SUPPLEMENTARY MATERIALS

Exploiting Natural Language Services: A Polarity Based Black-box Attack

Fatma GUMUS[1,2], **M. Fatih AMASYALI**[1]

1 Department of Computer Engineering, Yildiz Technical University, Istanbul 34220, Turkey
2 Department of Computer Engineering, Air Force Academy, National Defence University, Istanbul 34149, Turkey

Many decision making, planning and pattern analyzing processes involve natural language services, such that their output affect lives and revenues. They need to be reliable and precise even in the very probable presence of malicious input. However such services, particularly classifiers, are vulnerable to perturbation attacks. Most of the literature that focuses on these types of attacks under black-box setting, either use an oracle to transfer the samples generated, or abuse the unknown vocabulary bias. In this paper, we craft adversarial samples by word-substitutions without an oracle. Our threat model assigns polarity scores for terms, using only an external test set. Then it queries the black-box and corrupts a separate set of samples to fool the classifier. The targeted and untargeted attacks to binary and multi-class models are able to significantly increase confidence in the attack direction. In binary models maximum average misclassification confidence goes up by 53%, while in multi-class models it is 45%. After analyzing the attack on naive Bayes and BiLSTM models, we show the applicability of the attack on a true black-box model: IBM Watson Natural Language Understanding. We also demonstrated that the polarity attack is undetectable by simple semantic similarity measure using ELMo embeddings. Experimental results indicate that the semantic cosine similarity is retained. Further analysis of detectability is conducted by human evaluations on the true black-box adversarial review text, which revealed that the attack is mostly unnoticeable.

E-mail: *fa.gumus@gmail.com*

Papernot et al. [1] suggested a taxonomy for adversarial attacks based on adversary knowledge and the goal. By this taxonomy, the adversary we propose aims to reduce classification confidence in complete black-box setting where there is no knowledge on the model except for its response to the adversary's queries. Specifically our adversary performs discriminative and non-discriminative exploratory attacks, in order to reduce classification confidence in black-box settings.

Our method focuses on targeted and untargeted attacks to black-box models, where word perturbations are at play. Our polarity based algorithm only uses the class label and confidence score information returned from the black-box service. We propose using an external dataset to calculate scores for each word between one-vs-one and one-vs-rest polarities. We conduct experiments on several datasets to analyze how well our scheme works on different classification domains. We also conduct a true black-box experiment on IBM's Watson NLU service to further confirm its applicability on real case scenarios. In addition to our polarity based adversarial attack, we introduce two baseline attack methods: random attack and hot terms swap. They both simulate the idea behind character perturbation attacks by swapping chosen tokens with an unknown token. Random attack chooses the tokens randomly, while hot terms attack uses an attack direction specific universal list of important tokens.

We present this material to help clarify the threat model in the main document as well as to provide the results of additional experiments. The layout of this document is as follows: We clarify our proposed threat model in Section 1, in Section 2 we introduce the base-

line models we used to assess capability of our model, Section 3 explains the additional experiments we performed, Section 4 presents the comparative results between main and additional experiments, and finally we state an extended discussion in Section 5.

1 Threat Model

The adversary has three dimensions: its goal, knowledge and capability, and strategy.

Adversarial Goal. The adversary ultimately aims at a higher rate of misclassification, compromising the system's integrity. To attain its goal, the adversary iteratively increases misclassification probabilities by probing of the black-box.

Adversary Knowledge and Capability. We address two assumptions; minimum knowledge and the degree of resourcefulness.

- **Assumption 1.** The adversary recognizes the task; where the task is the problem (e.g., sentiment analysis) and the set of classes (e.g., positive/negative or positive/neutral/negative).
- **Assumption 2.** The adversary can collect a virtually unlimited number of unlabeled samples relevant to the task from an external source (e.g., web crawling) at no cost.

Given the two assumptions, the adversary has two settings of knowledge: perfect and limited knowledge. The adversary has *perfect knowledge* if the black-box system returns probability scores for each class given an input sample, and it has limited knowledge if the output is only the resulting class label. We only focus on the perfect knowledge setting in this paper.

Adversary Strategy. The samples can be crafted to have a few big changes, or they can have many small changes. Alternatively, hybrid strategies can be employed. We define four strategies in the Experimental Design section.

The threat model definition allows us to outline the detailed context of the environment the adversary performs under. We mainly disregard the possible limitations such as query budget and availability of confidence scores in this paper.

Our aim is to perform attack on black-box systems with as little information on the model as possible. We essentially corrupt a legitimate sample to fool the model. Polarity scores are calculated for each word and each source-target class tuple. To coordinate the substitutions, the adversary fills in the polarity table using the samples collected by the Assumptions 1 and 2. The attack intuition is that when converting a sample of one class to another, terms located at a high positive scale on the target class would be swapped with the terms located at a low negative scale on destination. The same term can have strong belongingness for different class couples. This is akin to probabilistic topic modeling techniques [2] where multiple terms can be assigned to more than one topic cluster. On the very high level, polarity based adversary strategy is described as follows:

1. Fill in a polarity table for the terms, in which polarity scores and part of speech (PoS) tags are present for the tokens attained from the collected samples.
2. For any given clean sample determine the replaceable tokens based on a PoS filter. Put these tokens in a swap queue at random order.
3. Look up in the polarity table, perturb tokens of the same PoS by swapping within a distortion range α.
4. Check if the black-box model returns the minimum desired confidence θ for the target class given the crafted sample. Discard the last changes if the objective probability score does not improve.
5. Stop perturbation if a maximum number of changes δ is made, or a maximum number of iterations is reached.

