A Tangled Web: The Faint Signals of Deception in Text Boulder Lies and Truth Corpus (BLT-C)

Franco Salvetti[†], John B. Lowe[‡], and James H. Martin[♯]
Department of Computer Science, University of Colorado at Boulder, 1111 Engineering Drive, Boulder, CO USA[†,♯]
Department of Linguistics, University of California at Berkeley, 1203 Dwinelle Hall, Berkeley, CA USA[‡]
franco.salvetti@colorado.edu[†], jblowe@berkeley.edu[‡], martin@colorado.edu[♯]

Abstract

We present an approach to creating corpora for use in detecting deception in text, including a discussion of the challenges peculiar to this task. Our approach is based on soliciting several types of reviews from writers and was implemented using Amazon Mechanical Turk. We describe the multi-dimensional corpus of reviews built using this approach, available free of charge from LDC as the Boulder Lies and Truth Corpus (BLT-C). Challenges for both corpus creation and the deception detection include the fact that human performance on the task is typically at chance, that the signal is faint, that paid writers such as turkers are sometimes deceptive, and that deception is a complex human behavior; manifestations of deception depend on details of domain, intrinsic properties of the deceiver (such as education, linguistic competence, and the nature of the intention), and specifics of the deceptive act (e.g., lying vs. fabricating.) To overcome the inherent lack of ground truth, we have developed a set of semi-automatic techniques to ensure corpus validity. We present some preliminary results on the task of deception detection which suggest that the BLT-C is an improvement in the quality of resources available for this task.

Keywords: boulder lies and truth corpus, deception detection, corpus construction

1. Introduction

Oh! what a tangled web we weave
When first we practise to deceive![1]

Deception, as a conscious and deliberate act, is a socially pervasive psycholinguistic phenomenon—from lies on the witness stand during legal trials to fabricated online product reviews. Its detection in human communication has long been of great interest in real-life situations involving law enforcement (Granhag and Strömwall, 2004), national security (National Research Council, 2003), and business (Xiao and Benbasat, 2011)—just to mention a few.

The techniques employed for the detection of deception are varied, ingenious, and often dramatic—from the ancient Chinese method of spitting dry rice to the modern polygraph.

Deception detection has been the subject of investigation within psychology, social science, and linguistics, where it has mainly been based on qualitative and quantitative observations of gesture, facial expression and voice analysis. Nonetheless, there is only a little scientific work, most of it quite recent, that has been done on the theoretical underpinnings of systems for automatically detecting deception in *written text*.

One of the principal challenges to making progress on computational methods for detecting deception is a peculiar characteristic of the task: human beings are not very good at it. In fact, human performance in detecting deception is no better than chance (Bond and DePaulo, 2006) and sometimes even below chance, due to (among other reasons) *truth bias* (Vrij, 2008). For this reason it is virtually impossible to build a corpus in the traditional way: judges annotating data, following guidelines, which in turn are leveraged to build machine-learned classifiers or studying deception invariants.

Furthermore, deceptiveness is an *emergent property of* an internal mental state. Ergo deceptive statements may be perfectly true, unbeknownst to the deceiver[2]. And the truth-polarity of most, if not all, deceptive statements can be reversed by simply changing the real-world entities referred to or by selecting a different verb. Such changes would not alter any invariants of deceptiveness that might be present. Conceptually, then, deceptiveness is strictly speaking independent of the truth values of the content.

In the section on related work below, we argue that some prior work demonstrating the ability to detect deception in text may be flawed due to uncontrolled confounders in the construction of the experimental corpora. In this paper we will present the details about the creation of the Boulder Lies and Truth Corpus (BLT-C) along with some preliminary results based on a study leveraging such a corpus.

1.1. Deception vs. Lying

We employ the following definition of deception in human communication[3]:

> to deceive $=_{df}$ *to intentionally cause another person to have a false belief that is truly believed to be false by the person intentionally causing the false belief.*

Additionally, we distinguish between *deception* and *lying*:

> to lie $=_{df}$ *to make a believed-false statement to another person with the intention that that other person believe that statement to be true.*

[1] (Scott, 1806), Canto vi. Stanza 17.

[2] Though of course in the canonical case, deceivers intend for a falsehood to be accepted as truth.

[3] http://plato.stanford.edu/entries/lying-definition

A *lie* is a form of *deception* and there are forms of deception (e.g., *omission*) which are not lies. The *intentionality* of the deceptive act may cause the speaker to leave traces (i.e., signals) in the communication that can be leveraged by a system to automatically detect deception. (Qin and Burgoon, 2005) provide a list of *types* or *dimensions* of deception: *lies, fabrications, concealments, omissions, misdirection, bluffs, fakery, mimicry, tall tales, white lies, deflections, evasions, equivocation, exaggeration, camouflage, strategic ambiguity, hoaxes, charades,* and *impostors*. Unfortunately, much of the literature on deception detection equates the concept of deception to lying. Although strictly speaking this equation is inaccurate, we, too, will also occasionally collapse this distinction. We further make a *significant* distinction between two kinds of deception: deception regarding known objects and deception regarding unknown ones. We will use the term *lie* to refer to deception regarding known objects and the terms *fabrication* or *fake* to refer to deception regarding unknown objects.

