A Fuzzy Approach to Hate Speech Classification with Two Stage Training for Ambiguous Instances

Han Liu, *Member, IEEE,* Pete Burnap, *Member, IEEE,* Wafa Alorainy and Matthew L. Williams

***Abstract*—Sentiment analysis is a very popular application area of text mining and machine learning. The popular methods include Support Vector Machine, Naive Bayes, Decision Trees and Deep Neural Networks. However, these methods generally belong to discriminative learning, which aims to distinguish one class from others with a clear-cut outcome, under the presence of ground truth. In the context of text classification, instances are naturally fuzzy (can be multi-labeled in some application areas) and thus are not considered clear-cut, especially given the fact that labels assigned to sentiment in text represent an agreed level of subjective opinion for multiple human annotators rather than indisputable ground truth. This has motivated researchers to develop fuzzy methods, which typically train classifiers through generative learning, i.e. a fuzzy classifier is used to measure the degree to which an instance belongs to each class. Traditional fuzzy methods typically involve generation of a single fuzzy classifier and employ a fixed rule of defuzzification outputting the class with the maximum membership degree. The use of a single fuzzy classifier with the above fixed rule of defuzzification is likely to get the classifier encountering the text ambiguity situation on sentiment data, i.e. an instance may obtain equal membership degrees to both the positive and negative classes. In this paper, we focus on cyberhate classification, since the spread of hate speech via social media can have disruptive impacts on social cohesion and lead to regional and community tensions. Automatic detection of cyberhate has thus become a priority research area. In particular, we propose a modified fuzzy approach with two stage training for dealing with text ambiguity and classifying four types of hate speech, namely: religion, race, disability and sexual orientation - and compare its performance with those popular methods as well as some existing fuzzy approaches, while the features are prepared through the Bag-of-Words and Word Embedding feature extraction methods alongside the correlation based feature subset selection method. The experimental results show that the proposed fuzzy method outperforms the other methods in most cases.**

***Index Terms*—Machine learning, Sentiment analysis, Cyberhate detection, Fuzzy classification.**

I. Introduction

Sentiment analysis is aimed at identifying the attitude or mood of people through natural language processing, text analysis and computational linguistics. In recent years, machine learning has become a very powerful tool for classifying sentiments. In particular, Support Vector Machines (SVM), Naive Bayes (NB), Decision Trees (DT) and its ensemble methods such as Gradient Boosted Trees (GBT) have been used extensively with good performance in broad application areas that involve sentiment analysis, such as cyberbullying detection [1], [2], abusive language detection [3], [4], movie reviews [5], [6] and cyberhate identification [7], [8]. In recent years, deep neural networks (DNN) have also been used for sentiment analysis and other types of text classification.

In the context of machine learning, the above algorithms (SVM, NB, DT, GBT and DNN) are all considered to belong to discriminative learning, since they all aim to distinguish between one class and other classes. In fact, the above algorithms work based on the assumptions that different classes are mutually exclusive and each instance is clear-cut and provided with a ground truth label. However, in the context of text classification, the above assumptions do not always hold, especially when considering the following examples:

In terms of the first assumption, for example, the same movie may belong to different categories, or the same book may belong to different subjects [9], [10], [6]. This example indicates that different classes may not necessarily be mutually exclusive, i.e. different classes could have overlaps, in terms of instances covered by these classes, and the instances can even be multi-labelled in real applications. On the other hand, while different classes are truly mutually exclusive, instances could be very complex and are thus difficult to be classified uniquely to only one category. For example, text such as "I LOVE my country but I HATE immigrants" involves both positive and negative speech [6]. This example indicates that an instance may not be clear-cut, i.e. an instance may partially belong to one class and partially belong to another class. Humans may agree this is hateful but for a discriminative algorithm this poses a challenge.

Furthermore, in sentiment analysis, the label assigned to each instance does not actually represent the ground truth but an agreed representation of the opinion of multiple human annotators, which means that different people may have different opinions about the polarity of a sentiment instance. Thus, sentiment analysis is essentially a task of opinion mining rather than discovery of externally verifiable patterns. The above examples indicate that textual instances are naturally fuzzy and discriminative learning methods are likely to struggle to compute such fuzziness. This has motivated researchers to develop fuzzy methods for text classification, which are able to deal with fuzziness, imprecision and uncertainty of text.

