WordSleuth: Deducing Social Connotations from Syntactic Clues

Shannon Stanton

Honors Thesis 2011

WordSleuth: Deducing Social Connotations from Syntactic Clues

Shannon Stanton
University of California, Irvine
Information and Computer Science
Campus-wide Honors Program
sstanton@uci.edu
shannonnstanton@gmail.com

0. Abstract

The realm of social and emotional connotation is often thought to be the purview of humans rather than machines. Namely, humans are generally capable of recognizing social connotations including emotions (such as embarrassment), intentions (deception and persuading), attitudes (confidence and disbelief), and tone (formality, politeness, rudeness), and recent work has suggested that machines may also be capable of this feat (Pearl and Steyvers 2010). This study extends the work done by Pearl and Steyvers, improving the data gathering methodology, feature extraction, and machine learning classification. Prior to the WordSleuth project, a major barrier to researching social cues transmitted through text has been a lack of annotated data. WordSleuth, an online Game-With-a-Purpose (von Ahn 2006), solves this problem, creating an effective means of encouraging a wide variety of participants to generate and annotate data. Salient linguistic features can then be extracted from the data gathered and used to train and test machine learning algorithms, effectively teaching machines to identify social connotations in text. In particular, as machines still currently lag behind human capabilities, this study extends Pearl and Steyvers' work by examining more complex linguistic features and exploring more sophisticated machine learning methods, with the aim of substantially improving machine recognition of social connotation.

1. Introduction

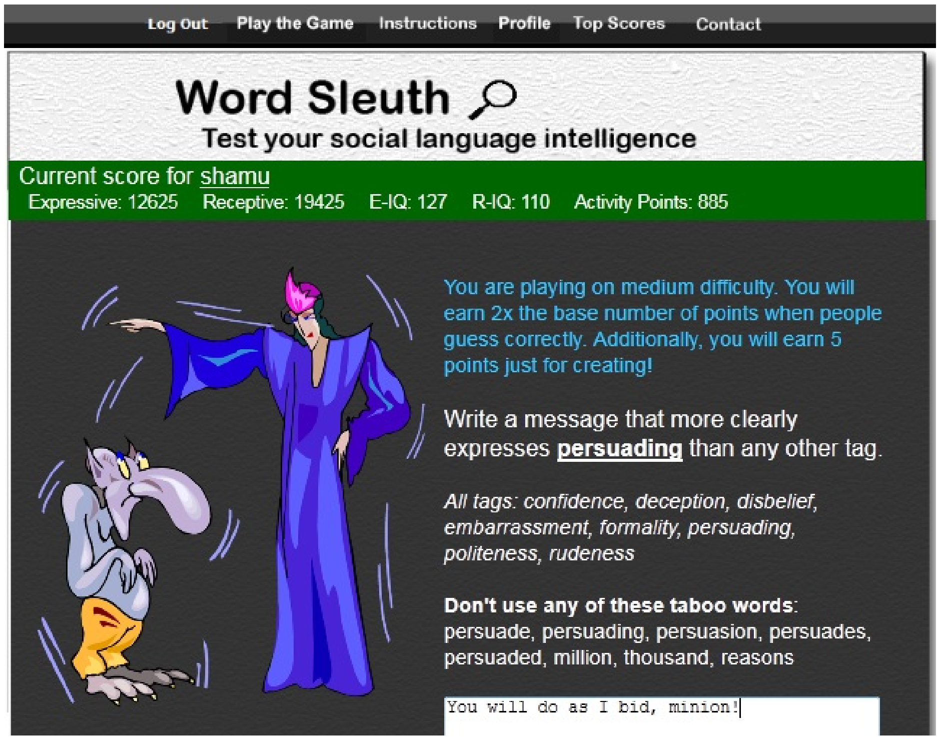
An important question in computational linguistics research is how non-linguistic information, such as emotions, intentions, attitudes, and tone, can be derived from language text. People are generally capable of it, but so far, machines have lagged significantly behind human capability. One approach is to identify possible features humans use, such as low level syntactic cues, and extract them from the input, allowing machine learning algorithms to make use of them, potentially even better than humans. This research project focuses on low level syntactic clues present in plain text.