2 Baselines

We introduce two baselines to compare against our polarity based adversarial crafting method. They both simulate the effect of character perturbation models: changing a word into an unknown token. Instead of reimplementing character perturbation attacks, we directly swap the target word with our unknown token $< unk >$.

The random baseline selects δ random words from the sample and swaps them with $< unk >$ token, until a probability score of θ is reached for the targeted class or all the selected words are swapped.

The second baseline is a naive selection of the words based on their popularity on the original class distribution. We use sorted polarity table in selecting top T words based on target and destination classes. We call this hot terms baseline. For instance, in Figure 1, for target class "World" and destination class "Sports" on AG's News dataset, we use the sorted "World to Sports" polarity scores for $T = 10$. They are the important words for their respective classes when we look at it as a binary problem: World vs Sports. In the attack described in Algorithm 1, we would swap some source tokens with some target tokens, in which selection is performed according to the constraints of the

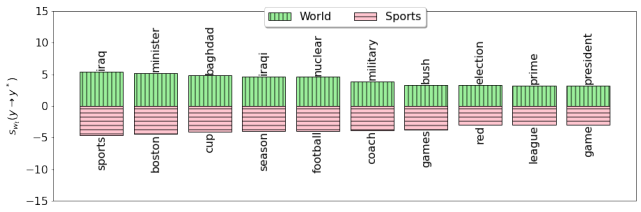

Fig. 1 Hot terms for T=10 when source:World & target: Sports.

algorithm. In hot term baseline, however, we craft the adversarial samples by swapping up to δ randomly selected hot terms with $< unk >$ token.

3 Experiment Settings and Additional Experiment Strategy

3.1 Datasets and Tasks

We use six classification datasets of two tasks: sentiment classification and categorization. Table 1 shows the properties of the datasets. 100 samples of each test set is held-out to be corrupted, and the rest of them is used to fill in the polarity tables. We train the NB and BiLSTM models with the train sets using 2^{16} features in all our models. We performed the following preprocessing steps: conversion to lower case, masking the numbers, fixing the common contractions such that "I've" is changed into "I have".

3.2 Classifiers

We use three classifiers in our experiments: Naive Bayes classifier (NB), BiLSTM Classifier and IBM Watson NLU Sentiment Classifier. NB and BiLSTM classifiers are word-based, where tokens are individual words. IBM Watson Sentiment Classifier is a complete black-box for us, we neither know of its architecture nor do we have any information on the data it is trained on. It is a web service, where we are provided with a class label and the probability score.

Both NB and BiLSTM classifiers have the vocabulary size of 2^{16}. NB classifier uses mono-gram TD-IDF features, whereas BiLSTM learns word embeddings from the training data. We optimized our BiLSTM classifier through all datasets and come up with hyper-parameter values that work adequately on all of them. Networks' embedding size is 64, sequence length is 200, batch size 128, 128 units on a single BiLSTM layer. We regularized the model with 0.5 dropout and early stop at training. The accuracy scores on corresponding datasets are also shown in the Table 1.

https://www.ibm.com/cloud/watson-natural-language-understanding

3.3 Experiment Strategy

Our adversary can attack using a number of strategies based on distortion range (α) and maximum number of swaps allowed (δ). *Unbounded attack* virtually has no restrictions with $\alpha = 5.0$ and $\delta = 100$. It searches within maximum total distortion to achieve $\theta = 0.51$. *Confined attack* aims to construct the adversarial samples by minimum number of swaps and term distortion range ($\alpha = 1.0$, $\delta = 5$). We name the attack group that is neither unbounded nor confined as controlled attacks where special conditions applied; more specifically, *controlled distortion* ($\alpha = 1.0$, $\delta = 100$) and *controlled number of swaps* ($\alpha = 5.0$, $\delta = 5$).

Let $V = \{v_1, v_2, ..., v_N\}$ the held-out clean sample set and $V^* = \{v_1^*, v_2^*, ..., v_N^*\}$ the corresponding corrupt samples. While evaluating the success of the attack, we use the following measurements:

1. Mean difference in confidence as in the Equation 1.
2. Average number of word replacements.
3. Difference in accuracy of the model given V and V^*.

$$\Delta Confidence = \frac{1}{N}\sum_{i=1}^{N} Conf(y_i^*|v_i^*) - Conf(y_i^*|v_i) \quad (1)$$

4 Comparative Results

When a class attains 0.51 confidence for a sample, the result label would be of that class regardless of the number of classes. Therefore, for both binary sentiment and multi-class categorization problems we set $\theta = 0.51$ throughout all the experiments. Note that, in our algorithm we can set θ to a higher confidence value to increase the adversarial strength.

For the multi-class models, targets for the targeted attacks are selected by utilizing the confusion matrices on the whole clean test set. Wherever most confusion among two classes is, we regard it as the most susceptible target. In the consequent subsections, we present comparative results with the main and additional experiments.

Table 1 Dataset and task properties

Properties	Yelp [3]	Amazon [3]	IMDB [4]	DB Pedia [3]	AG's News [3]
Task	Sentiment	Sentiment	Sentiment	Categorization	Categorization
#Class	2	2	2	14	4
#Train	560K	3.6M	25K	560K	120K
#Test	38K	400K	25K	70K	7600
Voc. Size	234K	869K	75K	649K	67K
Avg. Seq. Len.	158	88	275	53	38
Classifiers	**Original Test Accuracy**				
Naive Bayes	0.84	0.79	0.80	0.95	0.88
BiLSTM	0.96	0.94	0.79	0.99	0.91
IBM Watson NLU Classifier	0.93	0.76	0.85	x	x

4.1 Polarity Attacks on Experimental Classifiers

For each respective victim model, the 100 held-out clean samples are corrupted. Sentiment models (Figure 2) and multi-class models (Figure 3) have their respective plots under the strategies, and the two classification techniques (NB and BiLSTM) are denoted. Each annotated point represents an attack configuration, where victim models are shown in the plot legends.

