# Boosting Text Classification Performance on Sexist Tweets by Text Augmentation and Text Generation Using a Combination of Knowledge Graphs

**Sima Sharifirad**
Department of computer science, Dalhousie University, Halifax, Canada
s.sharifirad@dal.ca
**Borna Jafarpour**
Huawei Technology, Toronto, Canada
borna.jafarpour@huawei.com
**Stan Matwin**
Department of computer science, Dalhousie University, Halifax, Canada
stan@cs.dal.ca

## Abstract

Text classification models have been heavily utilized for a slew of interesting natural language processing problems. Like any other machine learning model, these classifiers are very dependent on the size and quality of the training dataset. Insufficient and imbalanced datasets will lead to poor performance. An interesting solution to poor datasets is to take advantage of the world knowledge in the form of knowledge graphs to improve our training data. In this paper, we use ConceptNet and Wikidata to improve sexist tweet classification by two methods (1) text augmentation and (2) text generation. In our text generation approach, we generate new tweets by replacing words using data acquired from ConceptNet relations in order to increase the size of our training set, this method is very helpful with frustratingly small datasets, preserves the label and increases diversity. In our text augmentation approach, the number of tweets remains the same but their words are augmented (concatenation) with words extracted from their ConceptNet relations and their description extracted from Wikidata. In our text augmentation approach, the number of tweets in each class remains the same but the range of each tweet increases. Our experiments show that our approach improves sexist tweet classification significantly in our entire machine learning models. Our approach can be readily applied to any other small dataset size like hate speech or abusive language and text classification problem using any machine learning model.

## 1 Introduction

When it comes to machine learning algorithms, the dataset plays a pivotal role in the usability of those models. There are many problems where datasets are imbalanced, data is rare or data is hard to collect, hard to label or the overlap between the classes is high. One of the methods which handles these shortcomings in text classification is text generation. Text generation has been used widely for machine translation, summarization and dialogue generation (Sathish Indurthi et al., 2017) and (Uchimoto, K. et al. 2002). In addition, sentences contain different keywords and concepts. One way of understanding these concepts and getting more information about them is by using linked data and knowledge graphs. The popularity of the internet and advancements in linked data research has led to the development of internet-scale public domain knowledge graphs such as FreeBase, DBPedia, ConceptNet and Wikidata.



*Proceedings of the Second Workshop on Abusive Language Online (ALW2)*, pages 107–114

Knowledge in popular knowledge graphs is usually mined from available online resources such as Wikipedia using natural language understanding techniques or harvested by crowd sourcing or a combination of both. Knowledge graphs are used to represent concepts and their relationships in a computer understandable format. They have a wide range of application in the text analysis domain such as question answering (Xu, Q. et al., 2017) query expansion (Yao, Xuchen et al.,2014), recommendation engines (Voorhees, E, 1994) and many more.

ConceptNet is a common sense knowledge graph that represents approximately 21 million everyday concepts and their relationships using one of the 36 existing relationships such as IsA (e.g. jack IsA first name), UsedFor (e.g. car UsedFor driving ) or PartOf (wheel PartOf car) (Dzmitry Bahdanau et al., 2015). Each fact in ConceptNet has a weight value which shows the degree of truthiness. Higher values show more confidence. In other words, it shows the closeness of the concpts to each other. Wikidata is a wiki project that is used to crowd source structured data which is consumable both by humans and machines. Wikidata contains 4400 types of relationships between more than 45 million concepts. ConceptNet and Wikidata are far from perfectly consistent and complete. Therefore, we use both of these knowledge graphs in our approach for better coverage of word knowledge with more consistency. An interesting source of information in Wikidata is concept descriptions. We use these descriptions for augmenting tweets. For the text generation task, we replace words in each tweet by words that they are connected to in ConceptNet using some of its 19 relations such as IsA, RelatedTo, HasA, HasProperty, etc.

Another approach for improving the classification is text augmentation, adding more information or enriching the text semantically for the purpose of achieving better classification results. Text augmentation has been widely used in bioinformatics, image processing, computer vision, video and audio processing (Björn Gambäck and Utpal Kumar Sikdar. 2017) and (X. Lu, 2006). Even though the most prevalent applications of text augmentation are in the fields of vision or audio, we believe that introducing simple but effective ideas can be useful for text classification tasks. In addition, they can help in reducing the scarcity of the data, avoiding over-fitting due to lack of data and increasing the generality power of the algorithm.