In this paper, we focus on detection of online hate speech (cyberhate) in short informal text posted to social media platforms. This has become a priority research topic due to

Manuscript received xxxx; revised xxxx. The authors acknowledge support for the work presented in this paper from Economic and Social Research Council (Grant ref: ES/P010695/1).

H. Liu, P. Burnap and W. Alorainy are with the School of Computer Science and Informatics, Cardiff University, 5 The Parade, Cardiff, CF24 3AA UK (e-mails: LiuH48@cardiff.ac.uk, BurnapP@cardiff.ac.uk, alorainyws@cardiff.ac.uk).

M. L. Williams is with the School of Social Science, Cardiff University, King Edward VII Ave, Cardiff, CF10 3NN UK (e-mail: WilliamsM7@cardiff.ac.uk).

the concern that the spread of online hate speech could lead to anti-social outcomes [7]. In particular, we deal with four types of online hate speech, namely: religion, race, disability and sexual orientation, by proposing a novel fuzzy approach grounded in generative learning, especially for dealing with text ambiguity, which could result from the following cases: a) the same word may be used in different contexts leading to different semantic meanings; b) that similar instances are assigned different labels by different annotators due to their different opinions. The proposed fuzzy approach is different from existing fuzzy systems in two aspects:

Firstly, traditional fuzzy approaches typically aim at production of single classifiers with specific parameters setting and each single classifier is used independently for a classification task. In this aspect, our proposed fuzzy approach involves fusion (combining the membership degrees for each class) of multiple fuzzy classifiers produced with different parameters setting (e.g. T-norms and T-conorms).

Secondly, traditional fuzzy approaches generally employ a fixed rule (based on maximum membership degree) to provide a distinct class label as an output. In contrast, our proposed fuzzy approach involves a semi-fixed rule of defuzzification, i.e. when an instance obtains the same membership degree (typically a full membership) to both the hate and non-hate classes due to the text ambiguity, the above fixed rule of defuzzification is not suitable, so we introduce a complement rule for classifying the instance based on cosine similarity to other ambiguous instances from the training set.

In both of the two aspects, the proposed fuzzy approach can achieve effective disambiguation of text. Therefore, the bias of a single fuzzy classifier on the majority class (non-hate class) is much reduced, leading to reduction of the false negative rate. In order to evaluate the suitability of the proposed fuzzy approach for cyberhate classification, we compare its performance with the state-of-the-art methods previously used for cyberhate detection (SVM, NB, DT, GBT and DNN), as well as the traditional fuzzy approaches with only a fixed rule of defuzzification through a single fuzzy classifier.

The rest of this paper is organized as follows: Section II describes related work that is relevant to cyberhate research and fuzzy classification; In Section III, we present the proposed fuzzy approach and illustrates the procedure of fuzzy classification. In Section IV, we report an experimental study by using four hate speech data sets collected through Twitter and the results are also presented and discussed. In Section V, we summarize the contributions of this paper and suggest further directions towards advancing this research area.

II. Related Work

This section involves a review of feature extraction methods used for pre-processing of textual data, an overview of cyberhate research in the context of machine learning based text classification and the background of fuzzy text classification in real applications.

A. Review of Feature Extraction Methods

Due to the case that textual data is unstructured, it is necessary to transform textual data into structural data in order to enable the direct use of machine learning algorithms for text classification. This transformation is referred to as feature extraction. In general, there are two popular methods that have been applied in feature extraction for sentiment analysis and cyberhate detection, namely Bag of Words (BOW) and nGrams (NG). Recently probabilistic parse trees was incorporated by [7] through the use of Typed Dependencies (TD). Nowadays, word embedding has become the state-of-the-art method of feature extraction from text.

BOW extracts a bag of distinct words for textual data, and each of the words is used as a feature. In this context, the value of each feature could be binary, which indicates the presence (1) or absence (0) of the word in a textual instance (document). The value of each feature can also be numerical, which indicates the frequency of each word. The following example is given for illustration:

Here are two text instances:

1) Alice encrypts a message using a code and sends the message to Bob.
2) Bob receives the message from Alice and decrypts it using the same code.

Based on the two instances above, a list of distinct words is created: [“Alice, “Bob, “encrypts, “decrypts, “sends, “receives, “message, “a, “the, “and, “it, “from, “to, “using”, “same”, “code”]

Two feature vectors for the two instances are created:

1) $[1, 1, 1, 0, 1, 0, 2, 2, 1, 1, 0, 1, 0, 1, 0, 1]$
2) $[1, 1, 0, 1, 0, 1, 1, 0, 2, 1, 1, 1, 0, 1, 1, 1]$

In the above two feature vectors, each numerical value represents the frequency of a corresponding word.