We use test sets of the models to calculate the polarity scores, e.g. the models trained with AG's News train set (the victim model), and polarity table built from AG's News test set is used together in order to corrupt 100 left-out AG's News test samples.

In addition to the average δ and $\Delta Confidence$, we compare targeted and untargeted attacks for multi-class victims. Another point of consideration is the polarity source. In a black-box setting, the adversary does not have any information on the training data. Therefore, it is reasonable to expect that the data collected by Assumption 2 will not be of the same distribution as the data the victim model is trained with. Considering the sentiment tasks are all binary classifications of "positive" and "negative" labels, we utilize the use of *cross-domain polarity tables*. For instance, the models trained with Amazon train set is coupled with polarity tables from Yelp, Amazon and IMDB test sets individually to corrupt 100 left-out Amazon test samples.

Unbounded Attack: Figure 2 plot S1 and Figure 3 plot M1 present unbounded attacks to the victim models. Despite the high limit for δ (δ=100), all the attacks for sentiment models were completed with less than 7 swaps per sample at average, whereas multi-class samples needed more replacements. BiLSTM models were fooled with distinctively higher $\Delta Confidence$ than their NB counterparts.

When we evaluated the cross-domain polarity, Amazon test set was much better polarity source for composing less noisy attacks. It is impressive how a few swaps are needed to change a sample's class with Amazon polarity table; but not surprising, given the huge resource the table is built on. Sample size of the polarity source is a determiner on how accurately the scores represent the problem domain.

The attacks were more confident when they were untargeted for multi-class victims. As we expected, more swaps were needed for the DBPedia models, where there are far more classes than with AG's News models (14 and 4). It was more costly to perform attacks as the number of classes grew. In future work, we will look into whether the number of classes is conclusively important to determine a model to be "more susceptible" to polarity based attacks.

Controlled Distortion: In this set of experiments (Figure 2.S2 and Figure 3.M2), we assess the adversarial samples crafted with "many small changes". Similar to unbounded attacks, BiLSTM models are fooled with way higher confidence. As the changes got less noisy, more number of swaps were performed.

For sentiment victims, the cross-polarity gaps were closer than they were with the unbounded attacks. Except for Yelp polarity, where the adversary generally needed more number of swaps. We observed that Amazon polarity table is a good resource to craft the adversarial samples this strategy as well.

There was a significant drop in $\Delta confidence$ levels with the multi-class models. Similar to the unbounded results, AG's News victims were attacked with more confidence when the attack was untargeted, whereas for DBPedia victims it was the opposite.

Controlled Number of Swaps: Figure 2.S3 shows that NB can be fooled with less number of

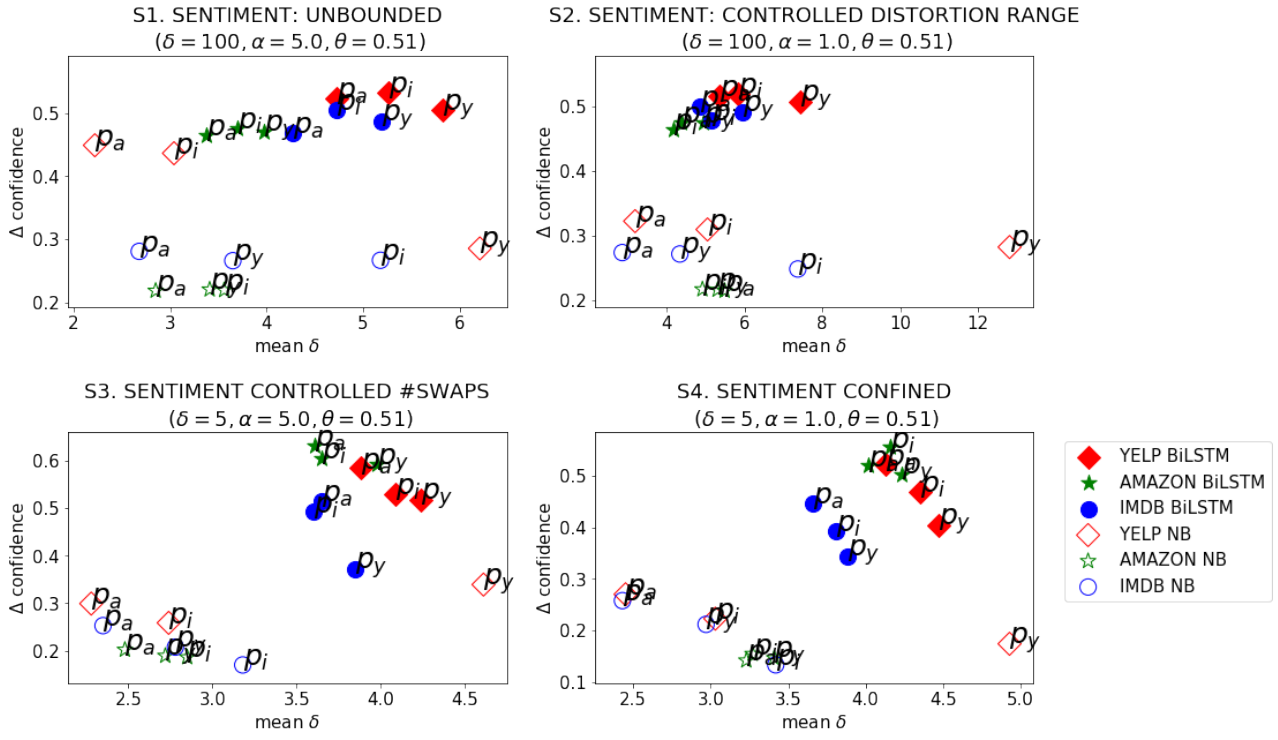

Fig. 2 Polarity Attacks to Sentiment Models.p_a: polarity attack that utilizes polarity table of AMAZON test set, p_i: polarity attack that utilizes polarity table of IMDB test set, p_y: polarity attack that utilizes polarity table of YELP test set.

