# Analyzing Deceptive Opinion Spam Patterns: the Topic Modeling Approach

Alibek Jakupov, Julien Mercadal, Besma Zeddini, Julien Longhi

# Analyzing Deceptive Opinion Spam Patterns: the Topic Modeling Approach

Alibek JAKUPOV
*CY Cergy Paris Université*
Cergy, France
alibek.jakupov@etu.u-cergy.fr

Julien MERCADAL
*CY Cergy Paris Université*
Cergy, France
julien.mercadal@cyu.fr

Besma ZEDDINI
*CY Cergy Paris Université*
Cergy, France
besma.zeddini@cyu.fr

Julien LONGHI
*CY Cergy Paris Université*
Cergy, France
julien.longhi@cyu.fr

***Abstract*—Deceptive Opinion Spam commonly takes the form of fake reviews (negative or positive) posted by a malicious web user to hurt or inflate a company's image. As these reviews have been deliberately written to deceive the reader, human reviewers are faring little better than a chance in detecting these deceptive statements. Thus, there is a dire need to address this issue as extracting text patterns from the fraudulent texts with meaningful substructures still remains a challenge. In our research, to obtain a deeper understanding of how lies are expressed in texts, we consider the task as a topic modeling problem, in which we constructed a model to learn the patterns that constitute a fake review, and then explore the outputs of this model to identify those patterns. Topic models may be useful in this task due to their ability to group multiple documents into smaller sets of key topics. As the linguistic cues of the lies are still unknown, a key advantage of this approach is that the algorithm encourages the mixtures composed of only few topics, which makes the representation more interpretable and provides additional opportunities to reveal the patterns and structures within the systems of documents. Our methodology proved to be useful for this study, revealing the lexical cues generally applied by human reviewers to generate deceptive language.**

***Index Terms*—topic modeling, deceptive opinion spam, natural language processing**

## I. INTRODUCTION

With ever-increasing popularity of web reviews, there has been an explosion of web authorship from individuals which may contain false reviews or Opinion Spam. Opinion Spam is inappropriate or fraudulent reviews which can range from self-promotion of an unrelated website or blog to deliberate review fraud with a potential for monetary gain [1]. One of the main risks of Opinion Spam in terms of its impact on client opinion mainly concerns the reviews that falsely praise inferior products or criticize superior products as they may significantly impact a potential consumer's actions, therefore companies are highly motivated to automatically detect and remove Opinion Spam [2]. While other Natural Language Processing (NLP) tasks, like sentiment analysis or intent recognition, have received considerable computational attention, relatively little study has been made into detecting Opinion Spam using text classification techniques [3]. Some types of Opinion Spam are easily identified by a human reader, e.g. advertisements, questions or other non-opinion texts [4]. These instances belong to Disruptive Opinion Spam, irrelevant statements, that are evident to a reader and pose a minimal risk, since the user can always choose to ignore them [1]. However, for more insidious types of fictitious texts, like Deceptive Opinion Spam, the task is non-trivial, as these are the statements that have been intentionally produced to sound authentic and mislead the reviewer [1]. Deceptive Opinion Spam commonly takes the form of fake reviews (negative or positive) posted by a malicious web user to hurt or inflate a company's image [3]. As these reviews have been deliberately written to deceive the reader, human reviewers are faring little better than a chance in detecting these deceptive statements [5]. Thus, there is a dire need to address this issue as extracting text patterns from the fraudulent texts with meaningful substructures still remains a challenge [3].

The problem is generally treated as a text classification task. In most of the cases, text classification systems consist of two parts : a feature extraction component and a classifier. The former allows to generate features given a text sequence, and the latter assigns class labels to this sequence, given a list of corresponding features. Commonly such features include lexical and syntactic components. Total words or characters per word, frequency of large and unique words refer to lexical features, whereas syntactic features are mainly based on frequency of function words or phrases, like n-grams, bag-of-words (BOW), or Parts-Of-Speech (POS) tagging [6]. There also exist lexicon containment features which express the presence of a term from lexicon in the text as a binary value (positive=occurs, negative=doesn't occur) [7]. The lexicons for such features may be designed by human expert [8, 9] or generated automatically [10, 11]. Some approaches suggest using morphological links and dependency linguistic units in the text as input vector for the classifier [12]. In addition to this, there are semantic vector space models, which are used to represent each word with a real valued vector based on the distance or angle between pairs of word vectors, for a variety of tasks as information retrieval [13], document classification [14], question answering [15], named entity recognition [16] and parsing [17]. Besides these common linguistic features, there are also so-called domain-specific features, for instance, quoted words or external links [18]. There also exist methods based on Knowledge Graphs (KG), which suggest mapping of terms of the text into an external knowledge source, and allows a more effective extraction of patterns from noisy data [7]. This technique may be robust as it allows

integrating the external knowledge source and add common sense knowledge to the analyzer. All the feature extraction algorithms mentioned above may be served to examine the weights of the features, which allows the researchers to shed light on the commonality in the structure of deceptive opinion spam that is less present in truthful sentences. Although this approach proves to be useful it has some significant drawbacks because the quality of the training set is difficult to control and building a reliable classifier requires a considerable number of high-quality labeled texts [19]. Moreover, certain classification models based on the embeddings approach may be strongly impacted by social or personal attitudes present in the training data, which makes the algorithm draw wrong conclusions [20]. In certain cases inferences of an algorithm may be perfect on the training set and non-generalizable for new cases [21] which may represent serious challenges for Deceptive Opinion Spam detection. From this point of view weakly-supervised or unsupervised models based on topic modeling may perform better due to their better generalization capacity and independence from the training data [19].