Our contribution in this paper is using ConceptNet, Wikidata and a combination of both for text generation and augmentation in order to improve sexist tweet classification. Even though we have used our approach for sexist tweet classification, it can be readily applied to other text classification problems using any of the existing text classification models. It can also be beneficial for hateful speech and abusive dataset where the data is scarce.

The rest of the paper is as follows: In the next section, we will discuss the prior work on sexism, text generation and text augmentation. Then, in the experiment part, we will go through the dataset, text preprocessing, classification algorithms, and the detailed method of text generation and text augmentation. In the results, we will show the result of text generation and text augmentation and finally the conclusion.

## 2 Related Work

### 2.1 Text generation

Text generation has been studied for many years and computational linguistic and diverse methods have been suggested ever since. Sentence structures are very different and these diversities expand in different types of social media whichmakes text generation harder. For instance, text in Wikipedia is well written and well structured. However, twitter sentences follow mandatory structures in being less than 280 characters (Robert Speer, 2017). Two directions for text generation systems have been suggested. The first method tries to keep the reusability and generality of the sentence without focusing on the structure of the sentence. The second approach tries to keep the structure and template of the sentence (Kingma & Welling, 2013). Uchimoto Kiyotaka et al. suggested their text generation method using keywords candidate coming from a dependency tree.

More recently, deep learning approaches have been utilized for this purpose. Deep generative models have been used for this task. One of these is to use Variational Autoencoders (VAEs). Kingma and Welling, 2013 took advantage of VAEs to encode the data examples to a latent representation and then new samples were generated from that latent space. There have been other works on text question and answer generation using knowledge graphs. Sathish Indurthi et al., 2017 produces the question and answer pair from a knowledge graph. They used

Freebase as their knowledge base to produce the triples and then used them for their question, answer pair. They argue that each triplet is a subject, predicate and an object set and these parts in the triplets can be used exchangeably in the question and answer.

### 2.2 Text Augmentation

Text augmentation is studied in many areas such as image processing, bioinformatics, and video and audio processing. One of the famous works in data augmentation is related to the study by (Krizhevsky et al., 2014). They tried to classify the images into many classes. They used data augmentation to avoid the problem of over fitting in their neural networks, having 60 million parameters. From a large image, they extracted all the smaller patches and used those patches along with the original image in the training. In addition, for other types of data augmentation, they accompanied the image with different intensities of the original image in the training phase. (Dosovitskiy et. al., 2014) tried data augmentation by first getting different batches from the original data; then they tried different transformations such as different scale, color and contrast on those batches and added them to that class. They trained a convolutional neural network and report higher accuracy using data augmentation. (Bouthillier et. al., 2015) suggested adding data from the same distribution as the original data in the training. They argue that it helps the classifier to have better generalization error. In line with the previous research, (Simard, Steinkraus and Platt, 2003), suggested text augmentation as their best practice in their article. They added different versions of the original data such as the rotated version of the data or random displacement fields to the data for training. They noticed an improvement in the classification error, training their convolutional neural network. In this article, we suggest text augmentation by adding concepts from Conceptnet and Wikidata and descriptions deprived from Wikidata. The detail of these methods is found in the following sections. In addition, we argue that the relations and the concepts in the ConceptNet are not complete and their combination with relation and concepts from Wikidata are more useful and complete for this process. For this purpose, we present Fig.2. It shows the knowledge graphs from Wikidata and ConceptNet; we limit the number of nodes (concepts) to 10 and the number of relations for the purpose of clarity. We chose the word "bitch" because it was the most frequent non-stop-word in our corpus. Figure 1a, the image of the Wikidata knowledge graph around this word, shows the related concepts to which "bitch" is related , "profanity" and "insult", with the relation, IsA. Figure 1b, is a ConceptNet knowledge graph with more relations such as IsA, Synonym, relatedTo and CapableOf to words such as "sugar_baby", "cunt", "canine", "difficulty" and "backbite". Figure 1c. shows the combination of the two knowledge graphs.

