Learning to Classify Inappropriate Query-Completions

Parth Gupta[1] and Jose Santos[2(✉)]

[1] Universitat Politecnica de Valencia, Valencia, Spain
pgupta@dsic.upv.es
[2] Microsoft, London, UK
jcsantos@microsoft.com

Abstract. Query auto-completion is a powerful feature anywhere users are querying and is nowadays omnipresent in many forms and entry points, e.g. search engines, social networks, web browsers, operating systems. Suggestions not only speed up the process of entering a query but also shape how users query and can make the difference between a successful search and a frustrated user. The main source of these query completions is past, aggregated, user queries. A non-negligible fraction of these queries contain offensive, adult, illegal or otherwise inappropriate content. Surfacing these completions can have legal implications, offend users and give the incorrect impression companies providing the query completion service condone these views. In this paper, we describe existing methods to identify inappropriate queries and present a novel machine learned approach that does not require expensive, human-curated, blocklists and is superior to these in recall and competitive in F1-score.

1 Introduction

Every day billions of queries are issued in commercial search engines in dozens of languages. These queries reflect users' needs, desires, behaviours, interests but also prejudices. These searches are also the main data source to build the query histogram models that power an auto-completion service [1]. Due to the organic nature of the query histogram model, we estimate 5–10% of the queries are inappropriate to surface to the end user as an auto-completion. The user is still able to type any query completely and get results.

We consider query suggestions inappropriate if they are offensive, condone violence or illegal actions or have a sexual intent. It should be noted a query may contain inappropriate terms but if the intent is clean the query should still be deemed OK e.g. "what is cocaine" vs "where to buy cocaine". A search engine deliberately wants to filter as many inappropriate suggestions as possible due to geopolitical and legal reasons, being preferable to incur in type I errors (false positives) than allowing a true inappropriate query go undetected (a type II error). Therefore, recall is preferred over precision.

J.M. Jose et al. (Eds.): ECIR 2017, LNCS 10193, pp. 548–554, 2017.
DOI: 10.1007/978-3-319-56608-5_47

A typical method to deal with detecting inappropriate queries are substring and pattern match block lists [4–6]. Substring-match blocklists contain strings that can never appear in a query, e.g. swear words. Pattern-match blocklists assume various forms. One is the <entity> <qualifier> pattern where a list of entities e.g. person names, ethnic/religious/political groups with common associated derogatory expressions. When a query contains both a known entity and a derogatory qualifier associated with that entity type, the query is identified as offensive. For instance, a suggestion of the form "X are Y" is blocked by a pattern-match blocklist if $X \in$ {jews, christians, muslims, blacks} and $Y \in$ {stupid, idiots, retarded} while the individual X and Y terms may be acceptable on their own.

While the combination of both blocklist techniques performs acceptably, there are severe limitations: it is a semi-manual process requiring list curation and maintenance; all possible variations of an entity and derogatory terms must be provided, e.g. singular, plural, synonyms; no generalisation power.

The existing literature focuses on natural language or social media text and the techniques are of limited use when a very small context is available as in the case of a web query. In this paper, we propose a new model which learns to represent queries in different clusters of inappropriateness. Such representation is learnt through a supervised latent semantic projection algorithm based on deep neural networks. The proposed clustering method helps to uncover more inappropriate patterns as evidenced by high recall.

2 Approach

Our approach is to create an abstract offensive space where queries can be clustered. The abstract space is built using supervised latent projection methods. Supervised techniques such as deep structured semantic model (DSSM) [3] can incorporate the label information into projection learning. As we aim to learn an abstract space of offensiveness, we adapt DSSM model to incorporate offensive categories by injecting an objective function which clusters queries from same inappropriate categories as described in Sect. 2.1.

2.1 DSSM for Offensive Clusters

DSSM is structurally a deep neural network which models the queries to represent in a low-dimensional latent space.

Let $c_k \in C$ represent the k^{th} inappropriate category where $|C| \geq 2$ and c_0 represents the appropriate (`OK`) category. Hence, $|C| = 2$ represents the binary setting with categories {`OK`, `inappropriate`}. Let $x_{q,c_k} \in \mathbb{R}^n$ be vector representation of query q labelled to belong inappropriate category c_k and n is input dimensionality. Queries are represented as word hashes because considering complete terms explodes the feature space while word-hashes have proven to be effective and efficient [3].

Query x_q is projected to $y_q = \phi(x_q)$ by DSSM (ϕ) where $y_q \in \mathbb{R}^m, m \ll n$ as shown in Eq. 1.

$$\begin{aligned} h_{q,l_1} &= g(W_1 * x_q + b_1) \\ y_q &= g(W_2 * h_q^{(l_1)} + b_2) \end{aligned} \tag{1}$$

where, W_i and b_i represent i^{th} layer weights and bias parameters of the network, $h_q^{(l_1)}$ represent the hidden layer activities and g is hyperbolic tangent activation function. The DSSM is trained to maximise the objective function presented in Eq. 2 using backpropagation [3].

$$J(\theta) = \cos(y_q, y_q^+) - \cos(y_q, y_q^-) \tag{2}$$

where, y_q^+ represents same category query to that of y_q and y_q^- represents a different category query to that of y_q. The objective function $J(\theta)$ encourages those configurations θ which produce higher cosine similarity between queries belonging to the same category and lower cosine similarity between queries that belong to different categories.

