Detecting Anomalies in Sequences of Short Text Using Iterative Language Models

Cynthia Freeman,[1 2] **Ian Beaver,**[1] **Abdullah Mueen**[2]
[1]Verint Intelligent Self-Service, [2]University of New Mexico
cynthia.freeman@verint.com, ian.beaver@verint.com, mueen@cs.unm.edu

Abstract

Business managers using Intelligent Virtual Assistants (IVAs) to enhance their company's customer service need ways to accurately and efficiently detect anomalies in conversations between the IVA and customers, vital for customer retention and satisfaction. Unfortunately, anomaly detection is a challenging problem because of the subjective nature of what is defined as anomalous. Detecting anomalies in sequences of short texts, common in chat settings, is even more difficult because independently generated texts are similar only at a semantic level, resulting in an abundance of false positives. In addition, literature for detecting anomalies in time ordered sequences of short text is shallow considering the abundance of such data sets in online settings. We introduce a technique for detecting anomalies in sequences of short textual data by *adaptively* and *iteratively* learning low perplexity language models. Our algorithm defines a short textual item as anomalous when its cross-entropy exceeds the upper confidence interval of a trained additive regression model. We demonstrate successful case studies and bridge the gap between theory and practice by finding anomalies in sequences of real conversations with virtual chat agents. Empirical evaluation shows that our method achieves, on average, 31% higher max F1 scores than the baseline method of non-negative matrix factorization across three large human-annotated sequences of short texts.

Introduction

We work for a company that designs and builds domain-specific Intelligent Virtual Assistants (IVAs) on behalf of other companies and organizations, typically for customer service automation. Many companies deploy IVAs for problem resolution and cutting costs in call centers.

A business manager using an IVA to enhance his company's customer service can analyze interactions between customers and IVAs to identify interactions leading to customer complaints like in Figure 1. This interaction indicates the business manager may need to include options to reprint a gift certificate on the company's website and have the IVA direct customers to it, or if such an option already exists, the IVA is unaware of the option and should be updated and the website should make the printing option more conspicuous.

CUSTOMER : `Reprint a gift certificate?`
IVA : `Gift Certificates can be purchased at XXXXX.com.`

Figure 1: An anonymized anomalous conversation between a customer and a live airlines IVA. Printing, not purchasing, gift certificates is the customer's intent.

The increasing adaptation of IVAs creates a problem; there is a corresponding increase in the number of human-computer interactions to be reviewed for quality assurance. Therefore, discovering a means to expedite review and analysis of these interactions is critical. This requires efficient detection of anomalies in conversations. Conversational turns tend to be short (45 characters per user turn on average in our data) and are ordered by time.

Detecting anomalies under such conditions is difficult. Textual anomaly detection, even without such constraints, is already a notoriously difficult problem for a multitude of reasons: **1**) What is defined as anomalous may differ based on application. The textual item `lmao doc martens are just emo timbs` would probably be anomalous to an IVA answering questions about airline travel but not so on Twitter. **2**) What is anomalous today may not be anomalous tomorrow which is especially true for applications such as IVAs. Introduction of a new type of promotion such as a credit card offer may create textual items that are found to be anomalous initially. However, they must be considered normal soon after introduction. **3**) It is unrealistic to assume that anomaly detection systems will have access to thousands of tagged data sets. For chat text, annotated data sets for anomalies are even more limited; we have the NPS Chat Corpus (Forsythand and Martell 2019) which is only tagged for speech and dialogue acts, the Twitter Triple Corpus (Sordoni et al. 2015) and Ubuntu Dialogue Corpus (Lowe et al. 2015) where both are not annotated for anomalies, and UseNet (Shaoul and C. 2013) which is also not annotated and specifically omits documents with less than 500 words. **4**) Non-anomalous data occurs in much larger quantities than anomalous data. This can present a problem for a machine learning classifier approach to anomaly detection as the classes are not represented equally. Thus, an accuracy measure might present excellent results, but the accuracy is

only reflecting the unequal class distribution in the data (the *accuracy paradox*).

We address these difficulties in detecting outliers in sequences of short textual data by using cross-entropies from *iterative language models*.