*Our approach*

In our research, to obtain a deeper understanding of how lies are expressed in texts, we investigate the usefulness of the other approaches. Particularly, we consider the task as a topic modeling problem, in which we constructed a model to learn the patterns that constitute a fake review, and then explore the outputs of this model to identify those patterns. Topic models may be useful in this task due to their ability to group multiple documents into smaller sets of key topics. Unlike neural nets, which model documents as dense vectors, topic models form sparse mixed-membership of topics to represent documents, which means that most of the elements are close to zero. As the linguistic cues of the lies are still unknown, a key advantage of this approach is that the algorithm encourages the mixtures composed of only few topics, which makes the representation more interpretable and provides additional opportunities to reveal the patterns and structures within the systems of documents [22], as, for instance, revealing clusters of words within documents [23], highlighting temporal trends [24] or inferring networks of complementary products [25] to finally show the textual patterns expressing the deception. Our methodology proved to be useful for this study, revealing the lexical cues generally applied by human reviewers to generate deceptive language. As existing topic modeling is often based off Latent Dirichlet Allocation (LDA) [23], our first experiments were conducted using this method. LDA analyzes a given corpus to produce a distribution over words for each latent topic and a distribution over latent topics for each document. By applying LDA for deception detection, we extracted the underlying topics and the rules LDA forms to attribute topics to documents and words to topics for a better understating of textual patterns of lies. We then enriched this approach and experimented with lexical databases to find more general patterns and analyze the importance of external information for deception detection. However, as LDA does not take advantage of dense word representation which can capture semantically meaningful regularities between words, we extended our research to other algorithms which can take advantage of word-level representations to build document-level abstractions, such as lda2vec [22]. lda2vec extends Skip Negative Sampling (SGNS) [26] to jointly train word, document and topic vectors and embed them in a common representation space which takes into account semantic relations between the learned vector representations. At the same time, this representation space yields sparse document-to-topic proportions, which allows us to interpret the vectors and draw the conclusions on the nature of deception. We also applied the state-of-the-art Bidirectional Encoder Representations from Transformers (BERT) [27] for topic modeling to see how pre-trained models may impact the overall quality and evaluate the importance of accurate representation of words and sentences.

*Our contributions*

Our contributions can be summarized as follows.

- An analysis of the previous approaches from the point of view of pattern extraction and not model training
- A new approach to extract the deceptive patterns based on topic modeling
- A novel technique to integrate the ontology into topic modeling

*Outline:* The rest of the paper is organized as follows: in Section II we provide an overview of related work; in Section III we summarize our methodology for topic modeling; in Section IV we present and discuss experimental results as well as the datasets used to benchmark our approaches; finally, conclusions and discussions are provided in Section V.

## II. Related Work

Ott *et al.* were the first to address this issue by applying the Machine Learning approach [1]. One of the important contributions of their work is the proof of the necessity of considering both the context and motivations underlying a deception, instead of focusing purely on a pre-defined set of deception cues, like Linguistic Inquiry and Word Count (LIWC), which is widely used to explore personality traits [28] and investigate tutoring dynamics [29]. Accordingly, they combined features from the psycho-linguistic approach and the standard text categorization approach, and succeeded to achieve the 89.8% performance with the model based on LIWC and bigrams. Nevertheless, these features are not robust to domain change, as they can do well only if training and testing datasets are of the same domain [30, 2]. For instance, simply shifting the polarity of the reviews for training and testing (i.e. training on positive reviews and testing on negative ones) significantly dropped the overall performance of the model [31]. Topic modeling, in this context, is more flexible as it has been demonstrated by Blei *et al.* when they applied their model to the domains as diverse as computer vision, genetic markers, surveys and social network data [23].

Li *et al.* succeeded to obtain 81.8% on Ott dataset by capturing the overall dissimilarities between truthful and deceptive

texts [32]. In their research they extended Sparse Additive Generative Model (SAGE) [33], a generative bayesian approach, which combines topic models and generalized additive models and creates multi-faceted latent variable models by adding together the component vectors. As most of the researches in the domain focus on detecting the deceptive patterns instead of training a single reliable classifier, the main challenge is to identify the features contributing the most to each type of deceptive review and evaluate the impact of such features on the final decision when combined with the other features. SAGE fits these needs due to its additive nature, whereas other classifiers may suffer from the domain-specific properties in cross-domain scenarios. The authors found out that the BOW approach is less robust than LIWC and POS modeled using SAGE and constructed the general rule of deceptive opinion spam with these domain-independent features. Moreover, unlike Ott *et al.* [1], who considered the lack of spatial data in the hotel reviews as a cue to find the deceptive patterns, Li *et al.* proved that this may not be a universal case, since some fake reviews may be written by domain experts. Additionally, according to their findings, another interesting cue to deceptive opinion spam is the presence of sentiments, as reviewers tend to exaggerate sentiment by using more sentiment-related vocabulary in their statements. Although the domain-independent features extracted during the research proved to be efficient and allowed to detect fake reviews with above-chance accuracy, it has been demonstrated that the sparsity of these features makes it complicated to leverage non-local discourse structures [34], thus the trained model will be unable to capture the global semantic information of a document. In our research, besides testing probabilistic topic models [23], generating sparse vectors to model documents, we also applied hybrid techniques [22], mixing sparse representations with dense topic and word vectors, taking into account the semantical relations between words.

Ren and Ji [34] expanded the previous work by proposing a three-stage system. At first, they constructed sentence representations from word representations with the help of convolutional neural network, as the convolution action has been generally applied to synthesize lexical n-gram information [35]. For this step they applied three convolutional filters as they are capable of capturing local semantics of n-grams, such unigrams, bigrams and trigrams, an approach that has been already proven successful for such tasks as sentiment classification [36]. After that they modeled the semantic and discourse relations of these sentence vectors to construct a document representation using a bi-directional gated recurrent neural network. These document vectors are finally used as features to train a classifier. The authors achieved 85.7% accuracy on the dataset created by Li *et al.* and proved that neural networks may be applied to learn continuous document representations to better capture semantic characteristics. The main goal of this study was to empirically demonstrate the better performance of neural features over traditional discrete features (like n-grams, POS, LIWC, etc.) due to their stronger generalization. Nevertheless, further experiments conducted by the authors showed that by integrating discrete and neural features the overall accuracy may be improved, thus discrete features still remain a rich source of statistical and semantic information. It therefore follows that jointly trained word, topic and document vectors, represented in a common vector space may improve the overall accuracy of deceptive spam classifiers.