In general, there are four types of word frequency, namely term absolute frequency (Eq. (1)), term relative frequency (Eq. (2)), inverse document frequency (IDF) (Eq. (3)) and term frequency-inverse document frequency (TF-IDF) (Eq. (4)).

$$tf(t, d) = n_{td} \tag{1}$$

where $n_(t, d)$ represents how many times word t appears in document d.

$$tfr(t, d) = \frac{n_{td}}{\sum_{k=0}^{m} n_{kd}} \tag{2}$$

where $\sum_{k=0}^{m} n_{k,d}$ represents the sum of the absolute frequencies of all the words (words $0 - m$) in document d.

$$idf(t, D) = \log \frac{|D|}{|d \in D : t \in d|} \tag{3}$$

where $|D|$ represents the total number of documents in a corpus D and $|d \in D : t \in d|$ represents the number of documents in which word t appears.

$$tfidf(t, d, D) = tf(t, d) \cdot idf(t, D) \tag{4}$$

For the above example regarding the illustration of BOW, if absolute frequency is used as the value of each feature, then

the same feature vectors would be extracted, since each of the distinct words either appears at least once or does not appear in either one of the two text instances above. More details on BOW can be found at [11].

Although BOW is one of the most popular methods of feature extraction, it has a few limitations that could affect the performance of learning from textual instances. In particular, from semantic perspectives, the same word may have different meanings, which could lead to the case that a word could be highly relevant to the positive class in some cases but also highly relevant to the negative class in other cases. For example, the word 'deserve' can be used to praise students who work hard by saying "You fully deserve the success", whereas the same word can be used to criticize students who failed due to low motivation by saying "That is what you deserve". Also, from syntactic perspectives, the same word may act as different parts of speech. For example, the word "approach" could be both a verb and a noun, which could lead to different abilities to discriminate between classes. In particular, when the above word is used as a verb, it could lead to a negative message such as "I approach you to do something for me". In contrast, when the word is used as a noun, it would generally show a neutral meaning. The above two points indicate that when a word has different meanings or acts as different parts of speech, it is not appropriate to simply treat the word as a single feature.

Due to the limitations of BOW, researchers have been motivated to use NG [12], [13], which is aimed at combining n sequential words as a feature instead of a single word and has led to enrichment of semantic information with improvements of classification performance. In this context, the value of each feature is also represented by different types of frequency, such as corpus frequency (CF), document frequency (DF) and sentence frequency (SF), apart from the commonly used ones (TF, IDF and TF-IDF). In particular, CF represents the frequency of a n-gram in the whole corpus, whereas DF/SF represents the number of documents/sentences in which a n-gram appears. More details on NG and these different types of frequency can be found at [14].

As mentioned in [7], [8], the extraction of NGs as features could result in high levels of distance between words that have correlations. An example given in [15] indicates that related words may appear near the start and near the end of a sentence, which could lead to a negative impact on the performance of learning. In order to improve performance of learning and classification through advancing feature extraction, the Stanford Natural Language Processing Group developed a Lexical Parser for extracting Typed Dependencies [16], which extracts the grammatical relationships between words in a textual instance. The following example [7], [8] is used for illustration of typed dependencies:

Consider the sentence: "Send them all back home". There would be five typed dependencies extracted: [root(ROOT-0, Send-1), nsubj(home-5, them-2), det(home-5, all-3), amod(home-5, back-4), xcomp(Send-1, home-5)]

The second one (nsubj(home-5, them-2)) indicates that there is a relationship between 'home' (the fifth word in the sentence) and 'them' (the second word in the sentence) and the relationship is named as nsubj (which stands for nominal subject). Similarly, the third one (det(home-5, all-3)) represents a determining relationship between 'home' and 'all'. In particular, 'home' acts as a noun phrase and 'all' acts as the determiner of 'home'. In all typed dependencies, word order within a sentence needs to be preserved towards providing features for classification. The use of typed dependencies as features has led to further advances in text classification, comparing with the use of BOW and NG as reported in[8].