We hypothesize that during a deceptive act there is unwanted and unintended *leakage* of deceptive signal into the media supporting the communication (e.g., text). We further hypothesize that such a signal can be turned into features which can be leveraged to detect deception. In support of this hypothesis we offer evidence from the psychoanalytic literature (Freud, 1901): the underlying state of mind of a speaker may distort the speaker's utterances, unintentionally exposing that state of mind.

1.2. Deception Detection

The goal of deception detection is to build either an automatic system or provide a coherent body of heuristics which, based on observable features of a human communication, will be able to determine whether a piece of communication is deceptive. It is important to point out that unlike other Artificial Intelligence tasks, this is a task for which it is known in literature (Bond and DePaulo, 2006) that humans—even trained law enforcement agents—perform at chance or even below chance. Therefore, a system whose quality is comparable to human performance on this task is useful neither for practical purposes nor for investigating the phenomenon.

2. Related Work on Corpus Creation

Previous corpora have been created and leveraged for the task of deception detection by exploiting a wide range of linguistic features, including lexical selection, morphological and syntactic patterns and hand-crafted lexical resources. We cite briefly some relevant work in creating such corpora.

Lying Words: (Newman et al., 2003) created a labeled corpus of elicited narratives marked as `lie` or `true`, then applied machine learning techniques (logistic regression) to rank the contribution of these linguistic categories.

Critical Segments: (Enos et al., 2007), employed CSC (Columbia SRI Colorado) Deception Corpus and hypothesized that there exists a class of speech segments, *critical segments*, whose truth or falsity can be used to compute the overall truth or falsity of the entire communication.

Deceptive Indicators: (Bachenko et al., 2008) describe a system built using NLP techniques and linguistic features as *deceptive indicators* for classifying truth-verifiable statements in civil and criminal transcripts as `lie` or `true`. The labelled corpus used in these experiments are real-world transcripts from actual criminal statements, police interrogations, and legal testimony.

The Lie Detector: (Mihalcea and Strapparava, 2009) used Amazon Mechanical Turk (AMT) as a source of annotations to build an annotated data set of `true` and `lie` texts on three topics (*abortion*, *death penalty*, and *best friend*). This was further extended in (Perez-Rosas et al., 2014) to include thermal, and visual responses of human subjects under three deceptive scenarios

Differences in Deception: (Conroy et al., 2015) present three types of fake news and a set of requirements for building a fake news detection corpus.

Deceptive Opinion Spam: (Ott et al., 2011) describe an experiment in automatic detection of *deceptive opinion spam*. The authors built positive and negative sets of opinion spam, using on-line hotel reviews and AMT. Following on this work, (Li et al., 2014) addressed issues arising from the use of AMT by broadening the set of review writers to include domain experts and by adding new domains.

Boulder Lies and Truth Corpus: (Salvetti, 2012), presented here, emphasized the potential for linguistic and extra-linguistic features to bias the results of machine-learned classifiers, describes the process of building a balanced corpus of deceptive and non-deceptive texts that controls for possible confounders. A particular focus of this work is to avoid confounding the truth status of the reviews with the source of the reviews (i.e, truthful reviews from actual customers and deceptive reviews from turkers).

This paper and the work reviewed in this section are mainly about deception detection using text-derived cues. For a more general overview of deception, see (Ekman, 2001) and (Vrij, 2008). In a post on Language Log[4], the reader will find a collection of references for deception detection based on speech cues.

2.1. Discussion on Corpus Creation

Previous studies show that it is possible, employing NLP techniques, to detect deception using *linguistic* cues. They also confirm earlier evidence that human performance is poor on this task and that automatic systems can significantly outperform humans. Such studies support the long-held intuition that there is enough signal in human communication to detect deception. These encouraging results, the many practical applications (e.g., deceptive spam filtering), and the fact that human performance is worse than machines, all suggest that pursuing statistical quantitative modeling of deception is both valuable and viable.

The chief difficulty in building a deceptive corpus is that *data cannot be tagged using human annotators*, as is standard for other NLP tasks. Indeed, because human performance in detecting truthful and deceptive narrative is so poor, explicit labels must be provided directly or indirectly by the speaker providing the data. And this in turn cre-

[4] http://languagelog.ldc.upenn.edu/nll/?p=3554

ates another confound—deception is a mental process impossible for other humans to directly observe so that corpus creators are forced to rely on judgments of cooperative subjects.

As a result of the difficulty in obtaining labeled data, the data used in previous studies is susceptible to bias arising from various sources. The largest difficulty with analyzing results from earlier corpora involves the use of real vs. elicited deceptive narratives. If we compare the *mock crime* interrogations in (Newman et al., 2003) with the real interrogations in (Bachenko et al., 2008), it can be easily imagined that the cognitive load of a real interrogation, compared with a *fake* one, must have consequences on the way in which deception is manifested. The small size of the many corpora is also problematic. For example (Bachenko et al., 2008) narrow down their investigation and their conclusions to a total of 275 propositions—this is unlikely a representative set for studying deception.