Vogler and Pearl [2] investigated the use of specific details for detecting deception both within-domain and across-domains. The linguistic features they covered in the research included n-grams, POS, syntactic structure, measures of syntactic complexity, stylometric features, semantically-related keyword lists, psychologically-motivated keyword lists, sentiments, discourse structure and named entities. The authors claim that these features are not robust enough, especially in cases where domain may very significantly, as most of them tend to rely on cues that are very dependent on specific lexical items, such as n-grams or specific keyword lists. Though there are some linguistically abstract features like POS, stylometric features or syntactic rules, the authors consider them as less relevant as they are not motivated by the psychology of verbal deception. In their research they considered deception as an act of imagination, and besides analyzing the linguistic approaches they also explored psychologically-motivated methods, such as information management theory [37], information manipulation theory [38], reality monitoring [39] and criteria-based statement analysis [40]. As more abstract psychologically-motivated linguistic cues may be more applicable across domains [41] the authors find it useful to apply these cues with a basis in the psychological theories of how humans generate deceptive texts. They have also relied on the results provided by Krüger *et al.* whose research focuses on subjectivity detection in newspaper articles and suggests that linguistically abstract features may be more robust when applied to the cross-domain texts [30]. For the experimentation Vogler and Pearl utilized three datasets for training and testing with domain changes varying from fairly minimal to quite broad, the Ott Deceptive Opinion Spam Corpus [1], essays on emotionally charged topics [42] and personal interview questions [37]. The linguistically-defined specific detail features the authors constructed for this research proved to be useful when training and testing domains vary significantly. The features are based on prepositional phrase modifiers, adjective phrases, exact number words, proper nouns and noun modifiers that appeared as consecutive sequences. Each feature is represented as the total normalized number and the average normalized weight. They succeeded to achieve the best F-score of 0.91 for the cases when the content doesn't change and the best F-score of 0.64 when the content domain changes most significantly, which demonstrates that the linguistically-defined specific detail features are more generalizable across domains. However, even if the classifier trained on these features had fewer false negative, it poorly classified the truthful texts. As it may be seen from the experimentation results, a mix of n-gram and linguistically-defined specific details features tends to be more reliable only in case the false positive is more costly

than false negative. It should also be mentioned that the n-gram-based features may have more semantic generalization capacity when based on distributed meaning representations, such as GloVe [43] and ELMo [44], whereas n-gram features in their approach are based on individual words and do not capture the semantic relatedness. This is in contrast to our approach, as we suggest utilizing the distributed meaning representations by learning the linear relationship between words.

Barsever *et al.* built a state-of-the art classifier using BERT and then analyzed this classifier to identify the patterns BERT learned to classify the deceptive reviews [3]. BERT is a neural network architecture pretrained on millions of words and using the Masked Language Modeling (MLM) by jointly conditioning on left and right context in all layers to train deep bidirectional language encoding [27]. The main advantage of BERT is the fact that it learns rules and features unsupervised, which allows BERT looking for the best solution unrestricted by preconceived rules. With their model, Barsever *et al.* achieved an accuracy of 93.6%, which proves the existence of features allowing to distinguish truth and deception. As the main goal of the research was to find rules and patterns of deceptive language, the authors performed an ablation study, by removing each POS and monitoring the performance of the network. Moreover they detected so-called 'swing' sentences, which are more important than the others for the classifier, to run POS analysis on them and shed light on the inner rules BERT constructed. Finally, the authors created the Generative Adversarial Network (GAN) based on their BERT model, whose goal is to fool the classifier to find out the trends reproduced in the synthetic data. Their findings indicate that certain POS (e.g. singular nouns) are more important for the classifier than the others and that truthful texts are more rich from the point of view of the variance of POS, whereas the deceptive reviews are more formulaic. Nevertheless, the approach applied by Barsever *et al.* may have some important challenges. In fact, the main disadvantage of BERT is the absence of independent sentence embeddings, which can play an important role as a higher means of abstraction [45]. Not surprisingly, the authors had to manually remove sentence by sentence from the initial dataset by replacing them with [MASK] tokens, and excluding the one-sentence entries. In addition, the rules generated by BERT are still unclear for the authors, and the results of the ablation study may reveal the other commonalities instead of identifying the patterns of the deception. For instance, the removing of the singular nouns resulted in a significant drop in the model's performance, which is interpreted as a strong weight of this POS in the classifier. We can nevertheless infer from these results that due to the prevalence of nouns in the speech, replacing them may also result in incomprehensible texts, which the classifier can hardly interpret. In this context, the LDA vector is much easier to reason about, due to its sparsity [22]. As the element of the document vector generated by topic modeling are non-negative and sum to 100%, this makes it more robust for deception detection, as a review may be partly truthful.

## III. Model

This section describes the methodology for our approach. The topic modeling we applied is based on LDA, its extension lda2vec, LDA with Wordnet and BERT-based text clustering. LDA is a hierarchical Bayes model utilizing Dirichlet priors to estimate the intractable latent variables of the model. The Dirichlet distribution is a distribution over probability mass functions with a pre-defined number of atoms, whose main property is conjugacy to the multinomial distribution. In LDA each document (a deceptive review in our case) in the collection is represented as multinomial distribution over topics and each topic is represented as multinomial distribution over words. The model simultaneously learns the topics by iteratively sampling topic assignment to every word in every document (in other words calculation of distribution over distributions), using the Gibbs sampling update. Each word-to-topic and topic-to-word distribution is then drawn from its respective Dirichlet distribution. For instance, given the corpus of online reviews, the generative process results in two outputs. The first corresponds to the topics-per-document distribution [46]:

$$document1 - pencil^{topic1}, pencil^{topic1}, umpire^{topic2}$$

$$document2 - ruler^{topic1}, ruler^{topic1}, baseball^{topic2}$$

whereas the second represents the terms-per-topic distribution:

| Topic1 | Topic2 |
|---|---|
| pencil | baseball |
| ruler | umpire |

In our approach we use both assignments for the deceptive opinion spam analysis. By limiting the number of topics to 2, we try to define two clusters. We assume that one cluster will correspond to the deceptive reviews and the other to truthful ones. The process is shown on the Figure 1.