BOW, NG and TD can all be generalized to frequency based features referred to as Bag-Of-Terms. In particular, each BOW feature is a single-word term; each NG feature is a multi-word term and each TD feature can be either a single-dependency term or a multi-dependency term. This kind of feature extraction methods could usually lead to high dimensionality and sparsity of the extracted feature vectors [17]. For short text in a small corpus, the above feature extraction methods can even result in extremely sparse feature vectors (0 vectors), due to the presence of all single-word terms of low frequency or absence of multi-word terms after text pre-processing. For example, a tweet that contains only one word does not present NG and TD features leading to the extraction of a zero vector using the NG or TD method. Also, a very short tweet may contain only a few words of low frequency (no greater than 2) that are likely to be excluded for creation of a bag of single word terms. In this case, it is even very unlikely to get high frequency multi-word terms (NGs) or multi-dependency terms (TDs) extracted from such a short tweet, i.e. the use of the NG or TD method is even much more likely to result in the extraction of 0 vectors than the use of the BOW method.

In order to address the dimensionality and sparsity issues, word embedding (Word2vec or Doc2vec) has been used in recent years for transforming words or textual instances directly into feature vectors, through training of deep neural networks. In general, word embedding is aimed at learning numerical representation of words, sentences or even more complex textual instances.

A word vector is represented as $[w, x_1, x_2, ...x_n]$, where w represents a word and x_i is the numerical value of each dimension that represents word w. The word vectors can be used to calculate the semantic distance between features corresponding to the vectors, such as $King - Man = Queen - Woman$ and $UK - London = China - Beijing$. Popular methods of learning word vectors (Word2vec) include continuous bag of words (CBOW), which predicts a target word given context words, and skip-grams, which predicts context words given a target word. Since each textual instance (document) consists of a list of words, the above two Word2vec methods can be extended for training document vectors. The two corresponding Doc2vec methods are referred to as distributed memory (DM) and distributed bag of words (DBOW), respectively. A document vector is represented as $[d, y_1, y_2, ...y_n]$, where d represents a document and y_i is the numerical value of each dimension that represents document d and that is obtained typically by averaging the numerical values of this dimension of all the word vectors in document d.

Due to the significant advantage of word embedding in addressing the issues of high dimensionality and sparsity [18],

the use of document vectors extracted through Doc2vec as features has led to advances in several text classification and sentiment analysis tasks, comparing with the other types of features [19]. More details on feature extraction for cyberhate classification will be given in Section II-B.

B. Overview of Cyberhate Research

Since cyberhate has been considered as a legal issue in many countries, these countries, most of which are located in Europe, have already taken actions against the posting of online hate speech. However, such actions have been complicated due to the case that the World Wide Web is naturally borderless [20]. Also, due to the different laws in different countries, it has become very difficult to prosecute the senders of online hate speech and even being powerless to remove any hateful contents posted from a location outside their territory [21]. Moreover, Signatories of the Council of Europe additional protocol to the Convention on Cybercrime have criminalized acts of racist and xenophobic nature committed through computer systems (16 EU states to date), but tensions remain over balancing freedom of speech with laws that criminalize hate speech. Outside of legal procedures, it has been required to take steps on social network sites towards restricting online hate speech significantly [21]. Researchers have thus been motivated to develop tools for automatic detection of hate speech, in order to manage effectively posts containing hateful contents. In particular, similar to other types of sentiment analysis tasks, machine learning approaches have become very popular for cyberhate detection.

In the context of machine learning based cyberhate detection, various methods of feature extraction and learning algorithms have been used for advancing this area. The NB algorithm was used in [22] for training classifiers on unigram (BOW) features, towards examining each single word for judging whether the tweets were fully hateful or not. In contrast, Mahmud et al focused their work [23] on examining the sentence structures, which tend to be indicative of offensive remarks. Another approach was developed in [24], which aims at assigning a "stereotype sense" to each term in a corpus based on the chance of the term appearing in a hateful post.

In [25], a three level approach of classification was proposed. In particular, a NB classifier is trained first for recognizing the textual patterns of offensive posts, then the most definitive features are identified and selected for feeding into the second level classification by multinomial updatable NB, and finally a decision is made towards classifying instances through the use of a probabilistic rule based system, referred to as Decision Table/Naive Bayes hybrid classifier (DTNB). This three-level approach led to a remarkable 97% accuracy on a data set that consists of messages from the Natural Semantic Module company log files and 1288 Usenet groups messages, but it also led to a lengthier process than other standard learning approaches. In addition, some other traditional learning algorithms, such as SVM [8], Logistic Regression (LR) [26], [27] and Random Forests (RF) [7], have also been used in the previous studies. A pragmatic approach was proposed in [28] for detecting hateful and offensive expressions, based on unigrams and automatically collected patterns for training classifiers by SVM, DT and RF.