Despite these challenges, there are some promising results regarding the value of *lexical features* similar to those identified in (Newman et al., 2003), (Enos et al., 2007), (Bachenko et al., 2008), (Ott et al., 2011) and (Li et al., 2014). Modeling deception across more linguistic and statistical dimensions, identifying commonalities and differences in approaches and the impact these have on the results, more careful experimental setup and interpretation of results, and working with an extensive, shared corpus are steps that would certainly help advance this new subfield of study—deception detection in text.

3. The Boulder Lies and Truths Corpus (BLT-C)

To study deception and its invariants in text, we need a corpus consisting of deceptive and non-deceptive documents. Such a corpus would allow researchers and practitioners to systematically validate their hypotheses against annotated data and, more specifically, to employ statistical methods to identify deception invariants—general linguistic phenomena that are typical of deception and at the same time invariant across text dimensions (e.g., sentiment) and other contextual factors (e.g., emotional involvement of the speaker). The primary goal of the work described here is to build and *validate* a text corpus to be used to study deception. The corpus we present here is not specifically intended to represent *opinion spam* generally; specifically, we focus on deceptive and non-deceptive reviews.

To overcome some of the limitations of currently available corpora and to address some of the concerns presented in the discussion in section 2.1., we include additional dimensions (e.g., *sentiment* and *domain*) and extend the deception labels to include not just *fakes* but also *lies*, where *fakes* (or *fabrications*) are deception regarding objects unknown to the writer and *lies* are deception regarding known objects.

3.1. BLT-C Dimensions

Humans perform at chance or even below chance when trying to detect deception, so the conventional approach to corpus development for most NLP tasks—collecting text and labeling it using annotators following strict guidelines constrained by high inter-annotator agreement—is simply not feasible. Therefore, in the BLT-C, labels (e.g., `T` for truthful) were assigned directly by the creators of the text as they created it. Because of the practical applications related to detecting deceptive online reviews, we have chosen to focus our attention on online reviews of products or services as our primary genre. The BLT-C corpus is built by eliciting reviews of various sorts using Amazon Mechanical Turk (AMT), now a typical approach to building text corpora which allowed us to have a corpus representing more than 500 different authors—a number which would be almost impossible to reach with traditional annotation methods. And it differs from typical corpus building efforts significantly in the instructions: some of those recruited were instructed to deliberately prevaricate.

Turkers[5] are known to be prone to cheating when they know that no ground truth exists for a given task. For instance, if we simply ask turkers "do you like $(object)$?", they may feel that a randomly generated answer would not be caught as cheating, which would lead to lower data quality. To avoid this, we employ several strategies in the task guidelines, including making the turkers believe that a ground truth, in fact, does exist and that we are actually testing their ability to find it. We also employed high redundancy of the annotations in the validation step—10 for each validation—to limit the effect of spam turkers.

For our corpus of deceptive online reviews, we have arbitrarily but not without motivation selected the following dimensions: *domain*, *sentiment*, and *deception type*.

Among the many possible domains for online reviews, we have included in our corpus reviews of electronics and appliances and reviews of hotels (`Electronics` and `Hotels`)—they appear to be *orthogonal*, valuable, and well-represented online.

A well-studied linguistic dimension of online reviews is the *sentiment* (`Pos` or `Neg`) attributed as the overall polarity of the opinion expressed in the review. We expect that it should be possible to identify commonalities or differences among the invariants of deception across different sentiment values—reducing the bias arising from studying deception only in positive reviews.

In order to better simulate the conditions under which real reviews are written, we asked turkers writing truthful (i.e., `T`), `Pos` or `Neg`, reviews to describe objects they know. We also asked the same turkers to produce a *lie* about the same object. This adds another value along the deceptive dimension that we will call `F` (i.e., *false* reviews). By using the same objects reviewed by turkers while producing the `T`s and the `F`s labels we also asked turkers to review those objects and *fabricate* a fake review, both `Pos` and `Neg`, for such unknown object. We called such reviews along the deception type dimension *deceptive* and labelled them with `D`. To ensure that those were true deceptive reviews we asked the turkers explicitly to confirm that they didn't have previous experience with that given object (e.g., an hotel). Both `F`s and `D`s are indeed deceptive with the crucial difference of knowing or not knowing the object reviewed.

Because of the way in which we collect the reviews for our

[5]Computer users recruited to work on small web-based tasks via AMT are called *turkers*.

corpus, we implicitly introduce a *latent* quality dimension (good and bad)—the quality of the object reviewed. We ensure that half of our Ds are collected using the URLs provided during the PosT task (i.e., the good objects) and the other half, from URLs harvested during the NegT task (i.e., the bad objects). Table 1 is a summary of all the corpus dimensions and their possible values.