LDA predicts globally which means that, unlike the embedding approach, it learns the representations predicting words inside of that document [22]. A generative model is favoured in our case because it doesn't require any strong assumptions about the link between the text and the class, and utilizes the pure distribution of words to mathematically model topics. This results in sparse simplex vectors (topics-per-document distributions) which are more interpretable than dense word vectors. These topics-per-document distributions are our main source of analysis, allowing us to uncover the rules LDA has drawn to cluster the corpus into two topics. Thus, to split the reviews into truthful and deceptive we first run the LDA over the whole corpus using the generative process described above (steps 1-2), and then take the topics-per-document distribution and apply the algorithm 1 on this distribution (3-4):

By default the first topic is set to truthful, but this is only needed to initialize the process. We then compare the predicted column (5-6) with the pre-defined label column and in case of anomaly we re-run the same process by changing the condition.

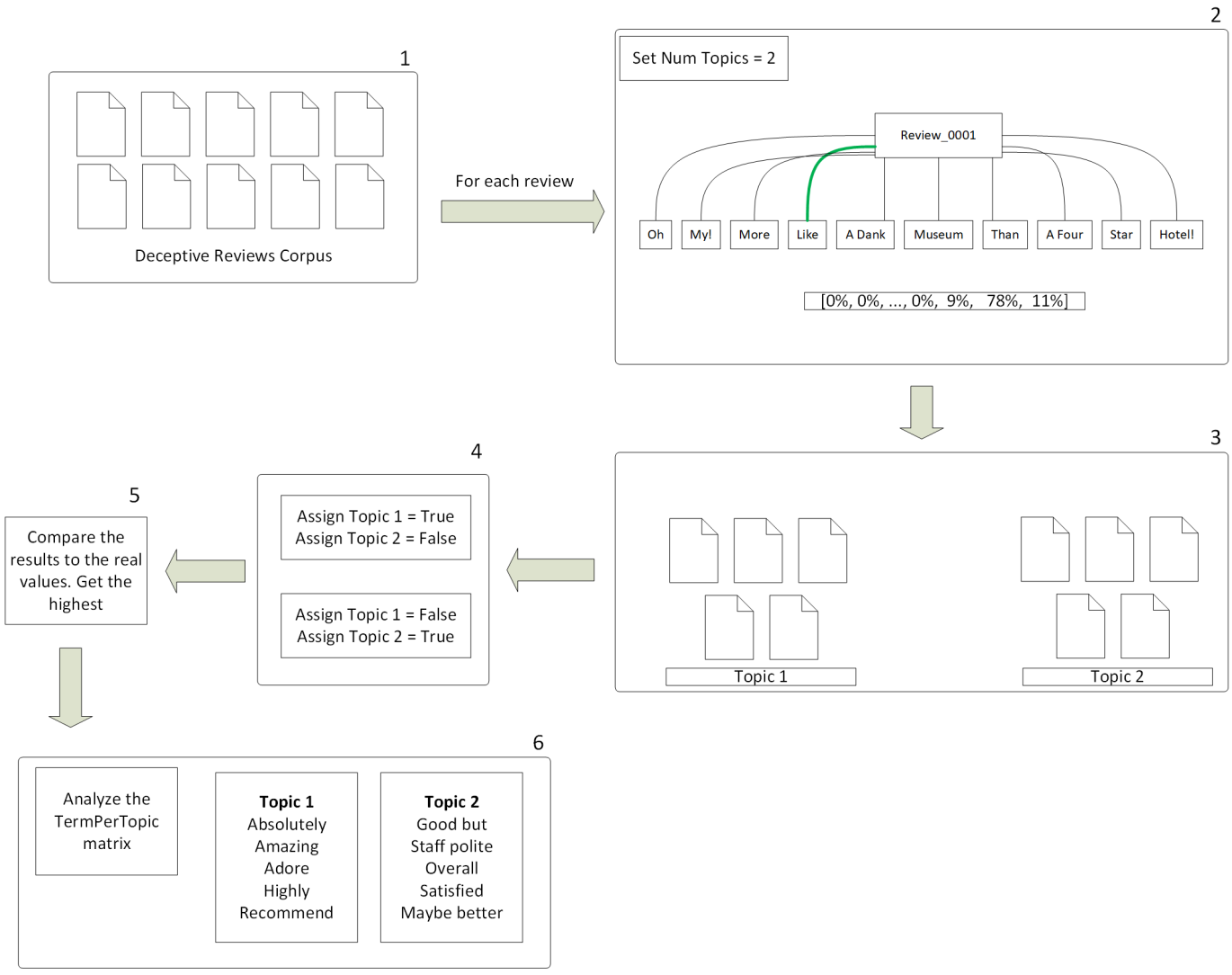


Fig. 1. LDA-based Topic Modeling for Deceptive Opinion Spam

**Algorithm 1** Clusterize the reviews into truthful and deceptive

```
topicsPerDocument ← LDA(corpus)
for all items ∈ topicsPerDocument do
    topic ← max(item.probabilities)
    if topic == Topic1 then
        predictedClass ← truthful
    else
        predictedClass ← deceptive
    end if
    item.class ← predictedClass
end for
```

Incorporating linguistic knowledge into analysis can also lead to better results [47]. The source of such knowledge is generally a large lexical database, as Wordnet, where nouns, adjectives, verbs and adverbs are grouped together into synsets, sets of cognitive synonyms expressing a distinct concept and interlinked by means of conceptual-semantic and lexical relations [48]. The most frequently encoded relations among these synsets is called hyperonymy which represents super-subordinate relation. Thus, we re-implement the previous model by replacing each term by its hypernym (a broader label) to retrieve more general patterns, as certain deception cues may be expressed at a higher concept-level.

We also extend the model by adding semantic relations between the learned vector representations to see whether word embeddings can improve the accuracy. To achieve this, we applied lda2vec which predicts the given word using both context and global document topics. The overall process remains the same except the step 2, which is described in Figure 2.

Thus, instead of learning a document vector that predicts words inside of that document, like in our first model, we try predicting the given term (e.g. ”like”) using both words from its context (”Museum”, ”Star”, ”Hotel” etc.) and global

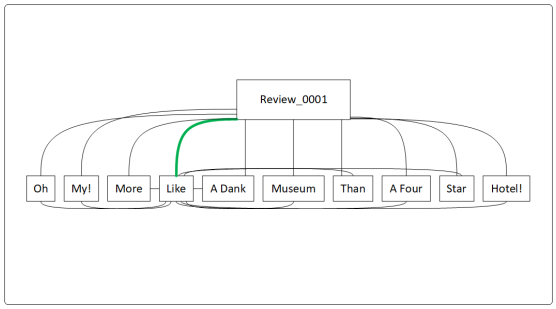


Fig. 2. lda2vec Process

document vectors ("Review_0001"), thus capturing global and local correlations to yield more coherent topics.