As we work for an IVA company, we have access to massive quantities of chat data. For our experiments, we selected a large international airlines IVA which interacts with users on the airline's website and mobile application, providing travel advice such as flight status information, baggage and security rules, and even helps with the booking process. This particular assistant was selected as it is a very active IVA with a diverse user base. On average, it responds to 4.6 user inputs per second and engages in 115.5 unique conversations per minute with a global user base.

The iterative language model updates in an adaptive manner, based on the perplexity of the language model. We compare the iterative language models to a non-negative matrix factorization method built for textual anomaly detection that has been reported to outperform many other commonly used baselines such as robust principal component analysis (RPCA) and singular value decomposition (SVD). Our iterative language model achieves, on average, 31% higher max F1 scores on a large human to IVA conversational data set and is also *unsupervised*.

Related Work

Existing studies cannot address the problem of how to detect outliers in sequences of short text. Of the anomaly detection methods that are designed for a textual domain:

1. Existing methods often assume that the pieces of text are large (Guthrie 2008) containing at least 1000 words or are full-length newspaper articles as in (Zhuang et al. 2017). Short text does not have enough content or words which hinders the application of conventional machine learning and text mining algorithms (Chen, Jin, and Shen 2011).
2. Existing methods are often built for very specific tasks such as authorship identification (Guthrie, Guthrie, and Wilks 2008) or detecting outlier sections in legal documents (Aktolga, Ros, and Assogba 2011). For example, in (Aktolga, Ros, and Assogba 2011), outliers in legal documents are detected by exploiting bill structure, specifically *Sections*, the smallest units of a bill. However, detecting anomalies in sequences of short text is more general and includes not just IVA conversations but also social domains like Twitter and Facebook (Bakarov, Yadrintsev, and Sochenkov 2018; Nedelchev, Usbeck, and Lehmann 2020).
3. Existing methods are typically not built for anomaly detection in time ordered text. For example, the work in (Jain et al. 2019) involves time ordered text, but the goal is to detect malicious chatbots instead of anomalous texts from real users.
4. Existing methods do not take into account the dynamic nature of data such as chat or tweets that make it difficult to keep models up to date. In (Xia, GAO, and others 2005), support vector machines (SVMs) trained on chat data annotated for anomalies need to be frequently updated or performance suffers. Frequent periodic retraining of the SVM is not feasible as this requires constant annotation of chat corpora.

We address these problems by identifying outliers via language model cross-entropies, inspired from the work in (Danescu-Niculescu-Mizil et al. 2013) where cross-entropies are used to predict how long a user will stay active in an Internet community.

Language models can be *parametric* or *nonparametric*. Parametric approaches include deep learning techniques but require large quantities of data and often cannot adapt to rapid changes in the distribution of the data in an online setting. Nonparametric approaches include count-based techniques. Although they tend to perform worse compared to parametric approaches, nonparametric approaches can efficiently incorporate new information and require significantly less data (Jozefowicz et al. 2016). Given our data's sparse and dynamic nature, we restrict ourselves to nonparametric approaches (more specifically, statistics on N-grams). By incorporating nonparametric techniques to account for *short* text and updating the language model on a sliding window when its perplexity is too high, we can also account for our data's *dynamic* nature.

Methods

We begin by giving some background on language models and how we use cross-entropies to detect anomalies in sequences of short text. We then introduce our iterative language models[1] and discuss our baseline.

Language Models

A language model is a probability distribution over sequences of symbols pertaining to a language (Jurafsky 2000), and the perplexity is used to evaluate the quality of a language model where the lower the value, the better. For bigrams, the perplexity of the sequence $W = w_1 w_2 ... w_N$ is: $(\prod_{i=1}^{N} \frac{1}{P(w_i|w_{i-1})})^{1/N}$. Our iterative language model "slides" (retrains) when the perplexity reaches a threshold.

Cross-Entropy

The cross-entropies of textual items (Danescu-Niculescu-Mizil et al. 2013) are used to determine anomalies. The cross-entropy of a textual item p given a language model LM is: $H(p, LM) = -\frac{1}{N} \sum_j log(P_{LM}(p_j))$ where p_j = n-grams making up p, N = number of n-grams making up p, and $P_{LM}(p_j)$ = probability of n-gram p_j under the LM. Higher n-grams are left for future work due to the shortness of data. The higher $H(p, LM)$ is, the more surprising the item p is given the recent, past linguistic state. In other words, a low $H(p, LM)$ means that p reflects what is commonly seen in the past.

