# What's the Deal? – Identifying Online Bargains

**John Cuzzola, Dragan Gašević, Ebrahim Bagheri**
Athabasca University, Ryerson University
{jcuzzola | dragang}@athabascau.ca, bagheri@ryerson.ca

## Abstract

The Internet is home to an ever increasing array of products and services available to the general consumer. This trend has given rise to a unique category of internet search where bargain seekers have conjugated towards deal collection databases. This is caused, in part, because traditional internet search engines do not perform well in this domain. Unfortunately, these deal databases are costly to maintain due to the heavy reliance on human participation in order to populate them. This has lead to an interest in the development of this class of internet search. Our research focuses on leveraging machine learning and natural language processing to develop a semi-supervised Web page classifier specific to this problem. We describe the design of our classifier with respect to the machine learning model chosen and the training features selected. We compare our model's effectiveness in classifying deal versus non-deal Web pages against other popular machine learning models such as decision tree, support vector machines, and neural net. Our results show that our proposed model performed the best given the features that were extracted for model training and testing.

*Keywords*: natural language processing, classification, Naïve Bayes, deals, products, web page classification.

## 1. Introduction

The World Wide Web has given rise to a digital marketplace where goods and services of all varieties are sold. This arena is no longer the domain of solely traditional brick and mortar retail outlets. Forrester research predicts, by 2016, Americans will spend $327 billion via e-commerce; an increase of 62% from 2011 statistics [1]. Perhaps the greatest indicator of this phenomenon is the emergence of deal collectors and deal aggregation services. Deal collector sites, such as GROUPON, have staffed 10,000 employees to locate special product offers that bargain hunters are on constant lookout for [2]. A plethora of such sites have led to the creation of deal aggregators – sites that track bargains found by multiple deal collectors. Even Google, arguably the reigning king of search engines, have their own deal locator service known as *Google Offers*.



However, even Google's dominance in information retrieval have yet to extend to Google Offers which is still in its infancy with only a beta deployment to a handful of cities. This suggests there is still an opportunity to make a significant impact in this category of web search.

Our main contribution in this paper is related to the challenge deal collectors/aggregators face with the heavy reliance on human intervention to find these bargains. Automation is difficult due to the unstructured nature of the web which presents problems for computers. Our goal is to find some method of semi-automatic classification specific to this domain. Our main contribution is a lightweight classifier capable of performing this category of web search to a satisfactory level through the combination of a Naïve-Bayes based algorithm and statistical probability. We also demonstrate how our algorithm can be used for webpage ranking as well as classifying. Our main contributions in this paper can be enumerated as follows:

- We have developed a classification mechanism that is able to consider various textual features of a Web page and determine whether the page contains information on daily deals and offers.
- We have collected a large set of features from Web pages including WordNet references, Named-Entity Recognition and Part-of-Speech tagging and evaluated their effectiveness for Web page classification in the area of daily deals.

The rest of the paper is organized as follows: In Section 2, we describe a classifier that determines whether a webpage contains information related to daily deals or not. Section 3 evaluates the effectiveness of the classifier from three distinct aspects, while related works (§4), future work (§5), and concluding remarks (§6) round off the paper.

## 2. Architecture Overview

Our industrial partner, SideBuy Inc., is a daily deal aggregator who has invested in intelligent techniques for gathering deal information from the Web. In what follows, we review the overall contribution that we have made to their architecture. The primary actors of our technology are comprised of intelligent agents, an internally developed AI library, and SideBuy staff working together in a semi-supervised model in order to find potential Web pages that contain daily deal information. The Agents function as web-crawlers that roam the Web either independently or as

directed by staff to specific target sites. The agents determine a classification of the target webpage as either 'deal' or 'no-deal' using an AI library that provides a combined Naïve Bayes classifier with Expectation-Maximization (NB/EM). The Web page is then indexed and verified correct by staff before inclusion into their master database, which includes all Web pages that provide some form of daily deal information.

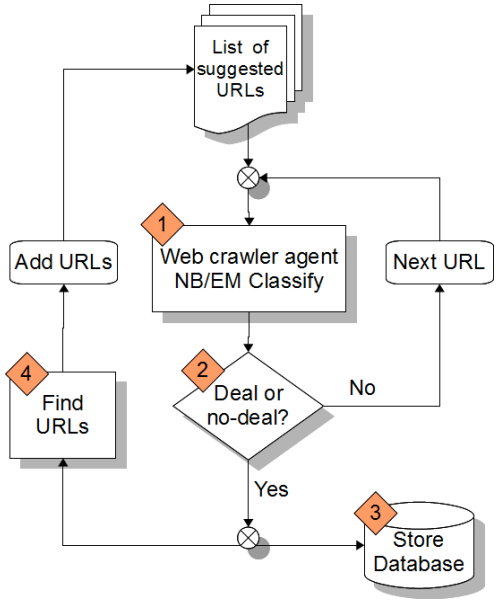


Figure 1. deal webpage classification in four steps

The overall process is outlined in Figure 1. First, a webpage is retrieved (1) and is classified as either a page containing deal information (deal) or a page with no deal related information (no-deal) (2). Deal pages are stored in the database for later use by SideBuy customers (3) while the crawler follows links for leads on possible more deals (4).