The primary technical barrier to research in social connotation up until this project was a lack of socially annotated data. In order to extract such social information from text, we must first have a reference point constituted by sufficient examples of each category: a database of reliable messages reflecting human perceptions of both the intended and perceived social information. We therefore cannot simply automate the process (until after this project), since the machine learning itself requires training data to learn from. We need also a diversity of examples and styles to generalize from, so simply annotating existing works may be insufficient, and is, at the very least, extremely time-consuming. While some sources of information annotated for select specific categories exist, such as a database for deception created from the online game Mafia Wars (Zhou and Sung 2008), these sources do not reflect the breadth of social connotations we are looking for. Thus, the goal is to obtain messages generated and annotated by many people. Simple survey techniques can only bring in so much data due to limited scope and appeal. Pearl and Steyvers' (2010) solution to this problem: a game, specifically, a game-with-a-purpose (von Ahn 2006), that can automate the acquisition of data and increase the amount provided by volunteers by making participation more enjoyable (Pearl and Steyvers 2010). We call that game WordSleuth.

2. Creating WordSleuth

2.1 The function and purpose of the WordSleuth game

WordSleuth's game play is bimodal, facilitating the gathering of both new annotated messages and annotations of old messages. In the first mode, message generation, players are presented with a contextual picture for inspiration and one of eight social cues, and asked to create a message that expresses that cue more than any of the others, without using particular taboo words that might make the task of identification too easy. This mode enables the generation of new annotated data, but that alone would be insufficient: we also want to gather data about people's perceptions of the message's social category.



Current score for shamu
Expressive: 12625 Receptive: 19425 E-IQ: 127 R-IQ: 110 Activity Points: 885

You are playing on medium difficulty. You will earn 2x the base number of points when people guess correctly. Additionally, you will earn 5 points just for creating!

Write a message that more clearly expresses **persuading** than any other tag.

All tags: confidence, deception, disbelief, embarrassment, formality, persuading, politeness, rudeness

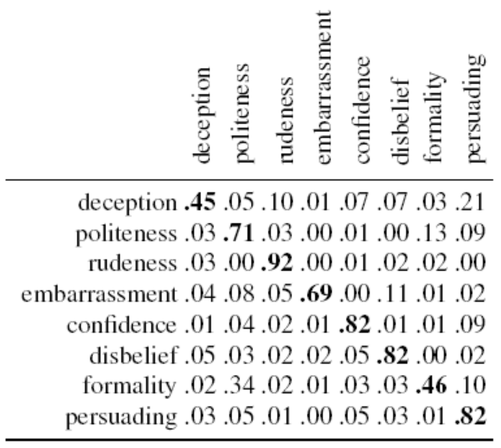
Don't use any of these taboo words: persuade, persuading, persuasion, persuades, persuaded, million, thousand, reasons

You will do as I bid, minion!

Illustration 0: *Message Creation Mode*

In the second mode, cue identification, players are presented with a message and the contextual image used to generate it, and asked to identify which of the eight social cues the message best communicates. This mode allows users to peer review each others' submissions, providing information about whether messages identified represent "good" examples of their social cue. Ideally, messages that are the best examples of their category are always agreed upon, while the worst examples show a high degree of confusion among the guessers. It also increases the appeal of the game play, as it appears players have a strong preference to the relatively simpler task of identification, perhaps because it is faster and less cognitively taxing, providing more instant gratification.

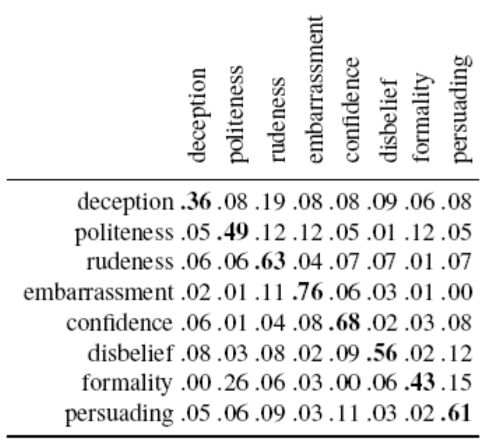
It has been shown that this type of communal effort of non-experts is capable of producing data as reliable as that generated by few experts (von Ahn 2006). For convenience and ease of comparison, the following tables show the initial results obtained by Pearl and Steyvers' participants, when the database included 1176 messages and 3198 annotations. The reliability of the data increased dramatically when we considered messages that have been agreed upon for at least 50% of at least two annotations (Pearl and Steyvers 2010).