[1]Implementations available on `https://anon-share.s3-us-west-2.amazonaws.com/ilm_2020.zip`

Iterative Language Models

How the iterative language model (ILM) is updated and used to detect anomalies is highlighted in Algorithm 1. If a language model is trained on texts from dates or indices a to b, we represent this language model as $LM_{a:b}$.

The model takes as input a list of textual items. In our IVA application, every textual item is a user turn in conversations with an IVA. This list is sorted by time. The ILM takes as parameters the following: **1)** $thresh_{perplex}$**=** the threshold for perplexity to retrain the language model, **2)** x = training window size, and **3)** n = the n-gram to use for cross-entropy calculation. The LM is trained on the first x many textual items, creating $LM_{0:x}$. For every textual item i onwards from x, the cross-entropy and perplexity are determined using the language model and chosen n-gram. If the perplexity $> thresh_{perplex}$, we slide the language model forward by retraining on new textual items.

We input the cross-entropies to Facebook Prophet (Taylor and Letham 2018), an additive regression model involving a special time series decomposition method with a piecewise linear or logistic growth curve trend, a yearly seasonal component modeled using Fourier series or a weekly seasonal component, an optionally user-provided list of holidays, and an error term that is assumed to be normally distributed. Parameters are estimated using MAP. Prophet formulates the forecasting problem as a curve-fitting exercise, similar to generalized additive modeling. Thus, Prophet can innately handle time series with missing time steps. For our iterative language model, we have a time series of cross-entropies, and we do not necessarily have cross-entropies for every time step; this is dependent on when a customer chats with an intelligent virtual assistant, but Facebook Prophet can deal with these irregularly sampled time series. We input the cross-entropies to Facebook Prophet and train a forecasting model, using the confidence interval of the model to detect anomalies. Prophet was chosen due to its ease of use, requiring little expert knowledge, and its open availability, but other viable options include any time series anomaly detection method that can incorporate irregular sampling.

Algorithm 1: Iterative Language Model

input : $textualItems$, a list where every element is a user turn in a human to IVA conversation
output : allCES, a list of lists containing the cross-entropies of textual items for every slide
parameter : x (initial training size), n (for n-gram), $thresh_{perplex}$ (perplexity threshold)

```
textualItems ← sorted(textualItems);
LM ← trainLM(textualItems[0:x], n);
CES ← [];
allCES ← [];
lastSlide ← x;
for i in range(x+1, length(textualItems)) do
    p ← textualItems[i];
    p_j ← determineNgrams(p,n);
    N ← length(p_j);
    CES.append(−(1/N) Σ_j log(LM(p_j)))
    if (Π_{k=1}^{N} 1/P(w_k|w_{k−1}))^{1/N} > thresh_perplex then
        LM ← trainLM(textualItems[i-x:i], n);
        allCES.append(CES);
        CES ← [];
return allCES
```

Baseline: TONMF

There are no existing studies that can address the problem of how to detect outliers in sequences of short text. However, we provide the closest baseline possible that we could find for this task. We use as a baseline a non-negative matrix factorization method adjusted for detecting outliers in text called *Text Outliers using Non-Negative Matrix Factorization* (TONMF) developed in (Kannan et al. 2017).

Non-negative matrix factorization (NMF) approximates a non-negative matrix $X \in \mathbb{R}^{p \times n}$ to a lower rank approximation $r \leq rank(X)$. A non-negative basis matrix $W \in \mathbb{R}^{p \times r}$ and coordinate matrix $C \in \mathbb{R}^{r \times n}$ are determined that minimizes $||X - WC||_F^2$ where F is the Frobenius norm.