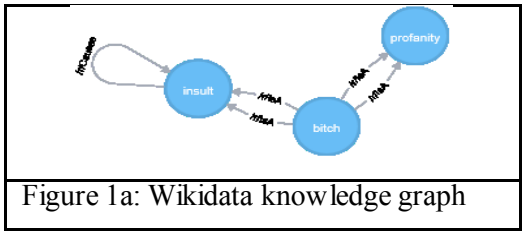

Figure 1a: Wikidata knowledge graph

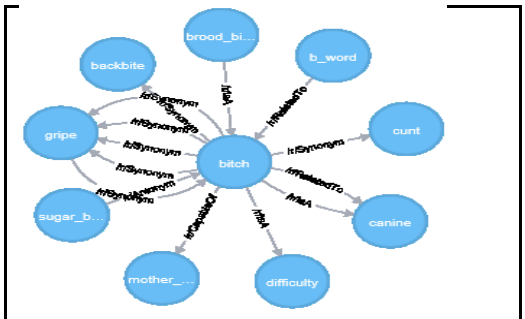

Figure 1b: ConceptNet knowledge graph

Figure 1c: Combination of Wikidata and ConceptNet knowledge graphs.

## 3 Experiment

### 3.1 Dataset

Z. Waseem and D. Hovy, 2016 were the first who collected hateful tweets and categorized them into sexism, racism or neither. Inspired by the study of McIntosh (Peggy McIntosh, 2003), Waseem categorized the tweets into being sexist or racist if they have any of the proposed 18 observations in the tweets such as the usage of any kind of slur to showing sexism racism, criticizing minorities and so forth. Jha and Mamidi 2017 (Jha, A., and Mamidi, R. 2017) solely focused on sexist tweets and proposed two categories of hostile and benevolent sexism. However, these categories were very general and simply ignored other types of sexism happening in social media. In one step further, Sharifirad S. and Matwin S 2018( S sharifirad and S Matwin, 2018), proposed complimentary categories of sexist language inspired from social science work. They categorized sexist tweets into the categories of (1) indirect harassment, (2) information threat, (3) sexual harassment and (4) physical harassment. Table 1 shows the distribution of the dataset along with the sample of tweets in each category mentioned in Table2.

**Table 1: The detail information of the sexist data distribution.**

| Category | Number of tweets |
|---|---|
| **Indirect harassment(#1)** | 260 |
| **Information threat(#2)** | 6 |
| **Sexual harassment(#3)** | 417 |
| **Physical harassment(#4)** | 123 |

**Table 2: Sample of each category**

| category | sample |
|---|---|
| **Indirect harassment** | -'act like a woman think like a man'<br>-'conservative and intelligent women did not take the day off'<br>-'everybody knows that every girl should only want to marry a sane man as good as her sane father nobody can top a girl'<br>- 'i am so sick amp tired of this attitude oh wow youre smart for a girl' |
| **Physical harassment** | -'a womans guide to st century sex naked paintball girls'<br>-'correction katie and nikki are really the dumb blonde ones<br>-'hoping to see the spice girls crash and burn'<br>- 'how can such an ugly girl win a beauty competition she must have been the only one competing that year'<br>- 'i ll never understand why pretty girls let below average guys treat them like shit'<br>-'nobody fucking likes you you ugly stupid fat bitch' |
| **Sexual harassment** | -birch bitch d.ck tosser theres no enough words for him but dead man walking id say forged f..cking v<br>-'bitch shut yall dumbasses up cosigning on bullshit'<br>-chloe and kelly you are a pack of cunts'<br>-'f..ck on that bitch and we lay up'<br>-'caramel girl misionary position naked girls',<br>-caribbean girls getting f..cked best porn shot', |

In this study, we focused on the sexual harassment tweets gathered by (Sharifirad and Matwin, 2018, 2018).

### 3.2 Text preprocessing

Preprocessing of the tweets involves removal of the punctuation, hyperlinks/URLs, emojis and tags. Before training the classification models, Wordnet lemmatization from NLTK library is applied on all the tweets. We set the maximum size of each tweet to 40 words, and padded the tweets of shorter length with zeroes. Next, tweets are converted into vectors using Word2vec (T. Mikolov , 2013), all with the length 300. For the out-of-vocabulary words, we use their character vectors and concatenate them to have the final representation of the word. Classification algorithms For multiclass classification, we considered a baseline along with some traditional classification algorithms utilized for this purpose and deep learning algorithms. We used one-vs-rest (OVR), and trained and evaluated K independent binary classifiers for each class separately. We considered all the samples in that class positive and the rest were all negative samples using LinearSVC in the Sklearn python package. We also considered Support vector Machines (SVM) and Naive Bayes (NB) as the traditional methods and Long-short-term-memory (LSTM) and Convolutional Neural network (CNN) for the choice of deep learning methods (Björn Gambäck and Utpal Kumar Sikdar. 2017 ).