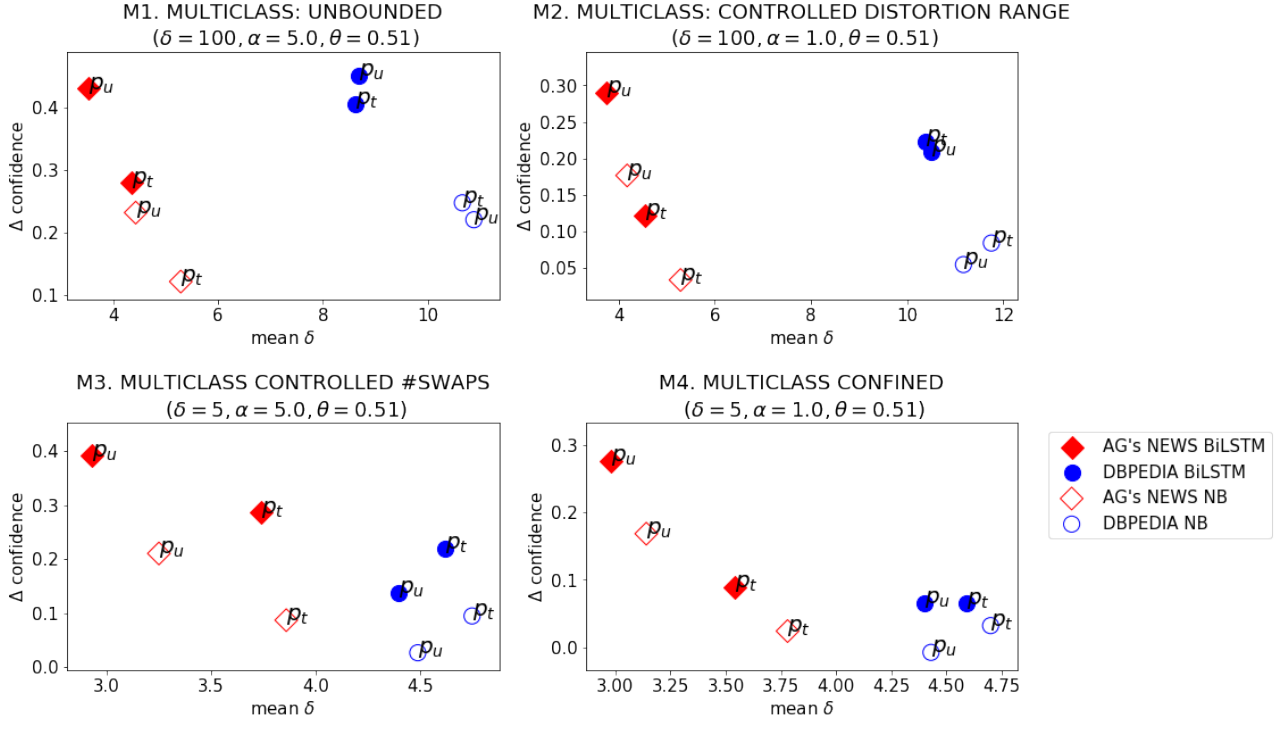

Fig. 3 Polarity Attacks to Categorization Models. p_t: targeted polarity attack, p_u: untargetted polarity attack.

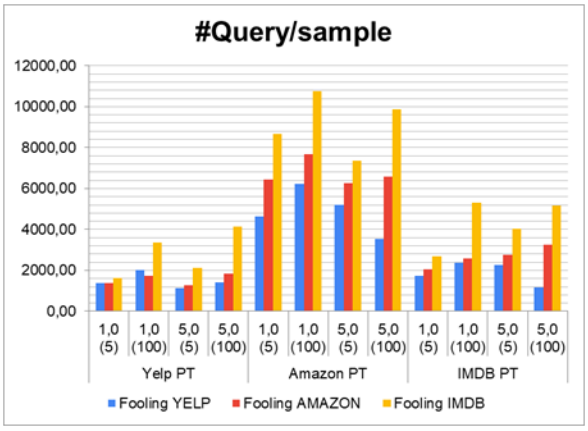

Fig. 4 Mean number of queries performed per sample on sentiment models.

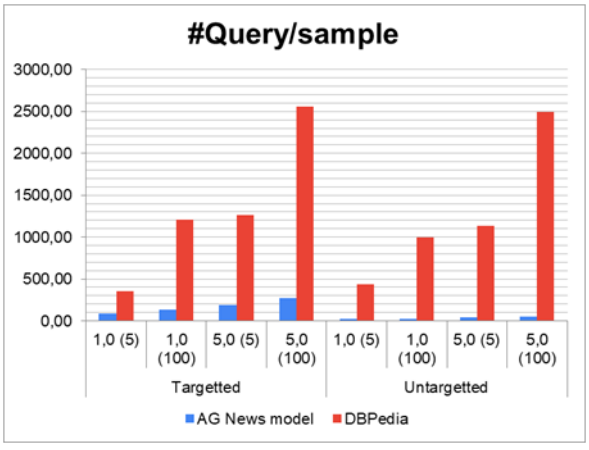

Fig. 5 Mean number of queries performed per sample on categorization models.

swaps, however the adversary could attack the BiLSTM models with more confidence. Again, Amazon polarity table was an important resource to corrupt samples with high confidence in the sentiment tasks. On the multi-class, $\Delta confidence$ on DBPedia is lower than the unbounded case, since these models require more swaps to construct confident attacks.