In (Kannan et al. 2017), the authors model the outliers along with the low rank space of the input matrix. Suppose A is the term-document matrix. In our application, every row represents a word, and every column represents a user turn in conversations with an IVA. $A \in \mathbb{R}^{m \times n}$ is represented as a sum: $A = L_0 + Z_0$ where $L_0 = W_0 C_0$ and $W_0 \in \mathbb{R}^{m \times r}$ and $C_0 \in \mathbb{R}^{r \times n}$. In other words, every document in A is represented as a linear combination of r topics. In situations where a document cannot be well-represented by L_0, it is placed in the outlier matrix Z_0 as a non-zero entry.

Outlier scores for documents are calculated by the L2 norm of columns in Z_0. We feed these outlier scores to Prophet (Taylor and Letham 2018) as like with the iterative language model. W_0, C_0, and Z_0 are determined via block coordinate descent for computational simplicity.

Empirical Evaluation

We begin with a description of the large, real world data sets used and how they were annotated. We then proceed with a description of how data was preprocessed and parameters chosen and conclude with results comparing the performance of the iterative language model and TONMF.

Data Set and Annotation

The airlines IVA that we selected for our experiments can recognize 1,230 unique user *intentions*, or interpretations of a user input that allows one to formulate the best response. For example, if the customer asks about upgrading his flight due to health issues, the IVA determines that the customer's intent is about *First Class Upgrades* and responds accordingly. The intentions are used as a class label within the IVA. Once the IVA determines the user intention, the response associated with that intention is returned.

We selected three intents of varying popularity levels to monitor for our experiment: *Find Companion Fare Discount Code*, *First Class Upgrades*, and *Gift Certificates*. *Find Companion Fare* and *First Class Upgrades* are in the top ten most frequently hit intents; in one year of logs *First*

Class Upgrades was hit 58,187 times and *Find Companion Fare Discount Code* was hit 56,389 times. We also wanted to experiment with an intent that was only moderately popular, so we included *Gift Certificates* which has 8,378 textual items.

Unlike newspaper articles or movie reviews, customer text in IVA conversations tends to be much shorter, presenting a significant challenge for tagging. For every intent, the user text was fed to a language model. IVA responses are excluded because the response is usually identical for the same intent. However, the IVA response was provided for annotators. A graduate student fully tagged the *Gift Certificates* intent data set for anomalies. However, for *Find Companion Fare Discount Code* and *First Class Upgrades*, only 3,000 user inputs of each intent were annotated due to the enormity of these two data sets. The annotator was instructed to mark any of the following as anomalous:

1. **Missed Intent (not due to preprocessing):** Sometimes the IVA will incorrectly classify a user's intent in the conversation (such as in Figure 1). As another example: in the *Gift Certificates* intent, the user may ask about Amazon gift cards, but the IVA incorrectly assumes this involves gift cards that can only be used for airline miles. For the *Find Companion Fare Discount Code* intent, the customer may ask about where to apply the code when purchasing a ticket online, but the IVA directs the user to an account link to see existing discount codes instead. For the *First Class Upgrades* intent, the IVA provides options on how to buy such upgrades instead of helping the customer determine if upgrades are even available in the first place or are already bought out for a particular flight.
2. **Spelling Mistakes and New Vocabulary:** A spelling mistake may be infrequent, and, therefore, be marked as an anomaly such as: `buyiing gift certificates`. Novel terminology assigned to an existing intent may mean that a new product or service has been released that needs to be added to the IVA's intent library.
3. **Preprocessing Errors:** A preprocessing step done by the IVA may cause an error in intent classification. For example, the user may state that he is looking for gift certificates because they are missing in his account. However, the preprocessing step converts `looking for gift certificates` to `search for gift certificates` which brings up a menu of gift certificates one can buy instead of pre-bought gift certificates under one's account.
4. **Multiple Intents:** The user may ask something that has *multiple* possible intents. For example, the user may ask for a gift certificate for miles because of a death in the family. Acceptable intents would include *Gift Certificates* as well as *Bereavement Fare*.