### 3.3 Text Generation

We generated new tweets using ConceptNet in order to improve coverage of our classes using three methods and compared their performance of classification using machine learning models. In the first approach which we call "All Words Replacement (AWR)", tweets were tokenizedand then each token (except for stop words), regardless of its grammatical role, was replaced with all their FromOf and IsA relationships target in ConceptNet whose weight is greater than 1.0. We started from the first token and went forward until a specific number of new tweets had been generated. Relationships other than the ones listed in table 3 led to meaningless tweets that did not represent the original tweet. As an example, the output of Conceptnet for the query of the word "girl" is as follows: [{'relationship': 'IsA', 'target': 'woman', 'weight': 2.0}, {'relationship': 'IsA', 'target': 'female person', 'weight': 1.0}, {'relationship': 'IsA', 'target': 'female young human', 'weight': 1.0}]. We then replaced the word "girl" with the words "women", "female person" and "female young human". The second method which we call "Verb Replacement (VR)" was to first tokenize the tweet and replace the verb by its synonyms in ConceptNet. The third method is called "Noun Replacement (NR)"; the process is the same as with the second approach, VR, but with the difference that we replaced only the nouns with the concepts coming from ConceptNet. Table3 shows the summary of the relation, the selected words and the generated sentence. For each sentence we show only one example out of many of the newly generated tweet. Table 4 shows the result of classification algorithms on the generated tweets.

**Table 3: Sample of generated sentence along with the relation type.**

| | Sentence sample | Types of relation | Generated sentence |
|---|---|---|---|
| **AWR** | "Kathy you **bitch need** to **slap** your **daughter**" | FromOf IsA | "Kathy you **cunt want** to **hit** your **mom**" |
| **VR** | "Kathy you bitch **need** to slap your daughter" | Synonym | Kathy you bitch **want** to smack your daughter |
| **NR** | "Kathy you bitch need to slap your **daughter**" | Synonym Related To | Kathy you bitch need to slap your **mother** |

**Table 4: Classification results on the original and on the generated texts.**

| | OVR | SVM | Naive Bayes | LSTM | CNN |
|---|---|---|---|---|---|
| **The original data** | 0.52 | 0.68 | 0.60 | 0.74 | 0.75 |
| **AWR** | 0.79 | 0.94 | 0.92 | 0.98 | 0.98 |
| **NR** | 0.77 | 0.83 | 0.85 | 0.92 | 0.91 |
| **VR** | 0.82 | 0.88 | 0.88 | 0.97 | 0.95 |

### 3.4 Text augmentation

For text augmentation, we added the concepts from ConceptNet for the first proposed method. In the second method, we considered the concepts from ConceptNet and Wikidata in a smart procedure. The first method is based on adding the related concepts to the original tweets. We tokenized each tweet, then considered "IsA" as the relation and chose the top ten related concepts based on their weight from ConceptNet and added them to the end of the tweet. Even though the number of tweets remained the same the length of the tweets increased to the length of a paragraph. Table 5 shows an example of text augmentation. In the second approach, in addition to the augmentation of tweets using ConceptNet, we augmented tweets by the definition of their tokens in Wikidata. We tokenized the sentence, then added the top related concepts from ConceptNet based on the sorted weight. After that, we combined ConceptNet with Wikidata. The output of the Wikidata around the word query "girls" is 39 tuples; we mention 4 of them as follows: [{'q1.description': 'painting by Lisa Milroy', 'relationship': 'IsA', 'target': 'painting' },{'q1.description': 'painting by Henri-Jean-Guillaume Martin','relationship': 'IsA', 'target': 'painting' }, , {'q1.description': 'young female human', 'relationship': 'IsA', 'target': 'female' }, {'q1.description': 'young female human', etc. Of all these concepts in Wikidata, only one of them is pertaining to the concept "girl" in ConceptNet. To choose the right concept from Wikidata, we first chose the top 10 concepts sorted by the weight, then calculated the cosine similarity between the averaged word vectors of these concepts using Word2vec and the averaged vector of the words in the description from Wikidata. After sorting the descriptions based on the similarity score, we added the most similar description to that tweet.