As discussed above, sparse vectors are more convenient to reason about, however we have found a way to adapt the BERT language model to topic modeling. The idea was inspired by Top2Vec [49] where easily interpretable topics are created by jointly using document and word embeddings. We have also taken some ideas from Maarten Grootendorst's blog post[1], where he used BERT and hugging face transformer embeddings to generate topics. In our approach the procedure consists of the following steps:

1) Generate BERT embeddings for a document
2) Reduce dimensionality
3) Apply a clustering algorithm
4) Aggregate documents by cluster
5) Apply the class-based variant of TF-IDF to extract the terms-per-topic matrix

This allows to have the same output as with LDA, but with deeper sense of language context by learning contextual relations between words and subwords in a text.

In our model the overall algorithm remains the same whereas the way the topics are generated varies depending on the approach. Each approach has its specific features and for a better understanding of the nature of the deception we apply them all on the same corpus and evaluate their performance. A brief summary of all the approaches applied in our model is given in Table I.

| Approach | Text representation | Key points |
|---|---|---|
| LDA | Sparse simplex vectors | Utilize the pure distribution of words to mathematically model topics |
| LDA with hypernyms | Sparse simplex vectors | Integrated a lexical database into LDA |
| LDA2VEC | Mixture of sparse and dense vectors | Extend LDA by adding semantic relations between the learned vector representations |
| Topic modeling with BERT | Dense distributed vectors | Learn contextual relations between words in a text to construct topics |

TABLE I
TOPIC MODELING VARIATIONS FOR DECEPTIVE OPINION SPAM

## IV. EXPERIMENTS

*Data*

One of the first large-scale, publicly available datasets for the research in this domain is Ott Deceptive Opinion Spam corpus [1] composed of 400 truthful and 400 gold-standard deceptive reviews. To solicit these high-quality deceptive reviews using Amazon Mechanical Turk, a pool of 400 Human-Intelligence Tasks (HITs) has been created. These HITs have been then allocated across 20 chosen hotels. They have also ensured that opinions are written by unique reviewers, by allowing only a single submission per Turker. For truthful opinions they mined 6977 reviews from the 20 most popular Chicago hotels on Trip Advisor. With their dataset the authors have shown that the detection of deception is challenging for human judges, as most of them performed roughly at-chance. As the Ott corpus is one of the most commonly used corpora to perform deception analysis, we used this dataset as a benchmark for our research.

*LDA Topic Modeling*

Preprocessing is an important part of any NLP task. In our experiment it is even more crucial, as it may impact certain patterns of the text. Thus we tested with different preprocessing techniques especially focusing on such aspects as stop words, lemmatization, numbers and punctuation. According to our observations the noise produced by stop words prevents our model from recognizing patterns in content-bearing words. Although some researches demonstrate that a parsimonious asymmetric Dirichlet prior inferred for terms-per-topic distribution allows model inference to isolate stop words into small low-quality topics without affecting the rest of the clusters [50], abnormally frequent terms (like 'a' or 'the') are likely to be prominent in all the topics. Consequently, to avoid generating meaningless patterns we removed the stop words from our corpus. The list of stop words used in our experimentation is available in Azure Storage[2]. We also apply lemmatization to replace the words by the lemmas and therefore reduce the size of our dictionary. Besides being useful from the computational point of view, techniques like lemmatization and stemming allow removing semantic duplicates. However, stemming may also add noisy data to the dictionary, as most of the existing stemmers include the tokens that are not real words, whereas lemmatization generally gives more coherent results leaving only topic-related terms. As LDA has no actual semantic knowledge of the words, steps like normalizing case to lowercase, normalizing whitespaces and removing punctuation are important, as they allow to construct a more relevant dictionary. We have also noticed during the experimentation that without removing digits the clusters are clearly identifiable but do not represent the deception and truth.

[1]https://towardsdatascience.com/topic-modeling-with-bert-779f7db187e6

[2]https://az754797.vo.msecnd.net/docs/Stopwords.zip

| Max dictionary Size | Rho | Alpha | Batch Size | Power | N-grams | Correct Clusters |
|---|---|---|---|---|---|---|
| 2 000 000 | 0.01 | 0.01 | 32 | 0.5 | 5 | 60.5% |
| 20 000 | 0.01 | 0.01 | 32 | 0.5 | 10 | 50% |
| 20 000 | 0.01 | 0.01 | 32 | 0.5 | 5 | 52% |
| 2 000 000 | 0.01 | 0.01 | 32 | 0.5 | 10 | 61.5% |
| 2 000 000 | 0.01 | 0.01 | 32 | 0.1 | 10 | 50% |
| 2 000 000 | 0.01 | 0.01 | 32 | 0.7 | 10 | 68% |
| 2 000 000 | 0.01 | 0.01 | 32 | 1 | 10 | 70% |
| 2 000 000 | 1 | 1 | 32 | 1 | 10 | 50% |
| 2 000 000 | 0.1 | 0.1 | 32 | 1 | 10 | 52% |
| 2 000 000 | 0.001 | 0.001 | 32 | 1 | 10 | 62% |
| 2 000 000 | 0.07 | 0.07 | 32 | 1 | 10 | 64% |
| 2 000 000 | 0.01 | 0.01 | 16 | 1 | 10 | 67% |
| 2 000 000 | 0.01 | 0.01 | 64 | 1 | 10 | 74.5% |
| 2 000 000 | 0.01 | 0.01 | 128 | 1 | 10 | 53.5% |
| 2 000 000 | 0.01 | 0.01 | 96 | 1 | 10 | 52% |
| 2 000 000 | 0.01 | 0.01 | 64 | 1 | 9 | **75%** (599/800) |
| 2 000 000 | 0.01 | 0.01 | 64 | 1 | 8 | **75%** (603/800) |
| 2 000 000 | 0.01 | 0.01 | 96 | 1 | 7 | 72% |