Confined Attack: Similar to the results of the previous strategies, BiLSTM corruption attained higher $\Delta confidence$ (Figure 2.S4 and Figure 3.M4). In sentiment tasks, it came with the expense of more token substitution. On the cross-domain polarity experiments, Amazon polarity scores resulted in the highest target confidence. We were still able to construct corrupt samples that assign high confidence in sentiment tasks, and the attacks on AG's News victims there was noticible confidence change, whereas the attacks to DBPedia barely showed any success.

Polarity attacks are highly reliant on confident results attained from the black-box model. The adversary must excessively send requests to the service, which can be expensive on both time and monetary cost. We recognize this is a weakness of any query-based composition of adversary samples on black-box. In order to provide further insight into this matter, we present Figures 4 and 5.

On sentiment model experiments, it is clearly noticeable that polarity table that is used in crafting the attacks is determinate on the frequency of requests sent for crafting a single sample. As previously stated, polarity scores attained by Amazon test samples result in higher attack confidence. We suggest the reason behind is the richness of the sample space (400K). Polarity scores can be calculated with more precision, because the adversarial crafting scheme experiences more legitimate samples.

IMDB polarity scores are better than Yelp scores, because it has longer sequences. In other words, richness of polarity source is both number of samples in the set and the sequence length. Richer the polarity source, closer the scores computed would become, which in turn results in excessive querying. Therefore, there is a trade-off between confident attacks and the cost of query per sample. As we expected, sequence length of the clean samples also appears to be a determiner in query cost, due to broader candidate search area. Nevertheless, we realize these inferences should be looked into to analyzed further in future work.

In both classification domains, we see that it becomes less costly as the strategy is closer to confined attacks. This is an expected outcome, as both δ and α determine the scope of the search space.

Possible automated defense against token swap attacks would be based on semantic similarity measures. Black-box service provider might apply a semantic evaluation for the inputs against adversarial samples. One way of semantic evaluation is by using word embeddings. ELMo embeddings [5], for instance, retain syntactic and semantic information of the words, and can be used to analyze semantic similarity between two text samples. We used Google's pretrained ELMo v3 and compared the cosine similarity between original and corrupted samples, in order to evaluate how the corrupt samples deviated from the originals semantically. Figure 6 and 7 show the mean similarity results under unbounded and restrained attack strategies. Results indicate that the corrupt sentiment samples are very similar semantically to the originals. Even though it is harder to perform confident targeted polarity attacks in multi-class setting, multi-class corrupt samples measure similar with scores around 0.90. The

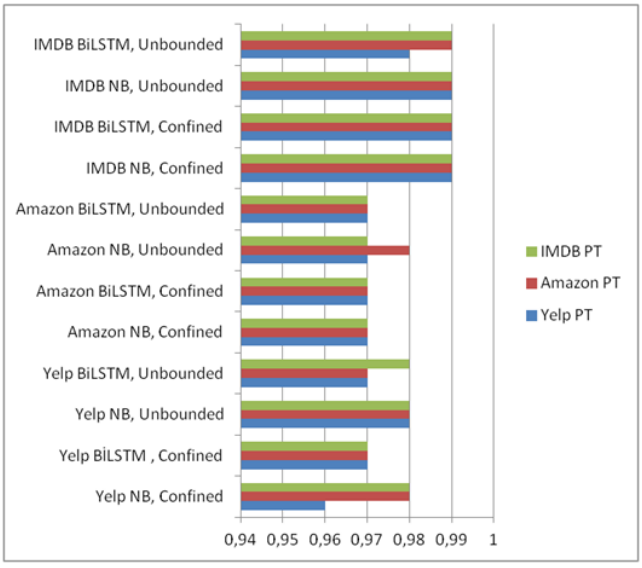

Fig. 6 Mean similarity scores on confined and unbounded attack strategies for sentiment models.

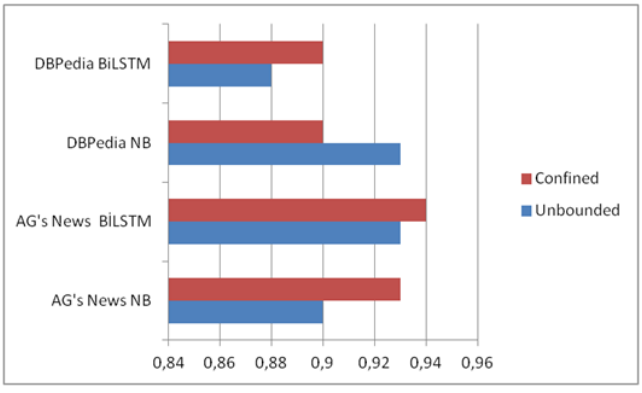

Fig. 7 Mean similarity scores on confined and unbounded attack strategies for categorization models.

results show that there is not much semantic change.

4.2 Transferability

In this section, we evaluate the transferability of the corrupt samples attained from the experiments. Since all the sentiment models perform binary classification of positive/negative, transferring the samples between models is easy. We again use the 100 held-out original samples and 100 corrupted samples for each model. Figure 8 shows the difference in accuracy of the respective attacks (rows) on the victim models (columns) on sentiment tasks which include cross-domain polarity configurations. Higher this difference, higher the success of our attacks.