DIMENSION	VALUES
domain	Hotels, Electronics
sentiment	Pos, Neg
deception	T, F, D
quality	good, bad

Table 1: The three corpus dimensions, and quality—the latent dimension.

It is important to note that these three dimensions are not the only dimensions which could be considered. For instance, explicitly considering *age* and *gender* of the writer might be two other obvious extensions of this corpus.

3.2. Guideline Challenges

Because, among the others, we wanted to collect lies (i.e., Fs) we started a pilot study in which we asked turkers to think of a hotel they did not like and write a positive review of it (i.e., a lie). After inspecting the results we started questioning whether or not these reviews were actually lies and not just actual positive reviews. It is known in the literature (Walczyk et al., 2003) that telling lies is cognitively more complex, and we conjectured that a turker obeying the cognitive economy principle would, instead of lying about an actually negative experience, would write a review based on an actual positive experience—truth is much easier to generate. For this reason, we conceived of a generic *cognitive trap* that should increase the likelihood that an F is actually a lie. Our intuition was that it would be easier for a turker to generate a lie having first generated a truthful review about the same object. We therefore asked turkers to write a truthful review, either positive or negative, and then to write a review of the same object with the opposite sentiment polarity, which should therefore be a lie.

In this preliminary phase, we also experimented with tasks using turkers to measure the quality of the reviews written by other turkers. We designed a *cooperative* task in which we asked turkers, given a review and a description of the task for which it was written, to determine whether the person who wrote the review did what was asked. This task evolved into a simplified *quality* task avoiding the subtleties related to the details of the elicitation task.

To avoid introducing artificial constraints we did not require any specific length for the reviews—we decided to leave it open, and instead of providing an actual number, we used phrases such as "in the style of those you can find online"—which, as we know, have high variability in length. We also decided to avoid strictly defining what a *good* review is and instead provided vague directions which rely on the turkers' experience using expressions like: "needs to be persuasive", "sound as if it were written by a customer" and "informative."

Because of the amount of Fs and Ts we ended up with more objects available than we needed to elicit Ds for, so we downsampled the URLs, also partially normalizing and deduped them to avoid to creating too many Ds for the same object (e.g., iPhone).

In (Salvetti, 2012) the reader can find the transcript in plain text and the actual HTML for all the guidelines used to elicit the BLT-C data.

3.3. BLT-C Creation

In the initial phase turkers were asked to generate pairs of reviews about objects (Hotels or Electronics) about which they had actual direct experience. Truthful reviews (i.e., T) reflected the writers true sentiment towards the object in question, either positive (Pos) or negative (Neg). Deceptive reviews (F) (i.e., lies) reflected sentiment opposite to the reviewers experience with the object. Such a task was divided in two, the first half in which we elicited truthful positive (PosT) and deceptive negative (NegF) reviews and the other half eliciting truthful negative (NegT) and deceptive positive (PosF) reviews. Turkers provided information about the objects in question in the form of a URL. To generate the fake, or fabricated, reviews an additional set of turkers were provided with the URLs from the first phase and instructed to fabricate both positive (PosD) and negative (NegD) reviews for a given object. Turkers were also explicitly asked to confirm that they were not familiar with the object provided to them. In the end, we had 625 unfiltered D reviews, which, added to the 956 T and F reviews, gave us a total of 1,583 unfiltered reviews.

3.4. BLT-C Validation and Filtering

Since turkers can (and do) cheat to speed up their work and maximize their economic benefit, we implement several methods to validate the elicited reviews. Turkers can provide reviews that are too short, they can replicate the same review multiple times, they can provide something which is not a review, they can make mistakes, or they can simply cut-and-paste a review from a legitimate review aggregator. To filter out reviews with these problems, we employ a set of *safeguards* to increase the likelihood of including only *good* reviews in our corpus, such as checking for plagiarism and intrinsic quality.

The validation and filtering of our corpus is performed automatically and also by having elicited reviews labelled for truthfulness by a set of judges (in our case, turkers) without prior exposure to the set and without knowledge regarding the relative distribution of truthful vs. deceptive reviews. The general expectation is that untrained humans should perform at chance or below and should show a definite bias toward the *truthful* label in a balanced corpus.

Reviewing the work of the turkers—1,583 reviews spread fairly uniformly over the three dimensions—it was clear that not all the reviews were appropriate for inclusion in the final corpus. A few were simply garbage text, some were not reviews, others were clearly plagiarized from review websites, others were far too short. After automatically eliminating plagiarized reviews based on search engine results, we developed three new turker-based tasks for measuring for each review: the initial rating (i.e., sentiment),