TABLE II
RESULTS OF LDA EXPERIMENTATION: NEGATIVE POLARITY

It is important to mention that removing digits in our setup implies removing numbers and all the digits from the strings. For instance, a string like $3432PoplarLaneG7$ will result in $PoplarLaneG$ after the preprocessing. With all this in mind, we have defined the following set of preprocessing steps:

- Remove stop words
- Apply lemmatization
- Normalize case to lowercase
- Normalize whitespaces
- Remove punctuation
- Remove digits

Removing special characters, email addresses, URLs and expanding verb contractions results in poor results and all the reviews are assigned to one topic. Additionally we applied POS tagging and tried removing different POS one by one but, according to our observations, removing POS may result in wrong clusters. Surprisingly, removing adjectives results in worse results than removing nous, which implies that adjectives are more important for deception detection. Moreover, we filter reviews by their polarity and apply the model on positive and negative texts separately, as sentiment may also represent a logical cluster.

The parameters we fine tuned for the model are:

- **Max size of ngram dictionary** : total number of rows in the n-gram dictionary.
- **Rho parameter** : prior probability for the sparsity of topic distributions.
- **Alpha parameter** : prior probability for the sparsity of topic weights per document.
- **Size of the batch** : number of rows processed in chunks.
- **Initial value of iteration used in learning update schedule** : learning rate start value, set to 0 in all the experiments.
- **Power applied to the iteration during updates** : learning stepsize.
- **N-grams** : maximum size of the sequences generated during hashing.
- **Number of training iterations** : Maximum number of repetitions the algorithm cycles over the data, set to 1024 in all the experiments.

From the results present in Table II, we can conclude that lowest Rho value results in the best results, which means that in deceptive reviews most words appear sparsely (not equiprobable). Additionally, we observe that the lower the Alpha parameter, the better deception detection works. In other words, prior probability for the sparsity of per-document topic weights is low and most of the weight in the topic distribution of a review goes to a single topic, which supports our hypothesis that there are two global topics in the corpus (deceptive and truthful) and we can detect them by controlling the sparsity of topic distributions. At the same time, the highest results are achieved with the largest vocabulary size, which demonstrates the linguistic diversity of the corpus. Not surprisingly, the number of n-grams turns out to be the most important parameter, as any slight modification of the it's value dramatically impacts the overall performance. The best results correspond to the n-grams equal to 8-9 terms, that is to say the deception cannot be expressed in one or two key words. With this in mind, we applied the same algorithm with the best configuration on the positive reviews and achieved the overall score 69% (553 out of 800), which means that the majority of the deceptive clues are preserved even when the polarity changes.

*LDA Topic Modeling with hypernym*

In this part we define hypernymy as a relation between synsets, as is done in WordNet. We map noun phrases to a node in the taxonomy (Wordnet) and replace each noun phrase with the corresponding hypernym. We then run LDA with the best configuration from Experiment 1 on this modified corpus and follow the same procedure of topic assignment. Due to the specific procedure of hypernym generation we modified the text processing. For instance, as the hypernym generation function already included lemmatization, we removed this

| Model | n_neighbors | n_components | UMAP metric | min_cluster_size | HDBSCAN metric | Result |
|---|---|---|---|---|---|---|
| paraphrase-MiniLM-L6-v2 | 10 | 10 | cosine | 50 | p | 54.38% (435/800) |
| stsb-mpnet-base-v2 | 15 | 5 | cosine | 15 | euclidean | 54.00% (432/800) |
| paraphrase-mpnet-base-v2 | 10 | 10 | canberra | 50 | euclidean | 56.63% (453/800) |
| stsb-mpnet-base-v2 | 5 | 10 | cosine | 50 | euclidean | 57.75% (462/800) |
| paraphrase-mpnet-base-v2 | 5 | 10 | cosine | 50 | euclidean | 57.75% (462/800) |
| paraphrase-mpnet-base-v2 | 10 | 15 | cosine | 50 | euclidean | 60.63% (485/800) |
| paraphrase-mpnet-base-v2 | 10 | 10 | cosine | 55 | l2 | 60.88% (487/800) |
| paraphrase-mpnet-base-v2 | 10 | 10 | cosine | 55 | chebyshev | 60.63% (485/800) |
| paraphrase-mpnet-base-v2 | 10 | 10 | minkowski | 50 | euclidean | 62.25% (498/800) |
| paraphrase-mpnet-base-v2 | 10 | 10 | cosine | 50 | euclidean | 62.25% (498/800) |
| paraphrase-mpnet-base-v2 | 10 | 10 | cosine | 55 | p | 63.00% (504/800) |
| paraphrase-mpnet-base-v2 | 10 | 10 | cosine | 50 | p | **63.00%** (507/800) |

TABLE III
RESULTS OF BERT EXPERIMENTATION: NEGATIVE POLARITY

step. The final list of pre-processing steps we had in our experiment was:

- Normalize case to lowercase
- Normalize whitespaces
- Remove punctuation
- Remove digits

with the following LDA parameters:

- **Max dictionary Size** : 2 000 000
- **Rho** : 0.01
- **Alpha** : 0.01
- **Batch Size** : 64
- **Power** : 1
- **N-grams** : 8

With this configuration we obtained the overall score of **59%** (470 out of 800), which is much lower than the previous experiment. Moreover we observe that the topics are poorly distributed across the documents, as LDA model classified 91% of all the analyzed documents as deceptive. This implies that deceptive cues are expressed at lower levels, as we probably loose some rich statistical information when replacing the noun phrases with their hypernyms.