Most of these categories require attention from the business manager to improve IVA performance and prevent issues from reaching a larger set of customers over a longer duration of time. For example, assuming the intent under review was defined for answering questions around *Gift Certificates* that can only be used for flights, a business user may see the following anomalous scenarios that need attention:

- **For category 1: Missed Intent**, if customers ask about using Amazon gift cards and the IVA incorrectly returns the *Gift Certificates* response which states they may be used for purchasing flights, this is indicative of user confusion on how other types of gift cards can be applied. If other gift cards cannot be used for purchasing flights, the requirements must be made more explicit on the airlines website or there needs to be a new intent generated in the IVA for these type of redemption questions.
- **For category 2: Spelling Mistakes and New Vocabulary**, identifying novel spelling mistakes are helpful for our IVA developers in building our word to vocabulary term mappings. As a preprocessing step in the IVA, words are stemmed and then mapped to specific vocabulary terms. For example, words such as `baggage` or `carry-on` are mapped to a `Luggage` vocabulary label. We have a standalone tool to build up this vocabulary by loading a set of user inputs and then exposing all of the words that are unknown to the IVA. Content creators can then quickly associate misspellings to which vocabulary labels they belong to and then export these changes for inclusion in the next IVA version. Also, the identification of unexpected words in a given intent may mean the context around the words intended to map to the intent may have changed. For example, before 2013, the word `pixel` in a device help desk IVA would have been associated with a measurement of screen resolution. But within that year, there would begin to be occurrences of the bigram `Google pixel` in conversations for device support that would be flagged as anomalous during the first several occurrences. This would alert IVA designers that they need to differentiate between questions about screen resolution and a specific smart phone device in the IVA intents.
- **For category 3: Preprocessing Errors**, converting `looking for gift certificates` to `search for gift certificates` also indicates a problem in our word to vocabulary term mappings in our IVA preprocessing step. Normalizing `looking` to `search` may be too great an assumption, and we must consider the possibility of the word `looking` to apply to multiple situations such as a missing gift certificate in an account.
- **For category 4: Multiple Intents**, we can identify cases where the IVA can be more personable. If a customer asks for gift certificates because of a death in the family, the IVA can still direct the user to the *Gift Certificates* intent but also use apologetic language.

As our experimental data is IVA to human conversations, one might ask why we do not just consider sentiment analysis, escalation detection (e.g. `Can I talk to a human?`), or intent misclassification for anomaly detection. Simply performing sentiment analysis or escalation detection on the text, although possibly helpful, is not enough. For our IVA to human conversational data, once customers determine that their concerns are not being addressed, they

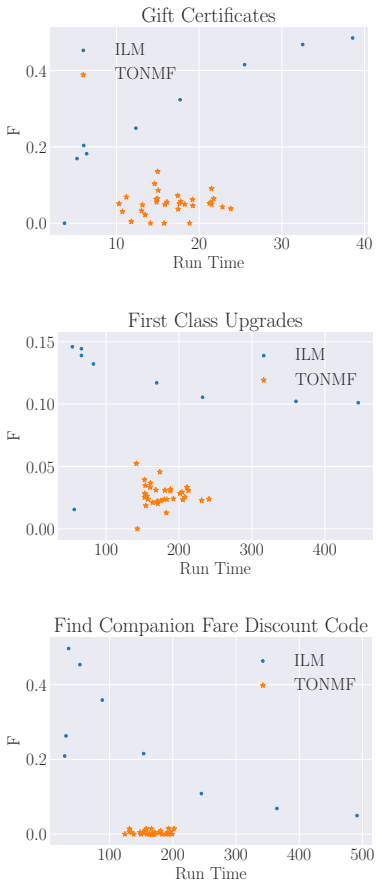

Figure 2: Run time (in seconds) versus F1 scores for the intents *Gift Certificates*, *First Class Upgrades*, *Find Companion Fare Discount Code* for the iterative language model (ILM, blue dots) and baseline (TONMF, orange stars) where every point represents a parameter configuration. The confidence interval was .95.

very frequently just leave the chat instead of spewing expletives or expressing frustration. Anomalies in IVA conversations also encompass more than just multi-label classification errors as shown in the above four anomaly categories. Note, however, there still needs to be a way to detect any missed intent classification. This is not as simple as just looking at classifier confidence. Intent classification can be done via a unique combination of machine learning classifiers, regular expressions, and conversation flow decision trees. We wanted a generic enough methodology that only requires customer text which is *independent* of the implementation details of the IVA.