| Table 5: Sample of text augmentation | |
|---|---|
| **Original tweet** | **Augmented tweet** |
| "local girls near you that are down to fuck rt what links do yall keep clicking on to get hacked" | local girls near you that are down to fuck rt what links do yall keep clicking on to get hacked public transport local organization smaller than national agent non geographical animanga character area unit passive verb feather hair highland strike get better of direction turn soft feather from goose hair feather mood landscape semisolid sexual intercourse rude word television station dehydrated may rehydrated right best human ear good all-purpose life but seeing difference film television show situation software solfa syllable travel create proceed carry through musical artist record confine have store stronghold grow lodge protect stay sound emission communicate destroy make buy return catch annoy touch hit seize get" |

## 4 Results

### 4.1 Experimental Setup

We made bigrams using the python NLTK package and changed them into vectors using word2vec. For word2vec, we used the genism library trained using CBOW and concatenated the vectors of length 300 to get a vector for the each tweet. We used multi-class Naïve Bayes in Scikit learn python, multiclass LSTM and multi-class CNN using Keras for the choice of classifiers. We divided the dataset into 70% train and 30% set. For each tweet, we made the labels in the form of one-hot encoding of length four and we used the same labels for all the classification process. We applied a CNN-based approach to automatically learn and classify sentences into one of the four categories. During the evaluation, a grid search was applied to get the optimal number of filters and filter sizes. Also, we tried with multiple configurations of convolutional layers of 2, 4 and 6. The best performance consisted of two convolutional layers of each followed by a max pooling layer. Convolutional of size 256 with filter size 5 applied for all the convolutional layers. A dropout rate of 0.5 was implemented to avoid over fitting. A fully connected layer with a length of 128 was followed by a second dropout function. This was followed by a dense layer with a size of 4 to represent the number of classification classes using the Softmax function. Our implementation was similar to the model presented in (Björn Gambäck and Utpal Kumar Sikdar. 2017). We trained a simple LSTM model including one hidden layer containing 256 nodes and rectifier activation on the hidden layer, sigmoid activation on the output layer ADAM gradient descent, and a negative likelihood loss function. We created 300 epochs and batch sizes of 5. Table 6 shows the results of the text generation. Our first classification experiment was over the original dataset with three classes, since in the original dataset, the second class, indirect harassment, had only 6 tweets and in comparison to the other classes, it didn't have enough tweets; thus we removed this class and performed our classification algorithm on the rest of three classes. Our second classification approach, verb replacement (VR), was based on the four balanced classes each having about 996 tweets, coming from the first text generation method, all word replacement (AWR). The third classification experiment, noun replacement (NR), was on the four balanced datasets coming from the second method of text generation, each class having about one thousand data points and the last experiment coming from the third approach for text generation; each class having the same number of tweets. We used five classification algorithms, the one-versus-all algorithm as the baseline, naive Bayes and SVM as more traditional classification algorithms and then two artificial neural network approaches, Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN). On the original dataset, the highest accuracy, we achieved was 75% using CNN (for more results please see Table 6). We believe this poor performance was due to poor coverage of our dataset and the imbalanced nature of the dataset. We aimed to alleviate these issues using ConceptNet. Experimenting on the second dataset, which was the generated data with all word replacement (AWR), showed a considerably higher performance in comparison to the original dataset. In this dataset, all four classes are balanced. LSTM and CNN both have the same high performance on the classes. The second performance relates to the SVM and the last one relates to the one-versus-all classification algorithm. In the third generated dataset, noun replacement, the highest performance relates to the LSTM and the second highest relates to the CNN with a very small margin. The highest performance is related to the LSTM for the third method of generated data, followed by CNN and the SVM and Naive Bayes. We ran different text generated methods to know the best way to

increase the number of tweets in each class and to balance it. It seems all word replacement (AWR) of the sentence elements with specific relations from ConceptNet in combination with neural network yields the highest performance boost. As mentioned in table 5. All the generated methods have better performance in comparison to the raw data. VR has better performance in comparison to the NR. The best performance for the text generation method is AWR (all Word Replacement) using ConceptNet as the generation method and LSTM and CNN as the classification method. In addition, all the augmentation methods have better performance in comparison to the original data. The method in which we augment the concepts from Wikidata and ConceptNet along with the description from Wikidata has better performance in comparison to the augmentation with ConceptNet. However, the performance is not as good as the text generation method.

