**Josh Grinberg**
**Adam Perelman**
**CS229 Fall 2013**

# Nail the Sale:
# Predicting Sales Outcomes with Textual Features

**Abstract**

We analyze the use of textual features in predicting whether a company's open sales opportunities will win or lose. A simple model involving bag-of-words features and logistic regression performs well on this classification problem, approaching the accuracy of a human benchmark and outperforming a variety of more complex features and models.

## 1 Introduction

Machine learning is increasingly playing a role in helping businesses to predict the future. In particular, we see the potential for machine learning to help sales departments predict the success or failure of their open sales opportunities, and thereby allocate time to opportunities more efficiently. For this project, we worked with Roam Analytics, a startup which uses data from clients' CRM systems, such as Salesforce, to help predict future sales. For each open sales opportunity in the CRM, Roam predicts 1) whether the sale will close successfully; 2) when the sale will close; and 3) how much the sale will be worth. Roam uses a variety of features, such as past outcomes for particular salespeople, to make these predictions.

CRM systems often contain large amounts of textual data related to open sales opportunities, such as emails between salespeople and their clients and notes added to the CRM by salespeople. However, the Roam platform does not yet leverage this textual data in its models. In this paper, we analyze the use of these textual features to improve the quality of predictions made by Roam.

## 2 Data

Our dataset for this project represents the sales pipeline for a rapidly-growing technology company among Roam's clients, which we refer to as SalesCo.

Text data in a CRM system falls into a variety of different categories. Occasionally, salespeople choose to record email conversations in the system. A sample email from the SalesCo database contains the following excerpt (note that names have been changed):

> *Adam,*
> *It was a pleasure talking to you. Thank you for taking the next step. I look forward to our next meeting with Kyle in the next few days. Here are some information sources that will help you get started with the evaluation phase.*
> *1) CompetitorCo to SalesCo comparison.*
> *2) PowerPoint from today's meeting (see "notes" for more information).*
> [...]
> *Talk to you soon,*
> *- Josh*

In addition to these logged email conversations, salespeople sometimes write private notes about sales opportunities in the CRM system. Consider the following sample note from the SalesCo database (names modified again):

> *Rob seemed to lead the discussion on their end. Larry and John on the call too.*
> *Showed them how to use SalesCo to:*
> *- Find prospect companies*
> *- Find people*
> *- Find connections (got excited about this, LinkedIn in particular)*
> *- Track business events*
> *Rob wanted until Thanksgiving. Told him he needed to talk to Ken.*

The size of dataset is summarized in Table 1.

| Opportunities with text | Text data per opportunity | Tokens per text datum |
|---|---|---|
| 2,349 total<br>1,028 wins<br>1,321 losses | 3 (median)<br>6.16 (mean) | 172 (median)<br>372.0 (mean) |

Table 1

## 3 Models and Results

The ultimate objective of Roam is to make predictions about the outcome of open sales opportunities. For this project, however, we decided to tackle a simpler classification problem: given an opportunity which has already closed, we attempt to predict whether it closed with a win or a loss, based on the text associated with that opportunity.

### 3.1 Human Benchmark

To set a performance benchmark, we tried predicting sales outcomes by hand and recording our performance, similar to the human benchmark used by Pang, Lee, and Vaithyanathan for movie review sentiment classification (2002). Since understanding nuances, sentiment, and intention in email text is complex, we consider the performance of human predictions to be a reasonable goal for our problem. We obtained the following results by reading through text for 44 opportunities.

| | Actual win | Actual loss |
|---|---|---|
| **Predicted win** | 19 (true positive) | 4 (false positive) |
| **Predicted loss** | 6 (false negative) | 15 (true negative) |

Table 2: Confusion matrix for human classification benchmark

In order to best capture and compare the results of our predictions, we decided to use the Matthews Correlation Coefficient (MCC), which represents the correlation between observed and predicted binary classifications and is often considered a better measure of performance than precision or recall alone (Matthews, 1985). It is computed from a confusion matrix of true positives, true negatives, false positives, and false negatives as follows:

$$\mathrm{MCC} = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})}} \quad \text{(Eq. 1)}$$

The value +1 represents a perfect prediction, 0 is equivalent to random guessing, and -1 represents total disagreement between prediction and reality. Based on the confusion matrix above, the MCC of our human classification is 0.545.

### 3.2 Learning Algorithms

As a simple initial model, we used a bag-of-words feature matrix for each opportunity with text, defining an opportunity's text to be the concatenation of all its associated textual data. Feature engineering is discussed in more detail below. We trained several different learning algorithms on the SalesCo dataset and used k-fold cross-validation with varying amounts of training examples to compare each algorithm's performance. The following plot illustrates performance (MCC) of each algorithm on a testing set.