The first thing that draws attention across all the

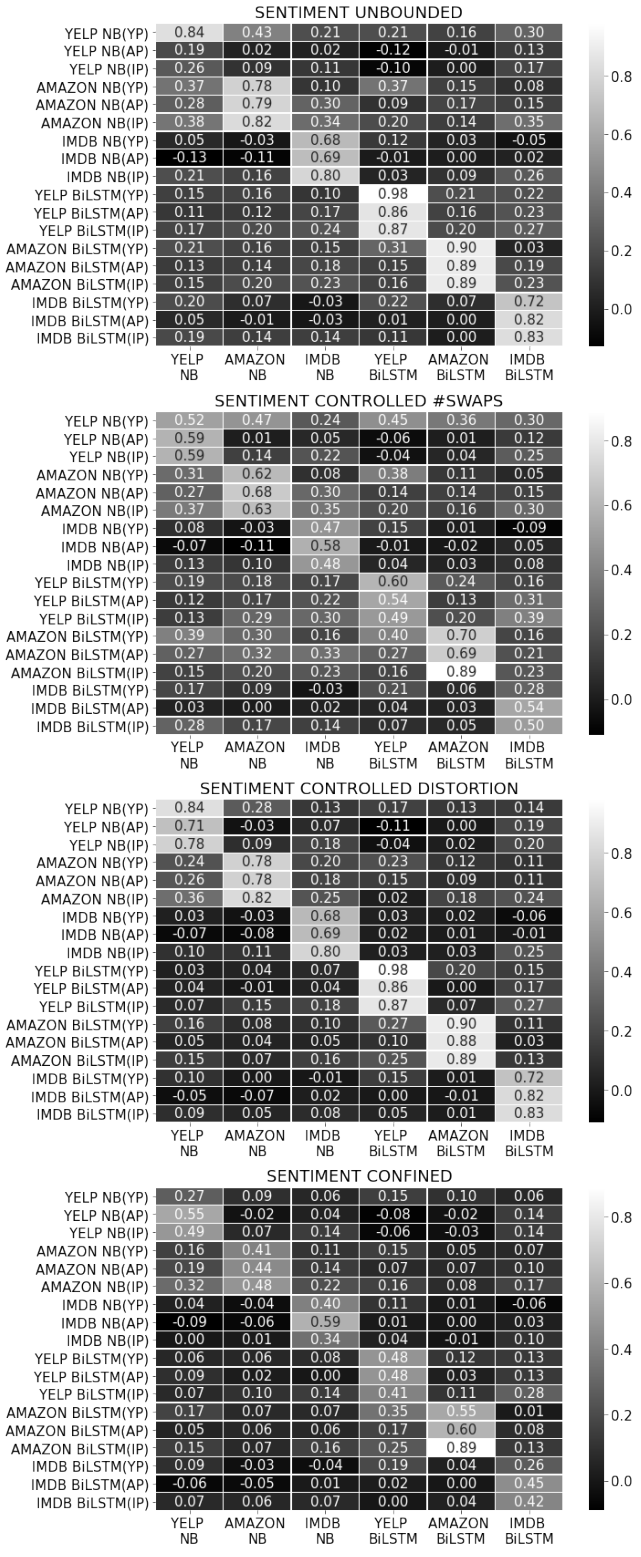

Fig. 8 Transferability of Adversarial Sentiment Samples. Rows represent attackers and columns the victim models. (AP: Amazon Polarity, IP: IMDB Polarity, YP: Yelp Polarity)

figures in this subsection is the fact that the attacks are not much transferable. This is an expected outcome, as we craft the adversarial samples into the mold of the black-box we are attacking. The corrupt samples are highly customized for the victim model the attacker queried while composing them.

Throughout the experiments on Section 1, we observe that BiLSTM models can be fooled with less number of substitutions with higher confidence levels, which in turn, increased the attack accuracy.

For a binary classifier, a 50% change in accuracy is a serious compromise of integrity. On the unbounded and controlled number of swaps strategies, we see that the adversarial samples generally were very effective. Samples crafted via cross-domain polarity were transferable among the NB classifier the corrupt samples produced with (diagonals), with the exception of the cross-domain transferability on Yelp NB victim. Even when the attacker applied confined strategy, the threat to system integrity was retained with the most of the settings.

4.3 Comparison with Baselines

Both our baselines substitute selected tokens with the out of character out-of-vocabulary indicator $< unk >$. Although black-box character perturbations and out of vocabulary token swap attacks may differ on their methodology, they eventually aim the victim model's bias towards unknown tokens. Our baselines test out how the polarity attack compare to the effects of such attacks.

For *random baseline*, up to δ tokens are randomly swapped with the token $< unk >$ until the desired confidence threshold is attained. Hot terms baseline performs a naïve selection and $< unk >$ substitution where $T = 1000$ for $y \rightarrow y^*$. For any sample, up to δ hot tokens are replaced with $< unk >$. Figure 9 and Figure 10 show the results under controlled number of swaps and unbounded strategies. Once more, we used the 100 held-out samples on each dataset to craft the new ones.

Since we swap directly with $< unk >$, the distortion was high even if only one replacement occurred. In this part of the experiment we set the upper bound $\delta = 100$ for the maximum distortion and $\delta = 5$ for minimum distortion.

Sentiment plots show clear separation between polarity based and baseline attacks (Figure 9). Given enough freedom of swap operation, random attacks can expectedly produce more confident malicious samples.

Although the multi-class plots did not exhibit such clear split, polarity attacks over all produced more confident samples with less number of replacements(Figure 10).