In order to build the classifier for labeling Web pages as deal or no-deal pages, we will first need to define a set of features for each page that would serve to describe the page and offer the grounds for building the classifier. The features are described in the following sub-section.

### 2.1. Feature Selection

Features extracted from each Web page are shown in Table 1. We incorporate lexical databases, such as WordNet, and natural language processing techniques to obtain the features. The reasons for the choice of features were manifold. Foremost, the individual occurrence of frequently appearing words was selected for obvious reasons particularly in the context of using a text classifier Naïve Bayes. However, we augment the frequency counts with the use of a lexical database WordNet to include words with similar meanings (synonyms) or closely related words to increase the likelihood of exposure to words that may appear in the wild but were not seen during training. We also utilize named entity recognition to capture semantics such as currency, percentage, organizational entity and so forth. Part of speech tagging is also used to include features that identify simple counts such as average dollar value of a sentence block and number of symbols (such as punctuation marks) in the block. During the selection process, we looked for features that were quick to tabulate with little computational overhead. Because an intelligent agent is tasked to spider many sites in an efficient manner – a lightweight feature set was imperative. Thus, deeper analysis techniques such as semantic role labeling were avoided due to their time and computational requirements although a limited use of these tools could be incorporated [19].

Table 1. Features extracted for training and testing (ALL)

| FEATURE | DESCRIPTION | SOURCE |
|---|---|---|
| A. Words | The words in the sentence block. WordNet lexical database is used to lemmatize. | WORDNET |
| B. ner_dateI<br>C. ner_organizationI<br>D. ner_timeI<br>E. ner_locationI<br>F. ner_percentageI<br>G. ner_moneyI<br>H. ner_personI | Number of [dates, organization entities, time, locations, percentages, money values, and person] instances as identified through named entity recognition. | NAMED ENTITY RECOGNITION (NER) |
| I. sym_dollarAvgI<br>J. sym_percentAvgI<br>K. sym_CD_posI<br>L. sym_SYM_posI | The average dollar value, average percentage, count of numerical values, and count of symbols, as identified through part-of-speech tagging. | PART OF SPEECH TAGGING (POS) |

### 2.2. The NB/EM Classifier

Once the features were identified, a Naïve Bayes (NB) classifier was used to determine the label of a web page as either deal or non-deal. We combine NB text classifier with Expectation-Maximization (EM) clustering for this task. The pairing of Naïve Bayes with EM is common [15] and offers advantages to using Naïve Bayes alone. EM allows for the discovery of clusters whose attribute distributions either lean toward deal or no-deal status. It is an unsupervised learning technique that is often utilized in supervised and semi-supervised applications as well. In the case of supervised learning, EM mitigates for an unbalanced training set of positive and negative examples [14]. It also allows for incomplete training samples with missing or unknown attributes. Despite the independence assumption, NB classifiers often perform well with text [16]. They are relatively easy to deploy and are fast classifiers. Speed is an important consideration in our model of the intelligent agent web crawler, which must quickly scour the Internet for products whose availability and pricing can change frequently.

We use both positive and negative examples of deal and no-deal web pages to train the classifier. The corpus for positive training samples was readily obtained through SideBuy.com's existing indexed database of deal offerings. Negative training examples were obtained through texts available under the Creative Commons license, or public domain through repositories such as *Project Guttenberg*.

Training and testing candidates are preprocessed by stripping of HTML tags leaving behind only the text (content of the web page). This text is then split into sentence blocks using sentence detection available through the OpenNLP machine learning framework. These sentence blocks, as determined by OpenNLP, are utilized as training/testing samples.

Although acquiring sufficient numbers of negative samples for machine learning training is sometimes a challenge, this classifier operates at the sentence level; making it easier to obtain negative samples since a single story (from Project Guttenberg for example) can contain thousands of sentences. Our procedure transforms a web page into similar sentence blocks suitable as counter-examples.