its deceptiveness, and its quality. For each of these tasks we used 10 distinct judges providing a score. The objective was to eliminate reviews with incorrect sentiment, or poor quality and to leverage the deceptiveness test for general validation. Each of our 1,583 reviews received a total of 30 judgments spread across the three tests, for a total of 47,430 individual tasks performed by turkers. In fact, there are even more judgments because the PosT and NegT reviews were used for both the *lie or not lie* and *fabrication* tasks in order to have variation between deceptive and non-deceptive reviews in both tasks. This added another 4,800 judgments, for a grand total of 52,230 judgments collected, and 53,811 turker assignments, if we also count the review elicitation assignments. We applied a reasonable set of filters with levels of thresholding (details in (Salvetti, 2012)) on the data and marked some of the reviews as REJECTED. Because all information is still present in the BLT-C each experimenter can employ a different filtering approach. We observed a length bias across various classes, nevertheless, we decided to keep this bias in our corpus—it can always be eliminated by down-sampling. The main reason for keeping it is that it appears to be an actual feature of deception, and hence, it should be considered in modeling. The full set of reviews is available free of charge from LDC as the Boulder, Lies and Truth Corpus[6]. The structure of the corpus after filtering out 91 *low quality* reviews is shown in Table 2.

DOMAIN	REVIEW-PAIR	REVIEW-PAIR	TOTAL
Hotels	PosT, NegF 229	PosD, NegD 154	383
	NegT, PosF 218	PosD, NegD 148	366
Elect.	PosT, NegF 224	PosD, NegD 154	378
	NegT, PosF 219	PosD, NegD 146	365
TOTALS	890	602	**1,492**

Table 2: Filtered corpus content by domain and review-pair.

In Table 3, we see that overall we have more Ds (both Pos and Neg) than Ts or Fs. This is not surprising since we elicited many more D reviews. The final corpus is not internally balanced respect to the three dimensions. Since the corpus is intended to be used through projections (e.g., the PosT), it is only worth balancing the projections as needed. Because the corpus can also be used as a *sentiment* corpus, we report in Table 4 the review counts paired with their sentiment and deceptive labels.

As a final step in validating the corpus, we measure the intrinsic difficulties that humans have in detecting deception. Literature in deception detection demonstrates that humans perform at chance or even below chance when trying to detect deception. Therefore, in a valid corpus of texts it should be difficult for humans to distinguish *truthful* from *deceptive* documents.

Because of the truth bias, there is a higher probability of a given document being judged truthful. This bias is reflected in our corpus by 67.81% *truth*, 32.19% *deception* breakdown in human judgments. Moreover, the overall accuracy restricted to the balanced mixed set of all Ts and all Fs is 51%, that is indeed, chance. Our corpus exhibits truth bias *and* chance-level performance by humans trying to discriminate *truths* from *lies* within the corpus, which we adduce as evidence that is a valid representation of deceptive human behavior.

[6]https://catalog.ldc.upenn.edu/LDC2014T24

LABEL	Hotel	Electronics	TOTALS
PosT	116	112	228
NegF	113	112	225
NegT	113	110	223
PosF	105	109	214
PosD	151	152	303
NegD	151	148	299
TOTALS	749	743	**1,492**

Table 3: Filtered corpus, total number of reviews and breakdown by category.

		SENTIMENT		
		Pos	Neg	TOTAL
DECEPTION	T	228	223	451
	F	214	225	439
	D	303	299	602
		745	747	**1,492**
		TOTALS		

Table 4: Filtered corpus content totals for sentiment and deception reviews.

3.4.1. Validation by Experimentation

The final mechanism we employed to validate our corpus relies on using supervised machine learning to create a set of binary classifiers trained on data extracted from some of the possible projections of our corpus.

We define a *projection* of our corpus as the subset for which specific dimensions are fixed; for example, the D projection of our corpus is the subset of reviews with the value D for the truth-class dimension, while the HotelsPos projection is the subset of reviews with value Hotels for the domain dimension and Pos for the sentiment dimension.

We take the accuracy of binary classifiers trained and tested on pairs of projections of our corpus to be a *measure* of separation and hence *distinguishability* of such projections. These measures of separation can be used to validate our corpus by comparing them with earlier published results. For instance, we can train a classifier to distinguish the projections of our corpus with respect to the sentiment dimension and compare the performance of this classifier with similar work on sentiment analysis. These measures also provide us with some preliminary insights regarding the feasibility of deception detection using our corpus.

We employ three classifiers (two versions of Naïve Bayes and a decision tree classifier) which we use with the same *settings* in all our experiments to derive a *metric* for measuring separability. Note, our intent here is not to improve on the state-of-the-art in opinion spam detection, but rather to validate the corpus.

If a binary classifier trained and tested on two sets of labels of the same cardinality achieve 50% accuracy (measured using an n-fold cross validation), we say that the two sets are *indistinguishable*—they have no separation in the given feature space. All of our measurements are carried out using the so-called *bag of words* as the feature space.

To ensure that the performance of our classifiers is sufficiently high to allow us to draw meaningful conclusions, we tested them against the Cornell corpus (Ott et al., 2011). We show that our classifier performs as well as those proposed in (Ott et al., 2011), reaching an accuracy of 88% when trained and tested on the set of *truthful* and *deceptive* reviews of the Cornell corpus using unigrams as features.

The performance of the same classifiers trained on different projections of the BLT-C varies, from close to chance when trained and tested on `T` vs. `F` to virtually perfect when trained and tested on `Electronics` vs. `Hotels`.