Once the DSSM is trained, all the labelled queries are projected into the abstract space. Centroid for each category is calculated as shown in Eq. 3.

$$\mu_{c_k} = \frac{1}{m_k} \sum_i y_{q,c_k}^{(i)} \tag{3}$$

Now a new query y_q is classified to category c_k for which Euclidean distance $d(\mu_{c_k}, y_q)$ is minimum.

3 Experiments and Results

Here we present the experimental set-up to evaluate the effectiveness of the proposed models along with a couple of strong baselines.

3.1 Data

Our dataset consists of 79174 unique queries. These queries were derived from a prefix set biased towards inappropriate terms as follows. The prefix set was created by randomly sampling queries that contained offensive terms and keeping only the first half of the query. These prefix sets were then scraped against the auto-completion service of a commercial search engine, for the US market, and the resulting unique queries gathered.

The unique queries were then human judged for various inappropriate categories via a crowd-sourcing platform, with at least 5 judgements per query. Significant care was put to ensure the quality of judgments with real time audits and by limiting the number of queries a single judge could judge to a few hundred. Real time audits were done by randomly interspersing with the queries to judge a small percentage of non-contentious queries for which we know the

Table 1. Distribution of judgements, in thousands, over the 4 query categories in the dataset.

Cat.	Description	#	%
c_0	Okay	627.2	95.0%
c_1	Violence/illegal/self-harm	7.6	1.2%
c_2	Race/religion/sexual/gender	18.0	2.7%
c_3	Other offensive/profane	7.3	1.1%

ground truth. If a judge did not agree on at least 85% of these non-contentious queries, it would be disqualified a posteriori and all its judgments discarded.

In total, there were 660267 judgements (average 8.3 judgements per query), with the vast majority, 95.0%, being appropriate (`OK`). The statistics of the data is presented in Table 1.

From this labelled dataset, a query inappropriate score is computed as the ratio of inappropriate judgements over all the query judgements. This score is then converted into a binary label, `Inappropriate` if score ≥ 0.2, otherwise `OK`. There are 7284 inappropriate queries, 9.2% of the corpus. The dataset was randomly split into 70% for training and 30% for test.

3.2 Baselines

Blocklists. The blocklist-based approach uses a set of substring and pattern-matching techniques. Semi-manually built blocklists are generated by extracting common inappropriate patterns from user reported feedback and from crowd-sourcing tasks whose goal is to spot inappropriate query leakage in a commercial search engine auto completion service.

The aggregated size of the multiple substring blocklists is in the order of the tens of thousands of terms which block in the order of a few million queries. If a query matches any of the terms in any of the substring blocklists, it is deemed as inappropriate. The pattern-matching blocklists only block a query if it has terms in two complementary lists. There are multiple pattern-matching blocklists, one for each domain, such as {offensive, adult, illegal, violence}.

Table 2 contains a few entries from the English substring-match blocklist and the violence pattern-match blocklists. These lists are updated regularly, grow over time and require manual effort to maintain.

SVM with Word Hashes. We also trained a classifier with lexical features to test whether it can learn a classification boundary from the training data. First, we featurize the input queries by the word-hashing technique reported in [3] which codifies each input term into character 3-grams after marking the start and end of the word. This 3-gram featurization helps handling the sparseness of the features and keeps the feature space limited, especially to scale at web-level. A total of 9590 character 3-grams was obtained from the training data. Secondly,

Table 2. A few entries of the English: (1) offensive substring blocklist (1st column), (2) pattern-match violence block-lists (2nd and 3rd columns)

Offensive	Violence	Viol. modifiers
Beating newborn	Beheading	Video
Blacks should	Execution	Movie
Cannibal recipes	Hanged	Image

we trained a support vector machine (SVM) on the 3-gram featurized training data to obtain a classification boundary.

3.3 Results

We evaluate the baselines and proposed models on the test partition and measure the performance of predicting the `Inappropriate` class with the standard precision, recall and F1-score measures. The results are presented in Table 3.

Table 3. Precision, recall and F1-score of various models. † denotes statistical significance (p-value < 0.01) to corresponding blocklist metric

Model	Prec.	Rec.	F1
Blocklists	70.0%	49.0%	57.6%
SVM with 3-grams	70.4%	43.4%	53.8%†
DSSM with L2 norm	51.1%†	65.0%†	57.2%

The SVM baseline performs worse than blocklists in recall, because blocklists are highly curated while SVM tries to learn the discriminative patterns. Although blocklists and SVM achieve high precision, they obtain relatively low recall pointing to their poor generalization, mostly because they filter only on the lexical features. The DSSM-based clustering method introduced in Sect. 2.1 obtains higher recall. Higher recall points to its power to uncover more inappropriate patterns which was not possible with the lexical methods. An ensemble random forest classifier using 3-grams and distances to centroids with various norms, including cosine distance, as features was also trained but it only marginally improved F1-score over the much simpler DSSM with L2 norm.