	deception	politeness	rudeness	embarrassment	confidence	disbelief	formality	persuading
deception	**.45**	.05	.10	.01	.07	.07	.03	.21
politeness	.03	**.71**	.03	.00	.01	.00	.13	.09
rudeness	.03	.00	**.92**	.00	.01	.02	.02	.00
embarrassment	.04	.08	.05	**.69**	.00	.11	.01	.02
confidence	.01	.04	.02	.01	**.82**	.01	.01	.09
disbelief	.05	.03	.02	.02	.05	**.82**	.00	.02
formality	.02	.34	.02	.01	.03	.03	**.46**	.10
persuading	.03	.05	.01	.00	.05	.03	.01	**.82**

Table 3: Confusion matrix for the human participants, where the majority of participants agreed on a message's intended social information and at least two participants labeled the message. The rows represent the intended social information for a message while the columns represent the labeled social information, averaged over messages and participants.

(Pearl and Steyvers 2010)



	deception	politeness	rudeness	embarrassment	confidence	disbelief	formality	persuading
deception	**.36**	.08	.19	.08	.08	.09	.06	.08
politeness	.05	**.49**	.12	.12	.05	.01	.12	.05
rudeness	.06	.06	**.63**	.04	.07	.07	.01	.07
embarrassment	.02	.01	.11	**.76**	.06	.03	.01	.00
confidence	.06	.01	.04	.08	**.68**	.02	.03	.08
disbelief	.08	.03	.08	.02	.09	**.56**	.02	.12
formality	.00	.26	.06	.03	.00	.06	**.43**	.15
persuading	.05	.06	.09	.03	.11	.03	.02	**.61**

Table 4: Confusion matrix for the machine learning classifier. The rows represent the intended social information for a message while the columns represent the labeled social information.

(Pearl and Steyvers 2010)

WordSleuth was originally created as an offline game, which limited its effectiveness in reaching participants and gathering data. A much larger database is required to truly generalize about such a nebulous subject as social connotations.

2.2 Bringing WordSleuth online

A solution to the data deficit problem is putting the game online (see http://gwap.ss.uci.edu/ for the current instantiation), increasing its accessibility to the general public and increasing the amount of data generated. HTML templates were used for the webpages forming the front end of the system, driven by Perl CGI scripts. More modern, flashy methods such as Ruby-on-Rails were contemplated and discarded in favor of quick prototyping and known compatibility with popular browsers. Finally, the front-end system was integrated with a mySQL database, an improvement in efficiency, availability, and methodology from the text files previously used.

Results to date are promising. Since bringing the game online in January 2011, the number of annotations has increased dramatically while the number of messages created has nearly doubled. As of May 2011 the database contains just over 3,500 messages and 20,000 annotations.

	confidence	**deception**	**disbelief**	**embarrassment**	**formality**	**persuading**	**politeness**	**rudeness**
confidence	**.81**	.03	.02	.01	.01	.07	.03	.02
deception	.08	**.60**	.04	.03	.02	.13	.05	.05
disbelief	.03	.03	**.79**	.03	.01	.02	.03	.04
embarrassment	.01	.03	.07	**.78**	.02	.01	.05	.02
formality	.04	.02	.02	.02	**.46**	.09	.34	.02
persuading	.08	.05	.01	.00	.02	**.77**	.04	.02
politeness	.02	.02	.01	.02	.13	.07	**.72**	.02
rudeness	.02	.01	.04	.02	.01	.04	.01	**.85**

Table 0: *Human annotations for database (as of May 2011)*
Mean accuracy: 0.74

	confidence	deception	disbelief	embarrassment	formality	persuading	politeness	rudeness
confidence	**.87**	.01	.01	.00	.01	.06	.02	.01
deception	.05	**.76**	.02	.02	.01	.09	.03	.02
disbelief	.02	.02	**.86**	.03	.01	.01	.03	.03
embarrassment	.00	.03	.05	**.86**	.02	.01	.03	.01
formality	.02	.00	.00	.01	**.68**	.04	.24	.01
persuading	.05	.04	.01	.00	.01	**.84**	.03	.01
politeness	.02	.02	.00	.01	.10	.04	**.80**	.01
rudeness	.01	.01	.03	.02	.01	.03	.01	**.88**

Table 1: *Human annotations for reliable messages (as of May 2011)*
Mean accuracy: 0.84

In general, the expansion of the database has seen an increase in user accuracy in identifying the intended social cue, as well as the reduction of certain ambiguities. Confusion of deception for confidence, for example, has been halved, even without filtering for reliably annotated messages. Rudeness is still easiest for users to identify, but by a slimmer margin. However, some sources of confusion remain prominent, for example formality for politeness, and less so, the reverse.