Our results show that for a machine learned classifier, deceptive documents (i.e., `F`) are almost indistinguishable from truthful ones (i.e., `T`), just as they are for human judges. They also confirm our hypothesis that the high degree of separation seen in (Ott et al., 2011) is likely a side effect of corpus-specific features (i.e., characteristics of the writers). Nevertheless, we conjecture that subtle differences between truthful and deceptive documents do exist and that more sophisticated feature engineering is needed to improve performance to better than chance.

As our general machine learning apparatus, we employ Weka[7], and specifically, three of its classifiers: two different implementations of Naïve Bayes that we refer to as Naïve Bayes (John and Langley, 1995) and Multinomial Naïve Bayes (Mccallum and Nigam, 1998) and the J48 decision tree classifier, which is an implementation of the well-known C4.5 classifier (Quinlan, 1993). We use as features the counts of 10,000 words.

To build a binary classifier, Weka takes as input a directory structure in which each directory represents a class (with the directory names serving as class labels) and each directory contains a separate file for each document in the class. Weka also provides a standard converter, which takes as input a directory of directories and produces as output an ARFF (Attribute-Relation File Format) file, which can be further converted into actual feature vectors. It is then possible to use the file containing all the feature vectors for the classes to perform n-fold cross validation.

To facilitate replication, we report in Listing 1 the sequence of command-line commands required to build and test a classifier (e.g., Multinomial Naïve Bayes).

We then used the precision of a certain classifier on a certain project as measure of separation between sets. The total number of corpus projection pairs is $36 \times 36 = 1,296$. However, most of these projection pairs are meaningless (e.g., (`T`, `Hotels`)) or at least not interesting. Therefore, we select those corpus projection pairs in which only one of the dimensions *changes* (e.g., (`HotelsPosD`, `HotelsNegD`)). The total number of such projection pairs is exactly 51. For each of these pairs, we trained and tested a Naïve Bayes, a Multinomial Naïve Bayes, and a J48 classifier and record the accuracy of each. We then rank these projection pairs by accuracy. Each run was actually split in two, one using all the data available for that give projection pair and one with a balanced (i.e., partially reduced) dataset. The balanced datasets are obtained from the full sets by downsampling the larger of the two sets. The balancing is done at the review level and does not take into account the length of each review.

```
# assuming that weka-input is a directory whose content are
# directories representing the document classes (e.g., T and D)
# and that in each directory each document is stored in a separated file

# we start by converting such directory in a AIFF file
$ java weka.core.converters.TextDirectoryLoader -dir weka-input > o.arff

# we then transform the strings into word vectors
# we use counts as feature representation with up to 10,000 features
$ java weka.filters.unsupervised.attribute.StringToWordVector
        -C -i o.arff -o r.arff -W 10000

# we then train and test using a naive bayes classifier
# using the first data element as class and a 5 folds cross validation
$ java weka.classifiers.bayes.NaiveBayes -t r.arff -c first -x 5

# or a multinomial naive bayes
$ java weka.classifiers.bayes.NaiveBayesMultinomial
         -t r.arff -c first -x 5

# or a decision tree (i.e., J48)
$ java weka.classifiers.trees.J48 -t r.arff -c first -x 5
```

Listing 1: How to classify using Weka.

In Table 5, we report some of the 51 projection pairs, presenting the accuracy achieved by training and testing two different Naïve Bayes classifiers on the balanced version of the datasets. As standard settings we used: 10,000 unigram features with count as the feature representation, no stemming, no down-casing, add-one smoothing, stop words preserved, and 5-fold cross validation.

CORPUS PROJECTION PAIR	ACCURACY
ElectronicsPos vs. HotelsPos	99.87%
Electronics vs. Hotels	99.73%
HotelsPos vs. HotelsNeg	94.49%
Pos vs. Neg	90.88%
HotelsPosD vs. HotelsPosF	70.00%
HotelsT vs. HotelsD	67.17%
NegT vs. NegF	66.29%
HotelsNegT vs. HotelsNegF	65.93%
HotelsD vs. HotelsF	65.14%
T vs. D	63.61%
(*) T vs. D	60.62%
D vs. F	56.83%
PosT vs. PosF	53.74%
(*) T vs. F	51.14%
T vs. F	42.71%
ElectronicsT vs. ElectronicsF	39.14%

Table 5: Ranked accuracy on selected corpus projection pairs for the Naïve Bayes Multinomial classifier using unigrams and counts as the feature representation on balanced versions of the projection pairs. Results marked with (*) are for the Naïve Bayes classifier—for comparison.