Application

We use the adaptive update language model, set $x = 2000$, and experiment with various $thresh_{perplex}$ (where $2 < thresh_{perplex} < 3$). Every intent data set has its own language model. In calculating cross-entropies, we normalize by just using the first c words of p. This form of normalization is employed in (Danescu-Niculescu-Mizil et al. 2013) as there is no consensus on how to normalize entropy measures. We use $c = 30$ where results are stable across multiple choices of c.

We also perform TONMF on every $2,000$ many textual items (to make it comparable to the ILM) and test various parameter configurations. Every intent data set has its own application of TONMF. In our implementation of TONMF, we restricted $\beta = 1$ as the creators of TONMF have stated that the algorithm is not overly sensitive to choice of β. We experiment with r =10 to 45 topics per intent. As for α, it balances the importance given to outliers against the matrix sparsity criterion during regularization. The larger α is, the more important the outlier portion of the regularization. However, for lower ranks of r, the increase in the value of α does not make any predictions. This is because, beyond a particular limit, the weights given to the outlier criterion do not supersede the optimization's main objective of extracting low-rank patterns from the data.

Results

Figure 2 shows the run time and F1 scores for every parameter configuration for the iterative language model versus the baseline TONMF, using a confidence interval of .95. For nearly all parameter configurations, the iterative language model (ILM) has higher F1 scores than the baseline (TONMF) across all time. In addition, the ILM can perform faster than TONMF in several cases.

For *Gift Certificates*, the highest F1 score for the ILM was .49 whereas the baseline can only reach .15. For *First Class Upgrades*, ILM achieves .15, and TONMF can only reach .05. For *Find Companion Fare Discount code*, ILM hits .49 whereas the baseline can only hit .014.

First Class Upgrades was the most difficult data set for both the ILM and TONMF. This is because this intent was quickly discovered by the annotator to encompass too many user questions that the IVA was not customized to address. The IVA response to a user question hitting the *First Class Upgrades* intent is: `You can use your miles to upgrade in advance, request a Paid Upgrade during check-in or at the departure gate, or if you're an XXXXX member, you can upgrade for free.` However, this does not cover questions regarding if a first class upgrade for a particular flight is available or bought out, first class upgrade code usage, or how to buy coach seats when only first class seats are available. Yet all of these questions are classified by the IVA under the *First Class Upgrades* intent. In addition, midway through the year, a new premium upgrade was promoted by the airlines company, but the IVA was never updated to reflect this change. Thus, all questions involving these premium upgrades was directed to *First Class Upgrades*, confusing customers. There was a lot more variety in the kinds of questions customers asked in the *First Class Upgrades* intent compared to the other two intents, making it difficult for ILM and TONMF to establish a textual norm.

The four types of anomalies were not distinguished during annotation for the sake of time and effort on the anno-

tator's part (especially as the IVA data is proprietary, and, thus, we cannot utilize annotation crowdsourcing tools like Mechanical Turk), but a deeper analysis on the categories would be valuable and is left for future work. For example, we determined that TONMF favors detecting anomalies belonging in the category of unique codes and numbers (e.g. a customer asks if a Discount Code XYZ is valid or if their ticket number is available for an upgrade). In fact, for *Find Companion Fare Discount Code*, textual items containing codes comprised over 85% of TONMFs predictions whereas textual items containing codes only comprise 40% of *Find Companion Fare Discount Code*. Such predictions make up 20% of predictions made by the ILM. Similarly, for *First Class Upgrades*, predictions containing only codes make up 8% of the data set, 20-30% of predictions made by TONMF, and 1-6% of predictions made by the ILM. TONMF generally performs worse; this is most likely due to the fact that TONMF expects larger bodies of text. In (Kannan et al. 2017), TONMF was only tested on Reuters newspaper articles, RCV20, and Wiki People. It would be beneficial to include such analysis on the other anomaly categories.

Accuracy-speed trade-off is universal. For the iterative language model, the higher $thresh_{perplex}$ is, the fewer slides the language model makes (fewer updates), and, in turn, the number of times the language model is retrained is decreased. Thus, the time required to determine cross-entropies for the entire data set goes down. However, even the slowest of the ILM runs was able to process over 6 user turns per second, which is fast enough to deploy in real-time alongside this airline IVA which answers 4.6 turns/sec. For TONMF, an increase in r (number of topics) increases the time needed to solve the optimization problem. Non-negative matrix factorization is a NP-hard problem; thus, the authors make use of block coordinate descent for computational efficiency.