2.3 Improvement Feature: Taboo word list

One potential complication that may arise with gathering data in a competitive framework is the possibility of amassing messages that are artificially representative of their classifications. Players motivated by point gain may specifically craft messages that are trivial to guess by including the social tag in the message or using words that are too closely related to the tag. For example, the task of identifying "politeness" in a message is trivialized if every message assigned to that category has the word "please". Therefore, users should be prevented from using select words. Rejecting messages containing variations of the tag and the tag itself was a simple starting point and solved the first half of the problem, but we also needed some way of tracking words that were becoming over represented in the database. Our solution was to dynamically generate a list of taboo words based on the theory of mutual information.

Mutual information is a measure of the inter-dependence of two variables (Peng 2005): in this case, word frequency and social category. Two independent variables should have a mutual information score of 0, while two variables that are dependent and closely related will have a higher score than two non-closely related. The following equation was used,

$$Mutual\ Information(x, y) = \log \frac{(p(x:y))}{(p(x)*p(y))}$$

where,

$p(x:y) = probability\ of\ word\ x\ given\ category\ y$,
$p(x) = probability\ of\ word\ x\ among\ all\ words$,
$p(y) = probability\ of\ category\ y\ among\ all\ categories$.

For each social category, the words with the highest mutual information score are declared to be taboo in the game, and players are not allowed to use them when generating a message for that particular category. Common words, such as articles and pronouns, should be automatically excluded, since they are evenly distributed among all the categories.

Taboo list functionality was implemented with a Perl script to calculate the mutual information scores for each word in each social category in the current database, set to update approximately once per day. Thus the taboo lists are dynamically updated to reflect the state of the database, automatically without requiring the direct supervision of the researchers. The following code fragment illustrates the implementation of the mutual information calculation:

```
# calculate p(x) = #occurrences of word/#total words
my $px = $wordFrequency{$word}/$totalWords;
# calculate p(y) = #occurrences of a given tag/totalMessages
my $py = $tagCount{$category}/$totalMessages;
# calculate p(x|y) = #word x in tag y/#words in tag y
if (! exists $wordCount{$category} || $px == 0) #if $py is 0,
      bigger problems to be alerted to (ie. social tag not
      existing)
{
      $pointWiseMutualInfo = 0;
}
else
{
      my $pxGy = $count/$wordCount{$category};
      $pointWiseMutualInfo = log($pxGy/$px/$py); #log(p(x|y)/
            (p(x)*p(y)))
}
$mutualInfo{$category}{$word} = $pointWiseMutualInfo;
```

Code Fragment 0: *tabooListGenerator.pl: calculating mutual information*

For example, as of May 2011 each category yielded the following taboo words:

Category	Taboo List: Top 7
confidence	wil, modest, mvp, talkies, rule, scruffles, sorts
deception	recommend, spreadsheet, dastardly, issue, nerdy, jan, suntan

disbelief	beats, megaphone, guitar, twenty, vat, goatse, smoothly
embarrassment	stew, conscious, mins, grease, mighty, private, spade
formality	delivery, abuse, form, grammy, greetings, martin, distinguished
persuading	million, thousand, reasons, captain, poverty, carrots, tonic
politeness	nicely, grateful, bumping, rough, shore, orphans, scores
rudeness	kangaroo, facts, uncalled, scum, listed, spotty, gingers

Table 2: *Taboo list results (as of May 2011)*

Many of these words are intuitively related to their given category: "modest" in confidence, "recommend" in deception, "million", "thousand", "reasons" for persuading, etc. However, many appear at first glance to be out of place.