4.4 True Black-box Attack

IBM Watson Natural Language Understanding service provides document-level sentiment analysis, where both label and the confidence score is returned. We set a query limit of 100 per sample. We perform δ-controlled attacks with parameters $\delta = 5$, $\alpha = 5.0$ and $\theta = 0.51$ with 10 select samples. A sample corrupted by IBM Watson and our other experimental models is shown in Figure 11. The attack parameters are the same across all the samples.

5 Extended Discussion

The cross-polarity experiments show that Amazon polarity, which is the richest source, is better on corrupting the samples. We suggest that the size of the polarity source is very important to record more precise polarity scores. Score precision, however, comes with the trade-off of the query cost. The richness of the polarity source is very important for the number of queries need to be made. When the scores are fine-tuned, they tend to get closer, and it expands the search space drastically. In future work, we will look into important determiner properties of the polarity source for these issues. It is harder to craft adversarial samples for multi-class problems. Nevertheless, we were able to produce successful attacks when we compromised on either distortion range or the number of swaps.

Polarity attacks are fine-tuned to the victim model, thus malicious samples do not transfer well between models. Nevertheless, polarity sources do transfer effectively. This means the polarity table the adversary fills in using the data it collected from an arbitrary source can be used to perform black-box attacks. Confidence, however, would depend on the richness of the source, which means at least an adequate representation of the classes.

We use ELMo emdeddings to show that the corrupted samples retain their semantic similarity with the originals, which means the attack is not easy to detect by simple automated counter-measures. Furthermore, the human evaluation in our true black-box attacks revealed that corrupted samples assigned mostly to the same semantic category as the original samples.

One defense direction that we suggest against our polarity attack is measuring temporal polarities of the terms in text streams to update NLU services' general polarity biases. There are several recent stud-

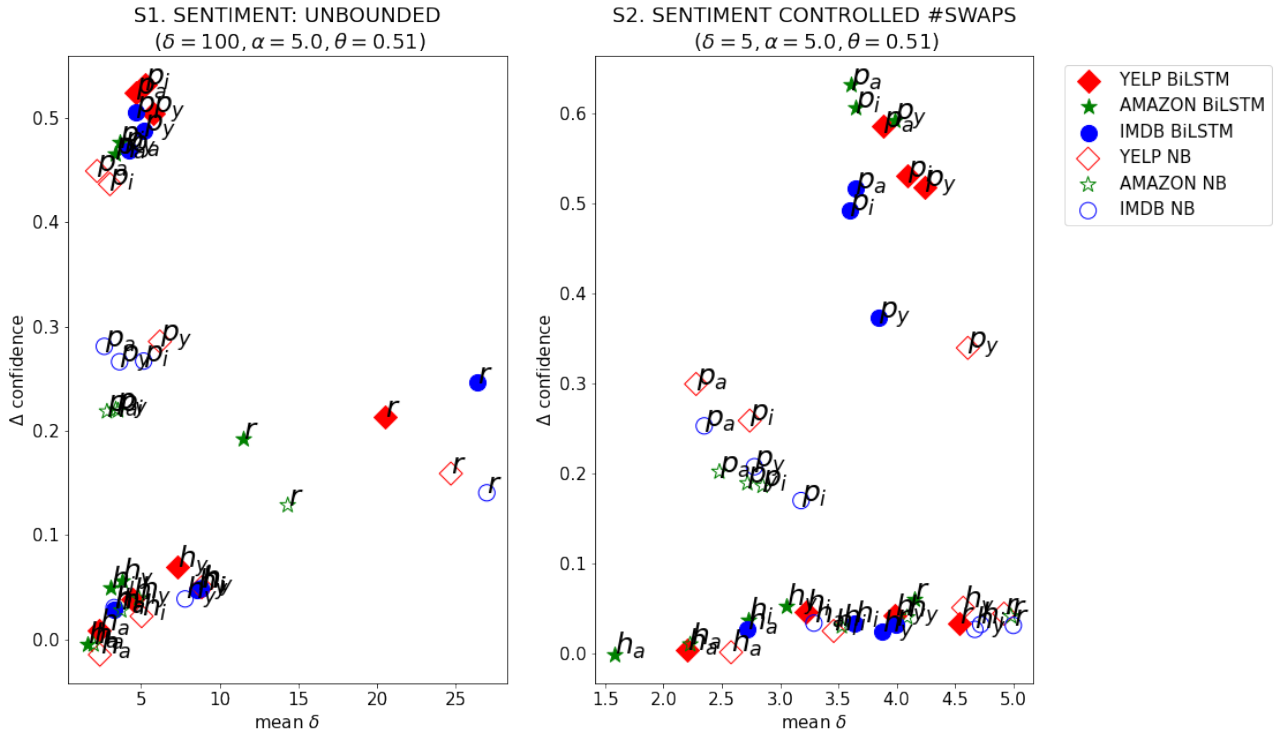

Fig. 9 Random, Hot Terms and Polarity Attacks. p_a: polarity attack that utilizes polarity table of AMAZON test set, p_i: polarity attack that utilizes polarity table of IMDB test set, p_y: polarity attack that utilizes polarity table of YELP test set