Table 6: Augmentation classification results

| Methods | OVR | SVM | Naive Bayes | LSTM | CNN |
|---|---|---|---|---|---|
| **Augmented with ConceptNet** | 0.55 | 0.65 | 0.64 | 0.90 | 0.88 |
| **Augmentation with ConceptNet and Wikidata** | 0.60 | 0.69 | 0.70 | 0.93 | 0.91 |

## 5 Conclusion

In this article we introduced simple but effective methods for text generation and text augmentation using general purpose knowledge graphs. For text generation we solely used ConceptNet and for text augmentation we used both ConceptNet separately and both ConceptNet and wikidata. Since there is no mapping between ConceptNet and wikidata, we used the cosine similarity of word vectors of related concepts in ConceptNet and words in description of wikidata in order to establish mappings between their concepts. Application of our method to the problem of sexist tweet classification shows drastic improvements in classification results. Our approach can be applied without any modifications to any other text classification problem. As the future work, it is interesting to add words and descriptions from Wikidata for the text augmentation task. We would liketo try this method on other abusive and hate speech datasets. It would also be interesting to combine ConceptNet, Wikidata and Emoji ontology for the text augmentation and text generation task. Also, investigating the impact of the methods on larger datasets are an interesting future work direction.

**References:**


Z. Waseem and D. Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. *In Proceedings of the NAACL Student Research Workshop, pages 88–93. Association for Computational Linguistics.*

Jha, A., and Mamidi, R. 2017. When does a compliment become sexist? analysis and classification of ambivalent sexism using twitter data. *In Proceedings of the Second Workshop on NLP and Computational Social Science*, 7–16.

S sharifirad and S Matwin, 2018. Different types of sexist language on Twitter and the gender footprint, CICLing 2018.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *In ICLR*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. CoRR abs/1609.08144.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. 2016. A hierarchical latent variable encoder-decoder model for generating dialogues. CoRR abs/1605.06069.

Sathish Indurthi , Dinesh Raghu , Mitesh M. Khapra and Sachindra Joshi. 2017. Generating Natural Language Question-Answer Pairs from a Knowledge Graph Using a RNN Based Question Generation Model, *in EACL*.

Uchimoto, K., Sekine, S. & Isahara, H. 2002. Text Generation from Keywords. In Proc. COLING (pp. 1037-1043).

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from over fitting. *The Journal of Machine Learning Research.*

Patrice Y Simard, Dave Steinkraus, and John C Platt. 2003. Best practices for convolutional neural

networks applied to visual document analysis. In Proceedings of the Seventh International Conference on Document Analysis and Recognition-Volume 2, page 958. IEEE Computer Society.

T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. *Accepted to NIPS*.

C. Cortes and V. Vapnik. 1995.Support vector networks. Machine Learning, 20:273–297.

Björn Gambäck and Utpal Kumar Sikdar. 2017. Using Convolutional Neural Networks to Classify Hate-Speech. *In Proceedings of the First Workshop on Abusive Language Online*. 85–90.

X. Lu, B. Zheng, A. Velivelli, and C. Zhai. 2006. Enhancing text categorization with semantic-enriched representation and training data augmentation. *Journal of the American Medical Informatics Association*, 13(5):526–535.

Xu, Q., Qin, Z., Wan, T. 2017. Generative cooperative net for image generation and data augmentation. arXiv preprint arXiv:1705.02887.

Yao, Xuchen and Benjamin Van Durme. 2014. Information extraction over structured data: Question answering with freebase. *In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. Baltimore, Maryland, USA,* pages 956–966.

Voorhees, E. 1994. Query expansion using lexical-semantic relations. *In Proceedings of A CM SIGIR International Conference on Research and Development in Information Retrieval,* pp. 61-69.

Sergio Oramas, Vito Claudio Ostuni, Tommaso Di Noia, Xavier Serra, and Eugenio Di Sciascio. 2017. Sound and Music Recommendation with Knowledge Graphs. ACM TIST 8, 2, 21:1–21:21.

Robert Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. *In Proceedings of AAAI.* San Francisco, USA.

Peggy McIntosh, 2003. Understanding prejudice and discrimination. Chapter White privilege: Unpacking the invisible knapsack, pages 191–196. McGrawHill.

Diederik P. Kingma and Max Welling. 2013. Autoencoding variational bayes. CoRR abs/1312.6114.

Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. 2014. Discriminative unsupervised feature learning with convolutional neural networks. In Advances in Neural Information Processing Systems, pages 766: 774.

K. Konda, X. Bouthillier, R. Memisevic, and P. Vincent,. 2015. Dropout as data augmentation," arXiv preprint arXiv:1506.08700.