A useful, if unexpected, outcome of applying this methodology was the identification of words that were non-intuitively highly correlated with particular categories. For example, just after the game went online in January 2011, the taboo list generator yielded "nancy" for confidence. Yet "nancy" does not seem to be a word that one would intuitively associate with the social category confidence; it seems rather arbitrary. In fact, that unigram was an artifact of the message generation system. In the beginning, when the game was offline and the database relatively small, a user happened to use the name "Nancy" in several messages for the category confidence. Because there were so few repeated words in general and that one happened to be used enough in a particular category, it had a relatively high mutual information score, even though it may not be truly representative of the category. Making "nancy" taboo for the confidence category prevents users from creating additional instances correlating the unigram to the category, thus eventually lowering its mutual information score. Thus, taboo functionality reduces the effect of coincidental correlation.

Eventually, as the database grows, trends can be examined to set an appropriate absolute boundary on the mutual information score, rather than simply using the highest relative scores. The taboo list should eventually resemble the game for which it was named and represent words that are highly correlated for each category within the current database. It is important to note that this will not necessarily reflect the correlation present in general language usage, since this model actively discourages high correlations. Therefore, taboo list functionality increases both the depth and breadth of data represented by discouraging trivially obvious words such as the categories themselves and by dynamically identifying and reducing coincidentally high correlations of words to categories.

3. Using WordSleuth

The data gathered in the WordSleuth database cannot be simply directly fed to a computer and expect coherent results. It must first be parsed and processed for salient, numerable features. Furthermore, many feature are only present for a few messages, listing only those features present for each message reduces the dimensionality of the data set, thus increasing the efficiency of the algorithms.

3.1 Features

Originally, Pearl and Steyvers used 12 features extracted for each message: the number of word types, number of word tokens, ratio of types to tokens, number of punctuation marks, number of question marks, number of exclamation marks, number of main clauses, average characters per word, mean log frequency of words used, and lists of unigrams, bigrams, and trigrams that appear more than once in the data set. This project added the following features: number of interrobangs, ratio of exclamation to question mark, average words per main clause, number of sub-clauses, average words per sub-clause, and accuracy and precision scores for human performance on each message. For example, interrobangs appear in the disbelief category more often than others, while formality and deception are often expressed with numerous sub-clauses distancing the speaker from the audience. Accuracy and precision scores give a sense of the usefulness of a particular message as an exemplar. Accuracy is calculated as the percentage of times a particular message was correctly identified, while precision represents a measure of the agreement (or lack of confusion) of the guessers, calculated as a percentage of the maximum possible entropy. Maximum entropy (which is 3 bits for an 8 category choice) represents the state of maximum confusion (each category is guessed 1/8 of the time), and thus the lowest precision (0). Minimum entropy (0) represents complete certainty (such as when all guessers guess the same category) and thus the highest degree of precision (1.0 or 100%). Thus precision is calculated:

$$precision=\frac{(H_{max}-H_x)}{H_{max}}$$

where,

$$H_x=\sum p(x)*\log_2\frac{1}{(p(x))}$$

and,

$$H_{max}=H(\frac{1}{8},\frac{1}{8},\frac{1}{8},\frac{1}{8},\frac{1}{8},\frac{1}{8},\frac{1}{8},\frac{1}{8})=3$$

I considered several ways to calculate precision, such that precision should represent the amount of agreement of the guessers on a particular message.

First, I considered precision to be simply the frequency of the most common guess, but quickly realized some flaws with this hypothesis. This calculated precision could never be lower than accuracy, and yet it occurs in other domains that precision is lower than accuracy. Further, this metric would not be sufficiently fine-grained. For example, consider 2 messages, one that is guessed 50% one category and 50% another, to be represented (.5, .5) for short, and the other, that is guessed 50% one category, 25% another, and 25% a third (.5, .25, .25). In both cases, this calculation for precision would yield .5, but it seems intuitively that the second case represents a higher degree of confusion among the participants, since more categories were under consideration.

Next, I considered various ways of penalizing precision based on the number of categories guessed. However, this method is insufficiently fine-grained as well. Consider 2 messages, the first (.5, .25, .25) and the second (.5, .24, .01). Simply accounting for the most commonly guessed and the number of categories would calculate the same precision for each of these messages, but again intuition says the second one might represent a lower degree of confusion, since the third category has so few guessers compared to the other two. Precision should take into account the relative frequency of each category guessed as well.