*lda2vec Topic Modeling*

For this experiment we applied the Pytorch implementation of Moody's lda2vec proposed by Antoshchenko[3]. To set the experiment up and running we adjusted the approach by setting the number of generated topics to 2. We initialized the topic assignments for each document using the basic LDA implementation (sometimes referred to as "vanilla"), to slightly improve the topic distribution. We then converted each negative review in the initial corpus to a set of tuples (id, word, window) and maximized the objective function for each of those tuples:

$$\mathcal{L} = \mathcal{L}^{dirichlet} + \sum_i \mathcal{L}_i^{neg},$$

$$\mathcal{L}_i^{neg} = \log \sigma(\vec{c} \cdot \vec{w}) + \sum_k \log \sigma(-\vec{c} \cdot \vec{w}),$$

$$\mathcal{L}^{dirichlet} = \lambda \sum_j (\alpha - 1) \log p_j, \alpha < 1$$

[3]https://github.com/TropComplique/lda2vec-pytorch

$$\vec{c} = \vec{w} + \sum_j p_j \cdot \vec{t}_j, \sum_j p_j = 1, p_j >= 0$$

where $\vec{c}$ is the context vector, $\vec{w}$ corresponds to the embedding vector of a word, $\vec{t}$ is the topic vector, $\lambda$ is the positive constant allowing us to adjust the sparsity, $k$ corresponds to the sum over sampled negative words, $j$ is the sum over topics and $p$ is the probability distribution over topics for a document. Moreover, as suggested by Antoshchenko we added noise to some gradients while training, reweighted loss according to document lengths and trained 50-dimensional skip-gram word2vec before running lda2vec. This resulted in 418 negative documents, correctly recognized as deceptive or truthful, which corresponds to 52,3%. As we can see, the results are lower than in any of the previous experiments, which may be probably due to the fact that the algorithm is prone to local minima, and greatly depends on values of initial topic assignments.

*Topic Modeling with BERT*

For this experiment we have used the SentenceTransformers framework [51] which offers a large collection of pre-trained models adapted for various tasks. To reduce the dimensionality we have applied the Uniform Manifold Approximation and Projection (UMAP) algorithm [52] as it allows to preserve the high-dimensional local structure in lower dimensional space. For clustering we have implemented the Hierarchical density based clustering (HDBSCAN) [53], which transform the vector space according to the density/sparsity, construct cluster hierarchy of connected components and extracts the stable clusters from the condensed tree. The parameters we fine tuned for this experiment are:

- BERT
  - **Pre-trained model**: one of the fine-tuned language models compared on different benchmarks, like Twitter Paraphrases or Duplicate Questions
- UMAP
  - **Number of neighbors** : allows to balance between local and global structure in the initial corpus
  - **Number of components** : determines the dimensionality of the target vector
  - **Metric** : controls how distance is computed in the ambient space

- HDBSCAN
  - **Min cluster size** : smallest grouping we wish to consider a cluster; this parameter allows us to control the number of generated topics
  - **Metric** : controls how distance is computed between document vectors
  - **Cluster selection method** : controls how to select flat clusters from the cluster tree hierarchy (set to eom in all the experiments, as leaf results in empty clusters)

From the results present in Table III, we can conclude that the model performs the best when the cluster distance metric is set p, which corresponds to Minkowski distance. This metric is the generalization of both the Euclidean distance and Manhattan distance for clusters, which means that the distances between document vectors in Ott corpus are non-negative, symmetric and the distance of each document to itself is equal to 0. It also implies that all the distances follow the triangular inequality rule, i.e. the distance between document A and document B is less or equal than the sum of distances between the documents A,C and C,B. We can also observe, that cosine metric for dimensionality reduction gives the highest precision, which proves the angular and correlation relationship between deceptive reviews. There is also a striking difference between paraphrase-mpnet-base-v2 and other models like paraphrase-MiniLM-L6-v2 or stsb-mpnet-base-v2. This is probably due to the fact that MPNet combines strengths of masked and permuted language modeling for better language understanding.

## V. Conclusion and Future Work

In this work we utilized the topic modeling approach to detect the common patterns underlying the deceptive reviews. For better understanding we applied different techniques, varying from classical LDA to embedding-based extensions. As we have seen in the previous section, classical LDA-based approach performs better than the other approaches, and, surprisingly, even better than the approaches based on dense vectors, like lda2vec or BERT-based topic modeling. Whereas the hypernym-based extension of LDA performs worse due to the important information loss, the reason of dense vector's poor performance is still not completely clear. One of the possible reasons may be the fact that in our LDA model we defined larger n-grams (8), whereas with BERT topic modeling we treat only individual tokens, which doesn't allow us to capture meaningful patterns. Even if the distributed words representations can capture semantically meaningful regularities between words, they are strongly dependent on the training set, which is not the case with LDA. In the future works we plan to refine the topic modeling approach by analysing the documents part by part, and trying to find meaningful patterns in them. Furthermore, we are going to keep experimenting with ontologies and external knowledge sources. Another idea is to identify text genre since some language usages embedded in grammatical constructions and word senses are related to the style of the text. Moreover, it is possible that not every part of the text in dataset is informative, which makes it sensible to keep splitting the reviews into multiple heterogeneous pieces. As genre identification allows to get more heterogeneous texts this may beneficial for finding more precise patterns.

## References


[1] Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T Hancock. Finding deceptive opinion spam by any stretch of the imagination. *arXiv preprint arXiv:1107.4557*, 2011.

[2] Nikolai Vogler and Lisa Pearl. Using linguistically defined specific details to detect deception across domains. *Nat. Lang. Eng.*, 26(3):349–373, 2020.

[3] Dan Barsever, Sameer Singh, and Emre Neftci. Building a better lie detector with bert: The difference between truth and lies. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2020.

[4] Nitin Jindal and Bing Liu. Opinion spam and analysis. In *Proceedings of the 2008 international conference on web search and data mining*, pages 219–230, 2008.

[5] Timothy R Levine and Charles F Bond. Direct and indirect measures of lie detection tell the same story: A reply to ten brinke, stimson, and carney (2014). *Psychological science*, 25(10):1960–1961, 2014.

[6] Peter F Brown, Vincent J Della Pietra, Peter V Desouza, Jennifer C Lai, and Robert L Mercer. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–480, 1992.

[7] Alex Marin, Roman Holenstein, Ruhi Sarikaya, and Mari Ostendorf. Learning phrase patterns for text classification using a knowledge graph and unlabeled data. In *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[8] James W Pennebaker, Martha E Francis, and Roger J Booth. Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates*, 71(2001):2001, 2001.

[9] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of human language technology conference and conference on empirical methods in natural language processing*, pages 347–354, 2005.