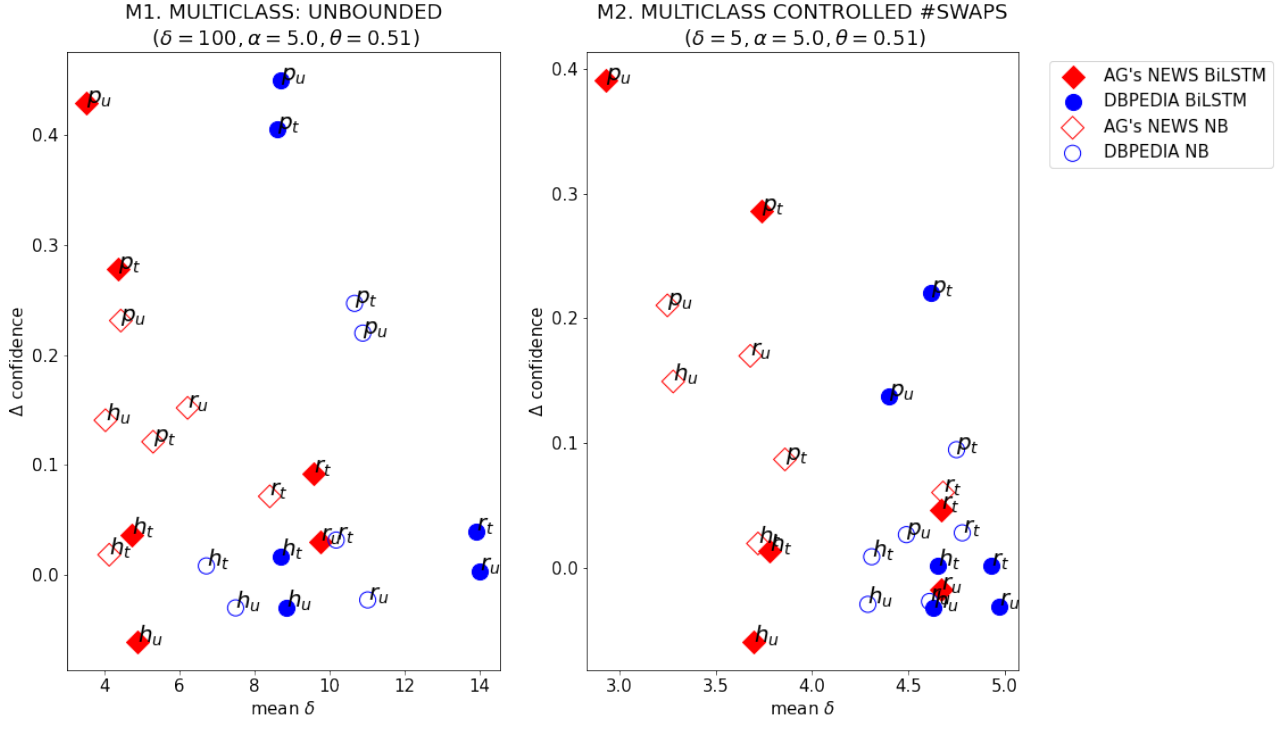

Fig. 10 Random, Hot Terms and Polarity Attacks. p_t: targeted polarity attack, p_u: untargeted polarity attack.

Fig. 11 Generated samples where α=5.0, δ=5 and θ=0.51. The same original sample is corrupted via different model-polarity table combinations. Replacements are shown bold and underlined.

ies that inspire such defense. AL-Sharuee et. al [6] updated the sentiment patterns over time in a unsupervised manner. Such pattern updates would make it harder for the adversaries to keep up with the polarity pattern, because it would most likely need to update the external data it collects and spare good amount of labeling budget for each update. The work of Fkih and Omri [7] is based on a similar scheme as well. Rather than looking at class-vs-class polarity, the model learned relevant-vs-irrelevant terms via an HMM module. They utilized elaborate linguistic features and used a sliding window. Such mechanisms can lead to promising research directions on mitigating NLU service exploitation. Nonetheless, our polarity score is a much more simple way of determining important terms on both one-vs-one (targeted) and one-vs-rest (untargeted) classification settings if the adversary can undertake the periodic expensive labeling cost. This potential defense; however, can be overwhelmed by detecting and calculating a "moving polarity" for bursty terms. A term is "bursty" if it appears with sudden and increasing frequency in a steam of text. They are important features for topic modeling in text streams [8]. We plan to investigate this research direction in future work.

Any defense we would employ prior to the query processing would mean additional computational overhead. This would, in turn, affect the response time, causing inconvenience to the client. Therefore, it is reasonable to first implement a light-weight anomaly detection scheme, then redirect the suspicious queries for further investigation. Kolomvatsos [9] suggested a processor assignment based on past behavior, like load and execution time. A similar approach can be employed by NLU services to mitigate our attack. A few specific past behavior of query request/response or non-semantic meta-features can be logged such that suspicious query can be caught and assigned to a specialized processor for further investigation.

References

1. Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pages 372–387. IEEE, 2016.
2. Ali Daud, Juanzi Li, Lizhu Zhou, and Faqir Muhammad. Knowledge discovery through directed probabilistic topic models: a survey. *Frontiers of computer science in China*, 4(2):280–301, 2010.
3. Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classi-

fication. In *Advances in neural information processing systems*, pages 649–657, 2015.

4. Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pages 142–150. Association for Computational Linguistics, 2011.
5. 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.
6. Murtadha Talib AL-Sharuee, Fei Liu, and Mahardhika Pratama. Sentiment analysis: dynamic and temporal clustering of product reviews. *Applied Intelligence*, pages 1–20, 2020.
7. Fethi Fkih and Mohamed Nazih Omri. Hidden data states-based complex terminology extraction from textual web data model. *Applied Intelligence*, pages 1–19, 2020.
8. Wayne Xin Zhao, Chen Liu, Ji-Rong Wen, and Xiaoming Li. Ranking and tagging bursty features in text streams with context language models. *Frontiers of Computer Science*, 11(5):852–862, 2017.
9. Kostas Kolomvatsos. An intelligent scheme for assigning queries. *Applied Intelligence*, 48(9):2730–2745, 2018.