The entropy ratio calculation solves these problems. It is possible for a message to have lower precision than accuracy (such as, for example, (.3, .1, .1, .1, .1, .1, .1, .1)), and there is sufficiently high granularity to distinguish the aforementioned cases.

3.2 Algorithms

Preliminary research with the machine learning algorithm Sparse Multinomial Logistic Regression (Pearl and Steyvers 2010) showed performance nearly on par with human performance, but not quite. Just as there is variation among the performance of individual humans on learning tasks, different machine learning algorithms vary in performance, with their own sets of strengths and weaknesses. This paper examines additional algorithms in an attempt to reach human proficiency.

3.2.1 KNN: K-Nearest-Neighbors

3.2.1.1 KNN Background

As a "peer pressure" multinomial classification algorithm, K-Nearest-Neighbors operates on an inductive principal of classifying a test data point based on the training data points proximate to it. Each unknown data point adopts the classification of those closest to it, or, in the case of disagreement, the most common classification of nearby training points. Let there be two subsets of data, one for training whose classifications are known to the algorithm and one for testing whose classifications are unknown to the algorithm, but known to the evaluator of algorithms. (Here the "correct classification" is defined as that specified by the user when the message was generated.) For each data point in the test data, KNN calculates the Euclidean distance between that data point and each data point in the training subset. It then assigns the classification of the test data point to the most common classification of the K training cases with the smallest distances.

There is some concern about efficiency. For n test cases and d training cases, the algorithm runs in at minimum O(n*d) time and can do no better, making it inefficient for large values of n or d. In reality, because of the way we parse features, n depends on both the number of messages and the number of features parsed, and thus grows rather quickly. KNN may not be practical if the database continues to grow in size as hoped.

To begin with, KNN was run on the database toward the end of May 2011 and fed only the features originally extracted by Pearl and Steyvers in 2010. Next, KNN was applied to the additional low level features. In both cases, performance was averaged over values of N ranging from 1 to 55.

3.2.1.2 KNN Results

	confidence	**deception**	**disbelief**	**embarrassment**	**formality**	**persuading**	**politeness**	**rudeness**
confidence	**.80**	.04	.01	.02	.02	.06	.02	.03
deception	.09	**.55**	.03	.01	.00	.11	.01	.00
disbelief	.03	.02	**.79**	.02	.03	.04	.04	.03
embarrassment	.02	.06	.03	**.79**	.02	.02	.03	.02
formality	.02	.01	.01	.02	**.60**	.03	.31	.00
persuading	.07	.06	.02	.01	.03	**.76**	.03	.02
politeness	.03	.02	.03	.11	.02	.05	**.71**	.02
rudeness	.02	.01	.07	.03	.00	.06	.01	**.80**

Table 3: *KNN on May 2011 data, original features*
Mean accuracy 0.76

	confidence	**deception**	**disbelief**	**embarrassment**	**formality**	**persuading**	**politeness**	**rudeness**
confidence	**.17**	.14	.12	.10	.08	.12	.16	.10
deception	.13	**.13**	.16	.09	.12	.16	.09	.12
disbelief	.12	.10	**.13**	.12	.14	.16	.12	.10
embarrassment	.06	.16	.11	**.11**	.16	.14	.15	.11
formality	.10	.15	.14	.18	**.10**	.10	.15	.07
persuading	.13	.13	.16	.11	.13	**.08**	.15	.12
politeness	.15	.08	.16	.08	.15	.16	**.18**	.13
rudeness	.12	.09	.12	.11	.11	.09	.15	**.20**

Table 4: *KNN on May 2011 data, all features*
Mean accuracy 0.24

Notably, KNN's mean performance on the original features is equivalent to human performance on all messages. Surprisingly, KNN performed much worse with all features than with the original features alone. However, KNN is sensitive to dimensionality and proximity, and it may be that the new features confused the algorithm by creating the illusion of proximity.

KNN is a naïve algorithm in that it overlooks certain patterns apparent in the data, such as clustering. Furthermore, as an inductive algorithm, it is only able to learn from the training set. Thus, it is unable to make use of test cases themselves, which would be particularly beneficial when the differing categories are highly interspersed, as is the case here. Transductive clustering suffers neither of these deficiencies.