It is worth noting the blocklists used are exhaustive, built over several years from larger corpus of queries than the one in this dataset, specifically to block inappropriate web search suggestions in a commercial auto-complete service. The presented blocklist is thus a very strong baseline. A fairer blocklist baseline would be humans crowdsource a blocklist based only on the training set queries and evaluate this blocklist performance on the test set. This was too expensive to do but would be required to filter inappropriate suggestions in a new language with the blocklist technology. In this regard the ML approaches are clearly superior as they can build the inappropriate model directly and inexpensively.

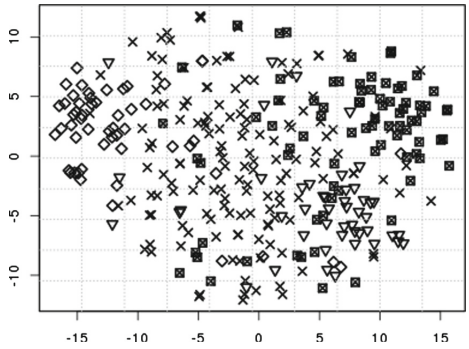

Fig. 1. t-sne 2D projection of the DSSM feature vectors, over-sampling the inappropriate classes for easier visualization. The $\times$ represents the `OK` category, c_0. $\Diamond$ represents c_2, $\triangledown$ represents c_1 and $\boxtimes$ represents c_3.

We show the discriminative power of DSSM technique by visualising the inappropriate clusters in Fig. 1. The 128-dimensional vectors are represented in 2-dimensional scatter plot using t-sne algorithm[1]. It can be noticed queries from particular inappropriate categories form dense well-separated clusters.

Table 4 presents a few correct and incorrect classifications made by the DSSM with L2 norm on the test set. There exists some noise in the judge scores which may be due to limited knowledge of the judges on specific topics. While analysing the false negatives, we noticed those queries are not as inappropriate as the correctly predicted ones. In fact, a false negative such as "batman jokes" is innocuous. In some cases, the query contains words which themselves are inappropriate but in the query context they are not, e.g. "UN office on crime and drug". The latent DSSM clustering features helped to identify such patterns.

Table 4. A few correct and incorrect predictions made by the DSSM.

Query	Judge score	Predicted class
Best guns for women	0.4	`Inappropriate`
Marijuana brownie recipe	1.0	`Inappropriate`
Batman jokes	0.2	`OK`
UN office on crime and drugs	0.2	`OK`

4 Related Work

Although offensive query classification has attained little research attention, offensive language identification in text is, in general, a well known problem.

[1] https://lvdmaaten.github.io/tsne/.

Nevertheless, literature on this problem is limited and mainly evolved around the use of a predefined block list [4–6].

Some work also exists on ML-based approaches to model lexical features for offensive language detection in social media text [2,7]. Our baseline SVM represents such class of methods. However, given the extremely short context in web queries, such ML methods are less attractive to web query classification.

5 Remarks

Detecting inappropriate queries is an important and timely problem with the internet being increasingly used to propagate violent views. We provided a principled solution for the inappropriate query classification problem based on deep neural networks. The experiments carried on a large labelled web query corpus suggest the DSSM approach significantly outperforms ML techniques based on lexical features. The DSSM approach is also superior in recall while being competitive in F1-score compared to expensive, human-curated blocklists.

References

1. Bar-Yossef, Z., Kraus, N.: Context-sensitive query auto-completion. In: Proceedings of WWW, pp. 107–116 (2011)
2. Gianfortoni, P., Adamson, D., Rosé, C.P.: Modeling of stylistic variation in social media with stretchy patterns. In: Proceedings of DIALECTS, pp. 49–59 (2011)
3. Huang, P.S., He, X., Gao, J., Deng, L., Acero, A., Heck, L.: Learning deep structured semantic models for web search using clickthrough data. In: Proceedings of CIKM, pp. 2333–2338 (2013)
4. Mahmud, A., Ahmed, K.Z., Khan, M.: Detecting flames and insults in text. In: Proceedings of ICON (2008)
5. Razavi, A.H., Inkpen, D., Uritsky, S., Matwin, S.: Offensive language detection using multi-level classification. In: Farzindar, A., Kešelj, V. (eds.) AI 2010. LNCS (LNAI), vol. 6085, pp. 16–27. Springer, Heidelberg (2010). doi:10.1007/978-3-642-13059-5_5
6. Spertus, E.: Smokey: automatic recognition of hostile messages. In: Proceedings of IAAI, pp. 1058–1065 (1997)
7. Xiang, G., Fan, B., Wang, L., Hong, J., Rose, C.: Detecting offensive tweets via topical feature discovery over a large scale twitter corpus. In: Proceedings of CIKM, pp. 1980–1984 (2012)