Significance and Impact

Detecting outliers in sequences of short text is a difficult problem because text is typically sparse and high dimensional. However, this detection is vital for IVAs in use by business managers who seek ways to improve the customer experience. Existing techniques assume that the text samples either have no ordering or are long; however, this is not always the case especially in the domains of chat, Facebook comments, or Twitter. The shortage of publicly available, annotated resources compounds our problems, and even if annotated data is available, the dynamic nature of conversational data necessitates constant retraining of models. In this paper, we demonstrate our technique for detecting outliers in sequences of short text using cross-entropies from iterative language models. We take advantage of our company's massive repository of chat data sets to address the lack of publicly available, annotated data. We compare the iterative language model to TONMF as a baseline and achieve, on average, 31% higher max F1 scores on real human to IVA conversations.

Our ultimate goal is to determine how to improve our IVAs given the outliers and the categories they belong to. By deploying our textual anomaly detection system alongside the IVA, we can record anomalies as they happen in real-time for downstream health monitoring of live production IVAs and help identify how the IVA can be improved.

References

Aktolga, E.; Ros, I.; and Assogba, Y. 2011. Detecting outlier sections in us congressional legislation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, 235–244.

Bakarov, A.; Yadrintsev, V.; and Sochenkov, I. 2018. Anomaly detection for short texts: Identifying whether your chatbot should switch from goal-oriented conversation to chit-chatting. In *International Conference on Digital Transformation and Global Society*, 289–298. Springer.

Chen, M.; Jin, X.; and Shen, D. 2011. Short text classification improved by learning multi-granularity topics. In *Twenty-Second International Joint Conference on Artificial Intelligence*.

Danescu-Niculescu-Mizil, C.; West, R.; Jurafsky, D.; Leskovec, J.; and Potts, C. 2013. No country for old members: User lifecycle and linguistic change in online communities. In *Proceedings of the 22nd international conference on World Wide Web*, 307–318.

Forsythand, E. N., and Martell, C. H. 2019. The nps chat corpus.

Guthrie, D.; Guthrie, L.; and Wilks, Y. 2008. An unsupervised approach for the detection of outliers in corpora. *Statistics* 3409–3413.

Guthrie, D. 2008. *Unsupervised detection of anomalous text*. Ph.D. Dissertation, Citeseer.

Jain, S.; Niranjan, D.; Lamba, H.; Shah, N.; and Kumaraguru, P. 2019. Characterizing and detecting livestreaming chatbots. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 683–690.

Jozefowicz, R.; Vinyals, O.; Schuster, M.; Shazeer, N.; and Wu, Y. 2016. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*.

Jurafsky, D. 2000. *Speech & language processing*. Pearson Education India.

Kannan, R.; Woo, H.; Aggarwal, C. C.; and Park, H. 2017. Outlier detection for text data. In *Proceedings of the 2017 siam international conference on data mining*, 489–497. SIAM.

Lowe, R.; Pow, N.; Serban, I.; and Pineau, J. 2015. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. *arXiv preprint arXiv:1506.08909*.

Nedelchev, R.; Usbeck, R.; and Lehmann, J. 2020. Treating dialogue quality evaluation as an anomaly detection problem. In *Proceedings of The 12th Language Resources and Evaluation Conference*, 508–512.

Shaoul, C., and C., W. 2013. A reduced redundancy usenet corpus.

Sordoni, A.; Galley, M.; Auli, M.; Brockett, C.; Ji, Y.; Mitchell, M.; Nie, J.-Y.; Gao, J.; and Dolan, B. 2015. A neural network approach to context-sensitive generation of conversational responses. *arXiv:1506.06714*.

Taylor, S. J., and Letham, B. 2018. Forecasting at scale. *The American Statistician* 72(1):37–45.

Xia, Y.; GAO, W.; et al. 2005. Nil is not nothing: Recognition of chinese network informal language expressions.

Zhuang, H.; Wang, C.; Tao, F.; Kaplan, L.; and Han, J. 2017. Identifying semantically deviating outlier documents. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2748–2757.