3.2.2 Transductive Clustering

The primary difference between induction and transduction in this case is the ability to make use of information from unlabeled points in the test subset (Chapelle, Scholkopf, and Zien 2006). While inductive KNN would only use training data near a test point, transduction also considers other as yet unlabeled test points and is able to make use of their proximity once labeled. Furthermore, clustering is able to take advantage of the patterns that exist in the data beyond the first level of nearby points.

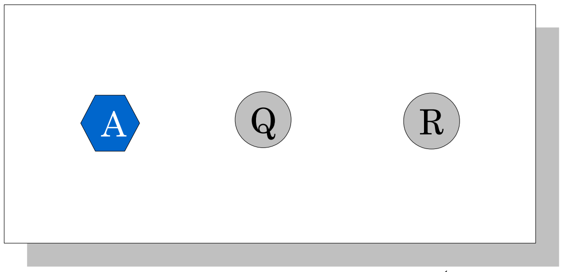


Illustration 1: *Training point A (gold hexagon), test points Q and R*

For example, consider Illustration 1: if training point A is near test point Q, and Q is near test point R, transductive clustering is able to infer that A and Q and R should have the same label, since they form a cluster, because the unlabeled test point Q between A and R joins them together. Both KNN and transductive clustering would label both Q and R with category gold hexagon, but with differing underlying logic. Inductive KNN would label Q according to A (gold hexagon), and then R according to A (also gold hexagon), and not explicitly understand that Q and R are the same category, because it is blind to point R when considering point Q (and vice-versa). This difference in logic becomes more salient if there is an additional training point of a different category, as follows.

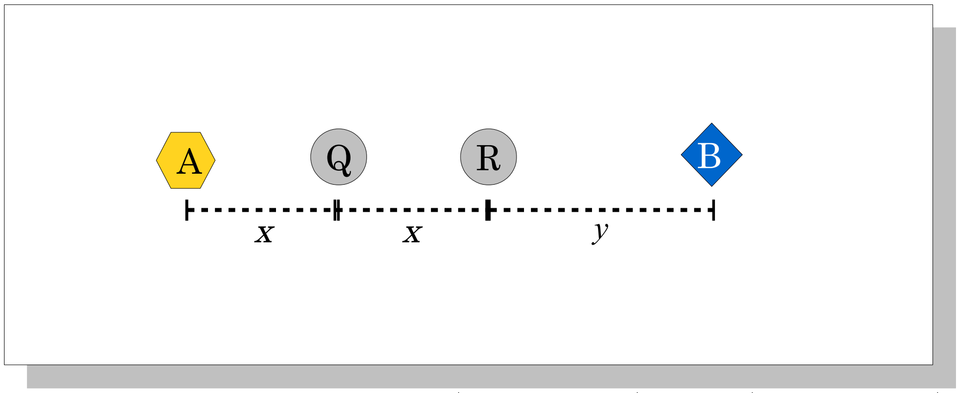


***Illustration 2**: Training points A (gold hexagon) and B (blue diamond), and test points Q and R. Distances* x *and* y *such that* x < y < 2x.

Now consider Illustration 2, in which another training point B exists (labeled with category blue diamond), closer to R than A is to R, and of a different label than A. KNN (K=1) would label R according to B, rather than according to A, since R is nearer to B, though intuitively A and R should probably belong in the same cluster, and thus the same category label. This intuition grows stronger with the introduction of more unlabeled points, as shown in Illustration 3.