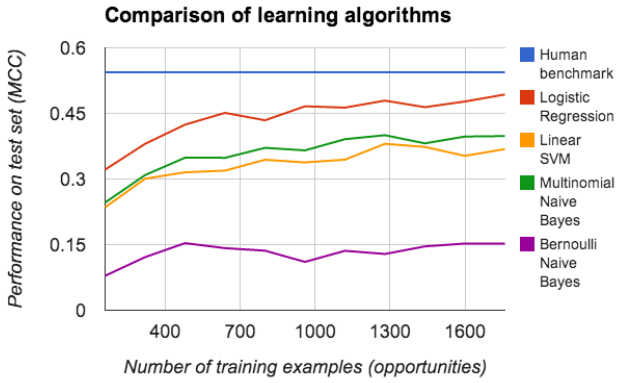


Figure 1

We also ran SVMs with nonlinear kernels (not shown), which yielded extremely poor results, with MCC scores close to 0 (equivalent to random guessing). We speculate that nonlinear SVMs did not perform well because we had roughly 10,000 features (distinct word tokens) but only 2,000 training instances (opportunities), and kernels associated with higher-order features spaces were more prone to overfitting than the simpler linear kernel.

Logistic regression consistently outperformed all other models, with a maximum MCC of 0.482, achieving both precision and recall of 0.75 with the optimal feature set discussed below (learning curve in Figure 2). By considering the model parameters with the most positive and most negative values, we determined which tokens are most strongly correlated with wins and losses, as summarized in the table below (including bigrams, discussed in the feature engineering section below). Since some of these tokens, such as "signed" and "approval", are likely part of text added after a sales opportunity already closed, it is possible that transitioning our model to predict outcomes for still-open opportunities may pose a challenge.

| | |
|---|---|
| **Tokens predictive of wins** | order, renewal, signed, approval, upsell, training, reached, service, today, fu email (*note: shorthand for "follow-up email"*), team, contract, invoice, trial, order form, sent feedback, usage |
| **Tokens predictive of losses** | revisit, follow, demo, called, cancellation, wish, alternative, exchange, follow called, nvm, deal, longer, 10 minutes, company, months, fit, speed |

Table 3: Tokens strongly correlated with wins or losses

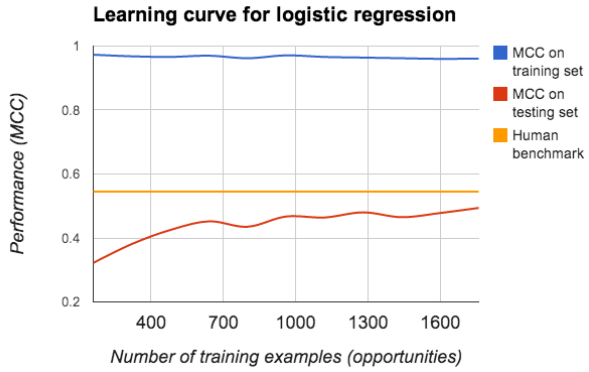


Figure 2

The large gap between the training and testing curves suggests that our model overfit the training data considerably. Since the gap narrows as we use more training examples, we believe data scarcity is an issue for our algorithm.

### 3.3 Features

#### 3.3.1 Bag-of-Words Variations

The logistic regression classifier discussed above yielded surprisingly good performance on a simple bag-of-words feature set (MCC of 0.387, and precision and recall of almost 0.70), with each column in the feature matrix mapping to a specific token, and each entry representing the count for a particular token in the corresponding opportunity. We experimented with stemming tokens and with removing stop words, as described by Mladenic (1999), but neither of these significantly changed performance.

We experimented with minimum document frequency, excluding tokens that appeared in our training corpus less than a minimum threshold. Although we expected that this would reduce overfitting, this in fact led to worse performance, suggesting that even rare tokens in our dataset can be effective indicators of wins and losses.

Some minor modifications of this bag-of-words model, however, led to improvement in results. Using binary features, for token presence or absence, proved more effective than using token counts, improving our MCC score from 0.387 to 0.434. Including bigrams and trigrams of tokens in our feature vector further improved our classifier, increasing our MCC score to 0.482. Including n-grams longer than 3, however, decreased MCC score on the test set, although not on the training set, suggesting that this increased overfitting.