These results reflect our expectations—the extremely high separation between domains (electronics and hotels are quite orthogonal), the high separation along the sentiment dimension, which matches other published results in the literature (Salvetti et al., 2004), and the statistical separation between the `D`/`T`) projection pairs, which confirms our ex-

[7] http://www.cs.waikato.ac.nz/ml/weka/

pectations by matching, though to a lesser degree, what is reported in (Ott et al., 2011). The fact that Ts and Fs are not separable in the unigram space is also not surprising and matches our expectations. By eliminating some of the biases in (Ott et al., 2011) (e.g., differences in the writers and their motivations), we see that what cannot be separated by human judges also cannot be easily separated by a machine—lies are indeed tough to detect (Vrij, 2008). We do not claim that there is no separation but that any separation that may exist is subtle, as we expected. Overall, these results match our expectations and increased the likelihood of the validity of the BLT-C.

3.4.2. Discussion on Measuring Separations

In this section, we present similar results regarding measures of separation between the two Cornell datasets (Ott et al., 2011)—deceptive reviews elicited on AMT and truthful reviews harvested from TripAdvisor—and some of the pertinent projections of the BLT-C.

The separation between BLT-C Ds and Ts is much lower than the separation we measured on the Cornell corpus, which is 88%. Such reduced separation confirms our hypothesis that the high separation in the Cornell corpus is mainly due to the effect of differences in the authors. Remember that the BLT-C Ts are elicited from turkers whereas the Ts in the Cornell corpus are actual (at least, supposedly) positive reviews collected from users of TripAdvisor who are customers of the top 20 hotels in Chicago. We argue that there is therefore a clear socioeconomic difference between the two groups. There might also be some difference due to inner motivations for writing the review itself: on the one side, payment of $1 for an elicited review; on the other, a true desire to share a positive experience with an audience. We also conjecture that even this residual separation between Ts and Ds is not necessarily due to a difference in the actual deception dimension but might be due merely to differences in the amount of knowledge about the hotels themselves or, even more importantly, differences in emotional involvement with the objects, which can be the cause of variations in the word usage and are only tangentially related with possible linguistic deception invariants. This conjecture is confirmed in Table 5 by the fact that our classifiers perform at chance, or below, when trying to separate Ts and Fs. For these reasons we argue that the study of deception invariants using fabricated reviews might be less effective in helping to isolate invariants than studies employing actual lies—lies, in fact, do not have some of these problems. In the lies we collected, there is explicit knowledge about the object described, and there is also some emotional involvement with it—which is totally missing from *all* cases of fabrication.

This leads to the next observation we can make using our data, which is that the separation between truths and lies is marginal, at least using unigram features. This is confirmed in Table 5 by the fact that our classifiers perform at chance, or worse, when trying to separate Ts from Fs. This buttresses our assertion that differences in deception are much more subtle when other co-occurring but unrelated signals are eliminated.

Specifically, when motivation, objective knowledge, and individual attributes and idiosyncrasies are controlled for, truth and lie become indistinguishable. We may imagine, and hope, that there are actually intrinsic differences between Ts and Fs but in order to detect such nuances more sophisticated analysis is needed. The fact that there is no easily detectable difference between Ts and Fs using the bag-of-words model suggests that future successful results on this set are likely to be the result of actual understanding of the deception invariants.

It is also interesting to note the separation between Ds and Fs (e.g., 70% for HotelsPosD vs. HotelsPosF). We conjecture that such a difference is only partially due to a difference in type of deception (i.e., fabrications vs. lies) and that probably at its core the reason for the separability is the same as for the separability of Ds and Ts—different amount of knowledge and lack of emotional involvement. Overall, in fact, the separation between Ts and Ds is similar to the separation of Fs and Ds, suggesting that fabrication is a much easier deception dimension value to identify than actual lies.

3.5. Potential Confounders in the BLT-C

Any corpus dealing with such complex psychological and textual phenenoma is bound to beset by confounders. These are a few:

- the skewed distribution of the number of reviews per writer. The reviews were been written overall by 497 distinct writers, with an average of just a bit more than 3 reviews per writer;
- for each of the twelve possible corpus labels (e.g., HotelsPosT, ElectronicsNegD, etc.) there would ideally be no more than one review written by the same turker. This is true for all of the Ts, all of the Fs, and some of the Ds, for a total of 1,210 reviews. However because of the intrinsic limitation of AMT, it was impossible to control the number of reviews written by a single turker for the D tasks;
- the remaining 373 reviews have some writer overlap within the same corpus label; for instance, one turker wrote 9 reviews for HotelsPosD instead of just one. Were these extra reviews written by the same writer within the same corpus label eliminated, 268 reviews, all Ds, would fail. This version of the corpus would then contain 1,315 unfiltered reviews.

Because the BLT-C contains all information collected it is then possible to create new experimental versions in which for instance certain turkers are eliminated. This is possible because all reviews are labelled with the turker anonymized identifier.

4. Conclusions

Deception is a complex, pervasive, sometimes high-stakes human activity; while the linguistic implementation of deceptive acts by speakers is sometimes brazen, sometimes subtle, it is a curious fact that it is difficult for other humans to detect, and at any rate humans exhibit a distinct bias towards believing what they are told.

Although human performance at detecting deception is at chance, this research and prior research suggest that the unconscious linguistic signals included in a conscious act of deceiving are sufficient to allow us to build automatic systems capable of successfully distinguishing deceptive documents.

