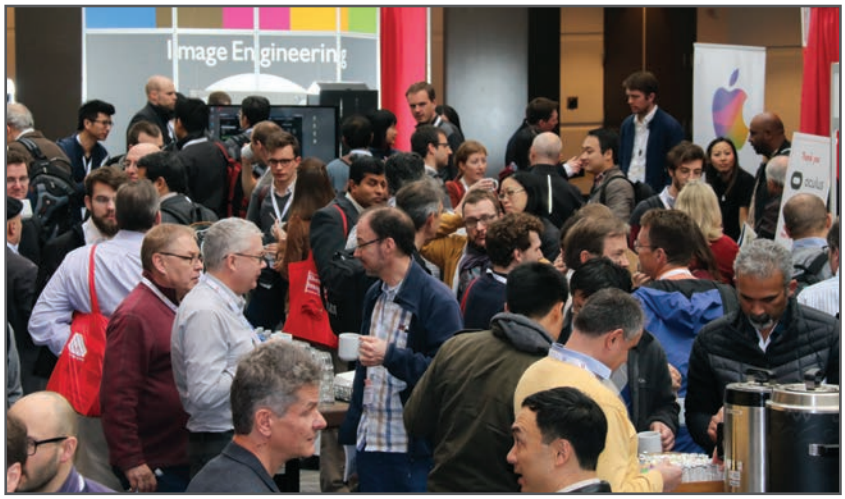
IS&T International Symposium on

Electronic Imaging

SCIENCE AND TECHNOLOGY

Imaging across applications . . . Where industry and academia meet!





• SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •
• INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •

www.electronicimaging.org

IS&T
imaging.org