[10] Alex Marin, Mari Ostendorf, Bin Zhang, Jonathan T Morgan, Meghan Oxley, Mark Zachry, and Emily M Bender. Detecting authority bids in online discussions. In *2010 IEEE Spoken Language Technology Workshop*, pages 49–54. IEEE, 2010.

[11] Alex Marin, Bin Zhang, and Mari Ostendorf. Detecting forum authority claims in online discussions. In *Proceedings of the Workshop on Language in Social Media (LSM 2011)*, pages 39–47, 2011.

[12] Caroline Brun and Caroline Hagege. Suggestion mining: Detecting suggestions for improvement in users' comments. *Research in Computing Science*, 70(79.7179):5379–62, 2013.

[13] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Cambridge university press, 2008.

[14] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.

[15] Stefanie Tellex, Boris Katz, Jimmy Lin, Aaron Fernandes, and Gregory Marton. Quantitative evaluation of passage retrieval algorithms for question answering. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 41–47, 2003.

[16] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics, 2010.

[17] Richard Socher, John Bauer, Christopher D Manning, and Andrew Y Ng. Parsing with compositional vector grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 455–465, 2013.

[18] Martin Potthast, Johannes Kiesel, Kevin Reinartz, Janek Bevendorff, and Benno Stein. A stylometric inquiry into hyperpartisan and fake news. *arXiv preprint arXiv:1702.05638*, 2017.

[19] Daochen Zha and Chenliang Li. Multi-label dataless text classification with topic modeling. *Knowledge and Information Systems*, 61(1):137–160, 2019.

[20] Orestis Papakyriakopoulos, Simon Hegelich, Juan Carlos Medina Serrano, and Fabienne Marco. Bias in word embeddings. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 446–457, 2020.

[21] Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

[22] Christopher E Moody. Mixing dirichlet topic models and word embeddings to make lda2vec. *arXiv preprint arXiv:1605.02019*, 2016.

[23] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[24] Laurent Charlin, Rajesh Ranganath, James McInerney, and David M Blei. Dynamic poisson factorization. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 155–162, 2015.

[25] Julian McAuley, Rahul Pandey, and Jure Leskovec. Inferring networks of substitutable and complementary products. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 785–794, 2015.

[26] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv:1310.4546*, 2013.

[27] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[28] François Mairesse, Marilyn A Walker, Matthias R Mehl, and Roger K Moore. Using linguistic cues for the automatic recognition of personality in conversation and text. *Journal of artificial intelligence research*, 30:457–500, 2007.

[29] Whitney L Cade, Blair A Lehman, and Andrew Olney. An exploration of off topic conversation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 669–672, 2010.

[30] Katarina R Krüger, Anna Lukowiak, Jonathan Sonntag, Saskia Warzecha, and Manfred Stede. Classifying news versus opinions in newspapers: Linguistic features for domain independence. *Natural Language Engineering*, 23(5):687, 2017.

[31] Myle Ott, Claire Cardie, and Jeffrey T Hancock. Negative deceptive opinion spam. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: human language technologies*, pages 497–501, 2013.

[32] Jiwei Li, Myle Ott, Claire Cardie, and Eduard Hovy. Towards a general rule for identifying deceptive opinion spam. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1566–1576, 2014.

[33] Jacob Eisenstein, Amr Ahmed, and Eric P Xing. Sparse additive generative models of text. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 1041–1048. Citeseer, 2011.

[34] Yafeng Ren and Donghong Ji. Neural networks for deceptive opinion spam detection: An empirical study. *Information Sciences*, 385:213–224, 2017.

[35] Cicero Dos Santos and Maira Gatti. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 69–78, 2014.

[36] Yafeng Ren, Yue Zhang, Meishan Zhang, and Donghong Ji. Context-sensitive twitter sentiment classification using neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

[37] Judee K Burgoon, David B Buller, Laura K Guerrero, Walid A Afifi, and Clyde M Feldman. Interpersonal deception: Xii. information management dimensions underlying deceptive and truthful messages. *Communications Monographs*, 63(1):50–69, 1996.

[38] Steven A McCornack. Information manipulation theory. *Communications Monographs*, 59(1):1–16, 1992.

[39] Marcia K Johnson and Carol L Raye. Reality monitoring. *Psychological review*, 88(1):67, 1981.

[40] M Steller and G Koehnken. Criteria-based statement

analysis. credibility assessment of children's testimonies in sexual abuse cases. *Psychological techniques in law enforcement*, pages 217–245, 1989.

[41] Bennett Kleinberg, Maximilian Mozes, Arnoud Arntz, and Bruno Verschuere. Using named entities for computer-automated verbal deception detection. *Journal of forensic sciences*, 63(3):714–723, 2018.

[42] Rada Mihalcea and Carlo Strapparava. The lie detector: Explorations in the automatic recognition of deceptive language. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 309–312, 2009.

[43] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[44] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[45] Hyunjin Choi, Judong Kim, Seongho Joe, and Youngjune Gwon. Evaluation of bert and albert sentence embedding performance on downstream nlp tasks. *arXiv preprint arXiv:2101.10642*, 2021.

[46] Justin Wood, Patrick Tan, Wei Wang, and Corey Arnold. Source-lda: Enhancing probabilistic topic models using prior knowledge sources. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, pages 411–422. IEEE, 2017.

[47] Sam Scott and Stan Matwin. Text classification using wordnet hypernyms. In *Usage of WordNet in Natural Language Processing Systems*, 1998.

[48] Alan Ritter, Stephen Soderland, and Oren Etzioni. What is this, anyway: Automatic hypernym discovery. In *AAAI Spring Symposium: Learning by Reading and Learning to Read*, pages 88–93, 2009.

[49] Dimo Angelov. Top2vec: Distributed representations of topics. *arXiv preprint arXiv:2008.09470*, 2020.

[50] Hanna M Wallach, David M Mimno, and Andrew McCallum. Rethinking lda: Why priors matter. In *Advances in neural information processing systems*, pages 1973–1981, 2009.

[51] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[52] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[53] Md Farhadur Rahman, Weimo Liu, Saad Bin Suhaim, Saravanan Thirumuruganathan, Nan Zhang, and Gautam Das. Hdbscan: Density based clustering over location based services. *arXiv preprint arXiv:1602.03730*, 2016.