We have focused our research on the definition, design, and creation of an extensible, demonstrably valid, and balanced text resource for the study of deception, with an apparatus (in the forms of algorithms, statistical tests, and procedures) to extend the research into new dimensions. The result is one of the largest publicly-available multidimensional deception corpus for online reviews, containing nearly 1,600 reviews in the style of those that can be found online. In an attempt to overcome the inherent lack of ground truth—since it is not possible to know for sure whether someone is lying—we have also developed a set of automatic and semi-automatic techniques to increase our confidence in the validity of the corpus.

Detecting deception using supervised machine learning methods is brittle. Experiments conducted using the BLT-C show that accuracy changes across different kinds of deception (e.g., lying vs. fabrication), demonstrating the limitations of previous studies. Preliminary results confirm statistical separation between fabricated and truthful reviews, but they do not confirm the existence of statistical separation between truths and lies.

The fact that there is no easily detectable difference using statistical models based on the bag-of-words model suggests that future successful results on this corpus will most likely be the result of actual understanding of deception invariants. More importantly, the preliminary results of the analysis of our corpus suggest that identification of deception in cases of lying reviews with explicit knowledge about the object under review is much harder than the identification of fabricated reviews, which supports our thesis that deception is a multifaceted phenomenon that needs to be studied in all its possible dimensions by means of a multidimensional deception corpus like the BLT-C.

5. Bibliographical References

Bachenko, J., Fitzpatrick, E., and Schonwetter, M. (2008). Verification and implementation of language-based deception indicators in civil and criminal narratives. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 41–48.

Bond, Jr., C. F. and DePaulo, B. M. (2006). Accuracy of deception judgments. *Personality and Social Psychology Review*, 10(3):214–234.

Conroy, N. J., Rubin, V. L., and Chen, Y. (2015). Automatic deception detection: Methods for finding fake news. In *Proceedings of the 78th ASIS&T Annual Meeting*, St. Louis, MO USA.

Ekman, P. (2001). *Telling Lies, Clues to Deceit in the Marketplace, Politics, and Marriage*. W. W. Norton & Co., New York, 2nd edition.

Enos, F., Shriberg, E., Graciarena, M., Hirschberg, J., and Stolcke, A. (2007). Detecting deception using critical segments. In *Proceedings of Interspeech*.

Freud, S. (1901). *Psychopathology of everyday life*. T. Fisher Unwin, London.

Granhag, P. A. and Strömwall, L. A. (2004). *The detection of deception in forensic contexts*. Cambridge University Press, New York.

John, G. H. and Langley, P. (1995). Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, San Mateo. Morgan Kaufmann.

Li, J., Cardie, M. O. C., and Hovy, E. (2014). Towards a general rule for identifying deceptive opinion spam. In *Proceedings of the 52th Annual Meeting of the Association for Computational Linguistics*.

Mccallum, A. and Nigam, K. (1998). A comparison of event models for naïve bayes text classification. In *AAAI-98 Workshop on Learning for Text Categorization*.

Mihalcea, R. and Strapparava, C. (2009). The lie detector: Explorations in the automatic recognition of deceptive language. In *Proceedings of the (ACL-IJCNLP) joint conference of the Asian Federation of Natural Language Processing*.

National Research Council. (2003). *The Polygraph and Lie Detection*. National Academies Press, Washington, D.C.

Newman, M. L., Pennebaker, J. W., Berry, D. S., and Richards, J. M. (2003). Lying words: Predicting deception from linguistic styles. *Personality and Social Psychology Bulletin*, 29:665–675.

Ott, M., Choi, Y., Cardie, C., and Hancock, J. T. (2011). Finding deceptive opinion spam by any stretch of the imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*.

Perez-Rosas, V., Mihalcea, R., Narvaez, A., and Burzo, M. (2014). A multimodal dataset for deception detection. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*.

Qin, T. T. and Burgoon, J. K. (2005). An empirical study on dynamic effects on deception detection. In *Proceedings of Intelligence and Security Informatics*.

Quinlan, R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo.

Salvetti, F., Lewis, S., and Reichenbach, C. (2004). Impact of lexical filtering on overall opinion polarity identification. In James G. Shanahan, et al., editors, *Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*, Stanford, US.

Salvetti, F. (2012). *Detecting Deception in Text: A Corpus-Driven Approach*. University of Colorado, Boulder.

Scott, W. (1806). *Marmion: A Tale of Flodden Field*. Project Guttenberg.

Vrij, A. (2008). *Detecting lies and deceit: Pitfalls and opportunities*. John Wiley & Sons, Ltd, Chichester, West Sussex P019 8SQ, England, second edition.

Walczyk, J. J., K. S. Roper, E. S., and Humphrey, A. M. (2003). Cognitive mechanisms underlying lying to questions: Response time as a cue to deception. *Applied Cognitive Psychology*, 17(7):755–774.

Xiao, B. and Benbasat, I. (2011). Product-related deception in e-commerce: a theoretical perspective. *MIS Quarterly*, 35(1):169–195.