Once trained, a probability is assigned to each sentence block indicating the likelihood the block is consistent with what would be seen in a deal-like web page. The Naïve Bayes classifier calculates the probability that a sentence block belongs to each EM cluster and then a weighted average across all clusters completes the calculation. Formally, given n-clusters ($C_n$), discovered through EM learning and (f) features ($F_f$) of sentence block (S), the probability of (S) belonging to a cluster ($C_i$) denoted $P(C_i|S)$ using Naïve Bayes is:

$$P(C_i|S) = \alpha P(C_i) \prod_{1..f}^{j} P(F_j|C_i) \qquad (1)$$

where $\alpha$ is the normalization constant.

A sentence block is classified as consistent with containing deal-like content if the sum of the likelihood of being a deal within all EM clusters exceeds a set threshold $\tau$:

$$\sum_{1..n}^{i} P(C_i|S)P(F_{deal}|C_i) > \tau \qquad (2)$$

Table 2 shows examples of classified sentences. We have decided that sentences with a probability of over 90% ($\tau$) can be labeled as 'deal'. Later, we provide empirical evidence to support the value for this threshold.

Table 2. Two example sentences with probability of being in a deal web page.

| SENTENCE BLOCK | PROBABILITY | >90% |
|---|---|---|
| Buy unlimited vouchers as a gift Package includes a 7" Google Android 2.3 Tablet with a 30 pin USB switch adaptor , charger and user manual Lightweight and easy to use Perfect idea for people on the go Makes a great gift ! | 0.9998 | Yes |
| Challenges address the conceptualization how e-business related knowledge is captured , represented , shared, and processed by humans and intelligent software. | 0.0248 | NO |

*2.3. Training/Testing sets and the Ensemble Method*

We employ the well known ML technique of ensemble voting in order to improve the classifier's accuracy. This method involves training multiple independent classifiers with different, but perhaps overlapping, training sets. Each classifier provides their own probability calculation to individually determine deal or no-deal. The final class label is achieved by majority vote of the participating classifiers thus improving overall accuracy by consensus.

The ratio of deal sentence blocks to no-deal sentences is compared to a threshold value to determine final deal/no-deal classification of the web page. We also include a sanity check where if the webpage meets this threshold but does not contain any monetary artifacts, as determined by named entity recognition, then the webpage must be classified as 'no-deal'. This is to filter out those sites that describe a product, but are not selling the product. Examples of this are vacation blogs that describe seasonal travel packages available and product review pages. By disabling this sanity check, the classifier can be extended to identify general product pages whether or not they are for purchase. Section III demonstrates the overall effectiveness of this technique.

## 3. Experimentation and Evaluation

In this section, we evaluate the proposed implementation and offer empirical results. In our evaluations we examine the effectiveness of the NB/EM classifier against three other common machine learning methods including SVM, NN, and J48.

To evaluate the effectiveness of the Naive Bayes (NB)/Expectation-Maximization (EM) model, we compared its classification accuracy at the sentence block level with three other popular machine learning models: support vector machine (SVM), decision tree (J48), and neural networks (NN). Together these four models comprise a broad range of general machine learning categories: probability (NB/EM), optimization problem (SVM), graph model (J48) and activation function (NN). In addition, two different vector normalization techniques were investigated for training: linear scaling, and z-score normalizing. The commonplace radial basis function kernel and sigmoid activation function were used for the SVM and NN models. The NN model was constructed with n-input neurons (one neuron for each attribute of the input vector), one hidden layer with n/2 neurons and a single output neuron to indicate boolean deal or no-deal. For J48, the Weka machine learning collection provided the implementation [3]. For SVM, the LibSVM library provided the functionality [4] while the Neural Net Framework (NNF) was used for the NN model [5].

Random sampling of 4,000 sentences from the SideBuy corpus of 1.6 million sentences became the training set of vectors with each vector comprised of the 12 features listed in Table 1. These 4,000 were equally divided between deal and no-deal sentences. For SVM and NN a sparse vector where each attribute corresponds to the encountered frequency of a recognized word or to a computed feature of Table 1 was constructed. The results of training with ten-fold crossover validation are shown in Table 3.

Table 3. Sentence classification accuracy of various models using ten-fold crossover validation.

| | NB/EM | J48 | SVM z-score | SVM linear scaling | NN z-score | NN linear scaling |
|---|---|---|---|---|---|---|
| Accuracy % | 96 | 88.7 | 77.33 | 51.39 | 48.3 | 48.3 |

The NB/EM model performed best followed by J48 and SVM/z-score. There was a noticeable improvement of SVM when trained with z-score normalized vectors versus linear scaling of the features. NN performed the worst regardless of which normalization method was used. These trained models were then tested against the SideBuy database. Ten rounds of samples of 600 sentences (300 deal/300 no-deal) were selected at random with replacement. Their individual classification accuracy was averaged with results given in Figure 2. Once again, NB/EM performed best with J48 a close second. SVM/z-score had a better than average accuracy where as SVM/linear, NN/z-score and NN/linear struggled.