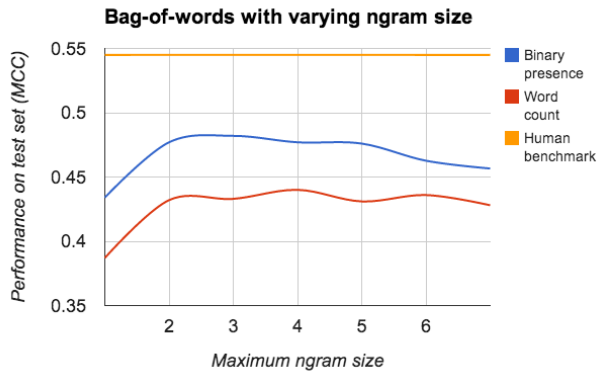


Figure 3

### 3.3.2 Text Classification

Text associated with a particular opportunity usually comes from one of three categories: (1) incoming emails, (2) outgoing emails, and (3) notes added by the salesperson, invisible to the client. We hypothesized that tokens might carry different implications depending on the type of message in which they appear. Therefore, in order to represent this distinction in our model, we parsed email headers and used them to separate all the text associated with each opportunity into the appropriate categories. Then, for each training instance, we created three bag-of-words vectors, one for each of these categories, and concatenated these three vectors to create a single large bag-of-words vector. Thus, for this resulting feature vector, the word "sale" in an incoming email was represented as distinct from the word "sale" in an outgoing email or a note.

This segmentation-by-text-classification, however, led to significantly worse performance, decreasing our classifier's MCC from 0.482 to 0.300. We hypothesize that this indicates that words have similar implications regardless of whether they appear in an email, outgoing or incoming, or a note; thus, it is more useful to keep all text in a single bucket. It is interesting to note, however, that the top 20 features for this segmentation-by-text-classification model were all from the "note" category. Tokens from the two email categories were much less significant. This seems to be because salespeople are very honest and direct in their notes, since they are private, whereas their emails are shaped more by politeness, social norms, strategic tones, and so on. Therefore, tokens in notes are more indicative of the ultimate fate of an opportunity.

### 3.3.3 Metatextual Features

In addition to these bag-of-words features, we experimented with a variety of metatextual features, such as: number of outgoing emails; number of incoming emails; number of notes; ratio of outgoing to incoming emails; and median time interval between email exchanges. However, these features were not very effective. When used alone, without bag-of-words features, they led to a classifier with performance equivalent to random guessing (MCC=0); when appended to our bag-of-words features, they did not change performance relative to the baseline bag-of-words classifier. Thus, these metatextual features do not seem to be strongly correlated with wins or losses.

In an attempt to gain insight into types of text, we used k-means clustering to cluster pieces of text associated with each opportunity. Then, for each cluster, we used the number of texts in that cluster as a feature for each opportunity. Baker and McCallum suggest that clustering text can lead to useful semantic understanding and higher classification accuracy (1998). However, for our dataset, clusters were not well-correlated with wins or losses, and this addition did not improve our classifier's performance. As future work, we propose exploring using Latent Dirichlet Allocation to perform topic modeling for fragments of text; this may allow us to "group" fragments more effectively than k-means clustering by grouping around topics rather than raw distance between bag-of-words vectors.

## Conclusion

We find that running logistic regression on a simple bag-of-words feature vector performs well at classifying sales, approaching the performance of human classification. When using optimizations such as inclusion of bigrams and trigrams, our learning algorithm achieves precision and recall of 0.72 on our training set, for an MCC value of 0.482. As future work, we propose using topic modeling algorithms to group text more effectively, and combining textual

features with non-textual features, making use of all available data to create a more accurate classifier.

**Acknowledgements**

We are extremely grateful for our partnership with Roam Analytics, and we would like to extend a sincere thank you to Andrew Maas, Kevin Reschke, and Joe Barefoot for providing invaluable insights and advice along the way.

**References**

Baker, L. D., & McCallum, A. K. (1998, August). Distributional clustering of words for text classification. In Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval. ACM.

Matthews, D. R., Hosker, J. P., Rudenski, A. S., Naylor, B. A., Treacher, D. F., & Turner, R. C. (1985). Homeostasis model assessment: insulin resistance and β-cell function from fasting plasma glucose and insulin concentrations in man. *Diabetologia*, *28*(7), 412-419.

Mladenic, D. (1999). Text-learning and related intelligent agents: a survey. *Intelligent Systems and their Applications, IEEE*, *14*(4), 44-54.

Pang, B., Lee, L., & Vaithyanathan, S. (2002, July). Thumbs up?: sentiment classification using machine learning techniques. In Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10 (pp. 79-86). Association for Computational Linguistics.