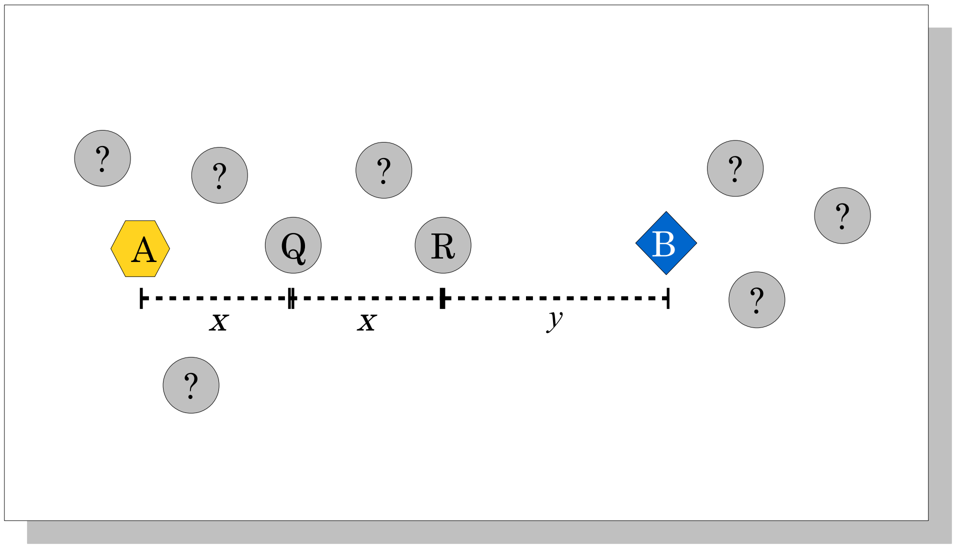


***Illustration 3**: Additional unlabeled data points enhance the intuition of two clusters, where the left cluster should be gold hexagon, and the right cluster blue diamond.*

3.2.2.1 Transductive Agglomerative Clustering

Transductive Agglomerative Clustering works by merging nearby points into clusters (Gashler 2011). Once a labeled point is merged into a cluster, the entire cluster gains the label of that point, and thus do all the unlabeled points within the cluster. In theory this sounds plausible. However, the mean accuracy of this algorithm was only about 0.13 (below the baseline of 0.15), when tested with 10 repetitions of 10-fold cross-validation. Upon closer examination of the algorithm, one finds that clusters of differing labels are never joined, which bodes ill for data that shows many small, interspersed clusters, or clusters that have some conflicting labels. These are in fact the characteristics inherent to the current WordSleuth data set.

3.2.2.2 Transductive Graph Cutting

Transductive Graph Cutting uses a min-cut/max-flow algorithm to separate out the various labels present in the data and deliminate clusters accordingly (Gashler 2011). When run on the May 2011 data set with only the original features present with both 10 repetitions of 10-fold cross-validation and 10 repetitions of 2-fold cross-validation, the mean accuracy was 0.97, much higher than the other algorithms or human annotations. A result so high seemed to indicate the potential of overfitting; to truly determine, additional testing data is required, but running 2 fold cross-validation to reduce the ratio of training to test data suggests the results are robust. When run on the same data set and cross-validation, but with all features extracted, the mean accuracy was 0.98, showing that the additional features did not cause this algorithm the level of confusion as inductive KNN experienced.

4. Future Directions

With additional time, the WordSleuth project could benefit from further research done in several areas, including additional feature research and machine learning techniques. For example, this paper only examines relatively low level syntactic clues; the success of certain classifiers relative to humans on such low level cues suggests that humans may cue into these low level clues, but they probably also use higher level data, including sentence structure. Input messages could be parsed into syntax trees to examine high level syntactic structures. I began tentative work on approximating these structures with simple parts of speech tagging which shows promise, but time constraints did not permit. Additional machine learning algorithms not examined in this paper, including additional inductive and transductive algorithms would be interesting to look into, and combining the strengths of multiple algorithms with methods such as bagging could yield more powerful, consistent, and robust results.

5. Works Cited

Gashler, Mike. *Waffles*. http://waffles.sourceforge.net/docs.html. March 2011.

Pearl, L. & Steyvers, M. (2010). Identifying Emotions, Intentions, & Attitudes in Text Using a Game with a Purpose. Proceedings of NAACL-HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text. Los Angeles, CA: NAACL.

Peng, H.C., Long, F., and Ding, C., "Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No. 8, pp. 1226-1238, 2005.

von Ahn, L. 2006. Games With A Purpose. IEEE Computer Magazine, June 2006: 96-98.

Zhou, L., Burgoon, J., Nunamaker, J., and Twitchell, D. 2004. Automating linguistics based cues for detecting deception in text-based asynchronous computer mediated communication. Group Decision and Negotiation, 13: 81-106.

Zhou, L. and Sung, Y. 2008. Cues to deception in online Chinese groups. Proceedings of the 41st Annual Hawaii international Conference on System Sciences, 146. Washington, DC: IEEE Computer Society.

6. Appendix Contents: code written specifically for WordSleuth

6.1 Taboo list generation script: taboo_list_generator.pl

6.2 Feature extraction script: get_features_shamu.pl