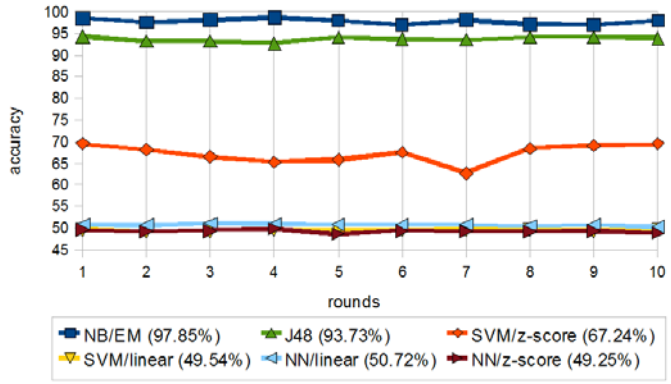


Figure 2. Average classification accuracy for ten rounds of randomly sampled sentences

It would appear obvious that the word feature would be an important attribute of the model. Particularly in this domain, words such as “deal”, “save”, “purchase”, and “discount” should weigh heavily on any model’s classification decision. In the next series of tests, the models were retrained with two different sets of features. The first set consisted of only the words feature while the second set contained all features except word. This word versus the-rest test was performed to determine the impact of this core attribute (word). Results are given in Table 4.

Table 4. Word-only versus the-rest. Average of 10 rounds of random sampling.

| | NB/EM | J48 | SVM z-score | SVM linear scaling | NN z-score | NN linear scaling |
|---|---|---|---|---|---|---|
| word (only) | 97.52 | 79.68 | 64.58 | 49.47 | 48.6 | 49.55 |
| the-rest | 89.19 | 88.27 | 89.07 | 49.5 | 52.6 | 49 |

When trained only with the word feature, the NB/EM model was equally effective as it was with all 12 features available. NB/EM, J48, and SVM/z-score achieved similar accuracy when trained without the word feature (the-rest). This may suggest that the-rest attributes are unnecessary and thus the model feature complexity can be reduced to the single feature. However, this is not the case as this suggestion presumes the test sentence always contains familiar words. These results demonstrate that the NB/EM model, relying on its other 11 features, can determine a sentence classification to 89% accuracy even when no recognizable words are present.

Of Interest is the observed accuracy results given in Tables 3 and 4, and Figure 2 across the various models. The disparity in accuracy between SVM and NN models may be attributed to numerous conditions. Both SVM and NN model representations are significantly different than NB/EM and J48. Specifically, SVM/NN used a sparse vector of attributes where each attribute position corresponds to a recognized word (A) and its frequency count plus an additional 11 attributes for the-rest (ALL-A) features. This results in a large vector of attributes based on the encountered words during training. For example, for the sampling of 4,000 sentences, the vector averaged 1,811 attributes (1,800 unique words plus 11 static the-rest attributes). In contrast, NB/EM and J48 models can represent the word feature in a single attribute thus having a simpler model representation of a fixed 12-attribute vector. Although this sparseness does not necessarily represent a problem, particularly for SVMs where sparse feature vectors are commonly used, this situation presents a few considerations of its own. First, the size of the training set may need to be larger in order to produce sufficient unique vectors to adequately train the model. Furthermore, SVMs operate by finding a maximal separating hyperplane across multiple dimensions. A 1,811-attribute vector requires a separation plane for 1,811 dimensions, thus potentially requiring a larger training set to achieve a well-represented separation. Second, the importance of vector normalization is well-known in such ML models hence the significant impact observed in accuracy with the change of normalization methods: linear versus z-score. These considerations appear to have been realized in Table 4 with the removal of the word feature. In this test, the vector attribute length shrunk from 1,811 down to a fixed 11 (the-rest);- resulting in identical vector of attributes for NB/EM, J48 and SVM. This reduction, combined with z-score normalization, gave SVM the same level of accuracy as NB/EM and J48.

Comparitively, the NN model may benefit from a combination of different selection of parameters such as a change in activation function, number of hidden layers, number of neurons per layer, adjusted learning rate as well as perhaps a different normalization method and larger sample training size. Further investigation is needed but the number of model parameter adjustments necessary make this model difficult to tweak.

## 4. Concluding Remarks

In this paper we described our algorithm for Web page classification for a specific category of Web content – daily deal identification. Empirical testing showed our combined model of Naïve Bayes and Expectation/Maximization

performed well in comparison with other machine learning methods. We also demonstrated how our model can be used for sorting and ranking in addition to binary deal/no-deal classification. Our future research will build upon this work with the goal of creating a system capable of identifying, extracting, and mapping properties-to-products from unstructured natural language Web page sources.

## 5. References


[1] (Leggat, H. (2012) Forrester: Online spending to reach $327 billion in 2016. Retrieved from: http://www.bizreport.com/2012/02/ forrester-online-spending-to-reach-327billion-in-2016.html

[2] Ghigliotty, D. (2011) Do You Really Want a Job at Groupon? Retrieved from http://sales-jobs.fins.com/Articles/SBB00014240529 70204528204577012073472414832/Do-You-Really-Want-a-Job-at-Groupon

[3] Hall, M. Eibe, F., Holmes, G. Pfahringer, B., Reutemann, P. Witten, I. (2009) The WEKA Data Mining Software: An Update. SIGKDD Explorations, Volume 11, Issue 1.

[4] C.-C. Chang and C.-J. Lin. (2011) LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27.

[5] Presta, A. (nd.) NNF2 a C++ library for neural networks http://nnf.sourceforge.net/

[6] Shen, D., Zheng, C., Qiang, Y., Zeng, H., Zhang, B. , Lu, Y., Ma, W. (2004). Web-page classification through summarization. In Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '04). ACM, New York, NY, USA, 242-249

[7] Yu, H., Han, J. Chang, K. (2002). PEBL: positive example based learning for Web page classification using SVM. In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '02). ACM, New York, NY, USA, 239-248.

[8] Qi, X., Davison, B. D. (2009). Web page classification: Features and algorithms. ACM Comput. Surv. 41, 2, Article 12 (February 2009), 31 pages.

[9] Selamat,,A., Omatu, S., Yanagimoto, H., Fujinaka, T., Yoshioka, M. (2003) Web page classification method using neural networks. IEEJ Transactions on Electronics, Information and Systems, 123 (5). pp. 1020-1026.

[10] Fiol-Roig G., Miró-Julià, M., Herraiz, E. (2011) : Data Mining Techniques for Web Page Classification. PAAMS (Special Sessions) 2011: 61-68

[11] Li, F., Yang, Y. (2005). Analysis of recursive feature elimination methods. In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '05). ACM, New York, NY, USA, 633-634.

[12] Hsu, H., Lu, M.; (2008) Feature Selection for Cancer Classification on Microarray Expression Data, Intelligent Systems Design and Applications, 2008. ISDA '08. Eighth International Conference on , vol.3, no., pp.153-158, 26-28

[13] He, X., Duan, L., Zhou, Y., Dom, B. (2009). Threshold selection for web-page classification with highly skewed class distribution. In Proceedings of the 18th international conference on World wide web (WWW '09). ACM, New York, NY, USA, 1081-1082.

[14] Tsuruoka, Y., JTsujii, J. (2003). Training a naive bayes classifier via the EM algorithm with a class distribution constraint. In Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4 (CONLL '03), Vol. 4. Association for Computational Linguistics, Stroudsburg, PA, USA, 127-134

[15] Calders, T.G.K. & Verwer, S.E. (2010). Three naive Bayes approaches for discrimination-free classification. Data Mining and Knowledge Discovery, 21(2), 277-292.

[16] Kim, S., Seo, H., Rim, H. (2003). Poisson naive Bayes for text classification with feature weighting. In Proceedings of the sixth international workshop on Information retrieval with Asian languages (AsianIR '03), Vol. 11. Association for Computational Linguistics, Stroudsburg, PA, USA, 33-40.

[17] Guyon, I., Elisseeff. A. (2003). An introduction to variable and feature selection. Journal of Machine Learning Research. vol 3. 1157-1182.

[18] St. Pierre, D. (2010) What Are Folksonomies and Why You Need Them. Retrieved from http://www.cybergenica.com/blog/ business-post/what-are-folksonomies-and-why-you-need-them

[19] Ciaramita, M., Attardi, G., Dell'Orletta, F., Surdeanu, M. (2008) DeSRL: a linear-time semantic role labeling system. In Proceedings of the Twelfth Conference on Computational Natural Language Learning (CoNLL '08). Association for Computational Linguistics, Stroudsburg, PA, USA, 258-262.

[20] Suchanek, F., Kasneci, G., Weikum, G. (2007) YAGO: A Core of Semantic Knowledge Unifying WordNet and Wikipedia. Proceedings of the International World Wide Web Conference.