In terms of feature extraction, both BOW and NG have been used for cyberhate classification. However, as identified in II-A, both methods have their limitations that could lead to incorrect classifications. In order to improve the classification performance through more effective feature extraction, Burnap et al reported in [7], [8] that 'othering' language could be used as useful features (othering terms), especially for cyberhate classification based on religious beliefs. In particular, the Stanford Lexical Parser was used to extract a bag of Typed Dependencies (TD) within tweets [16], towards capturing potential othering terms. When the extracted Typed Dependencies and hateful terms are combined as features, the experimental results reported in [7], [8] show an improvement of classification performance in the majority of classifiers trained using supervised learning algorithms (e.g. linear SVM and voting based meta learning). The improvement was likely due to the probabilistic nature of the features extracted - i.e. parse trees and assignment of linguistic labels associated with co-occurring terms based on probability (e.g. "us" and "them", or "send" and "home"). TD has also been used in [29], combined with other features (e.g. NG, Word2vec and Doc2vec), leading to advances in classification performance on finance and news data sets when compared with using a single feature set. In [30], various feature extraction methods have been investigated in terms of their effectiveness for processing German text. In particular, BOW, 2-grams, 3-grams, linguistics, Wod2vec, Doc2vec, extended 2-grams and extended 3-grams were used to prepare features for LR to train classifiers. The experimental results, obtained on a 75/25 split between training and test data [30], show that the best performing methods are Word2vec and Extended 2-grams.

In the recent years, deep learning methods have become more popular for both feature extraction and training classifiers. In particular, Convolutional Neural Networks (CNN) have been used in [31] for classifying hate speech with different types of word vectors as features and the results show that Word2vec based feature extraction led to the best performance on a multi-class classification data set [27]. Also, a comparison study was reported in [32] using the same data set. In this study, multiple deep neural networks architectures, such as CNN and Long Short Term Memory (LSTM) Networks, were adopted to learn semantic word embeddings used further for training classifiers. The experimental results reported in [32] show that the use of embedding features led to better classification performance than the use of BOW or NG. The results also show that the classifiers trained by using GBT outperforms the ones trained by using deep neural networks and traditional learning methods (SVM and LR), when using embedding features.

A two-step classification approach was proposed in [3] for detecting racist and sexist speech using the same data set as [27]. In particular, a hybrid CNN architecture was proposed to train features used for detecting abusive language in the first step and then identifying the type of abusive language (racist and sexist) in the second step. The two-step approach was compared with the one-step approach that only involves

identifying if racist or sexist languages are present. The results reported in [3] show that the use of hybrid CNN led to the best performance through the one-step approach, and the use of LR led to the best performance through the two-step approach, but the two-step approach performed marginally worse than the one-step approach. More recently, a gated recurrent unit layer was incorporated into CNN, optimized with dropout and pooling layers, as proposed in [33]. The results show that the proposed approach led to improved performance on 6 out of 7 data sets, compared with the state of the art approaches.

Overall, according to the reviews, SVM, NB and DT are state of the art learning algorithms, which have been used popularly for training classifiers on features extracted from text. Furthermore, DNN and GBT have recently been used for training classifiers on embedding features, which show improvements on the classification performance. In particular, the classifiers trained by DNN performed with the F-measure between 0.80 and 0.84 and the ones trained by GBT performed the F-measure between 0.85 and 0.93, on the data set [27].

C. Background of Fuzzy Text Classification

Fuzzy classification, which is based on fuzzy logic [34], is aimed at dealing with linguistic uncertainty that is involved in instances. In this context, each instance is typically not clear-cut, and thus belongs to different classes to different degrees.

A review of fuzzy approaches made in 2012 for natural language processing [35] indicated that there was a very low percentage of papers relating to fuzzy classification over all the papers published in the area of natural language processing. Also, the review indicated that there was a very low percentage of papers relating to natural language processing over all the application papers published in the area of fuzzy systems. However, the nature of text is its fuzziness, imprecision and uncertainty, which indicates the need of fuzzy approaches.

In recent years, fuzzy approaches have been proposed for a variety of applications. In particular, a fuzzy fingerprint text based approach has been proposed in [36] for classification of companies, which outperformed other popularly used non-fuzzy approaches. Another fuzzy approach was proposed in [37] towards automatically building a corpus that can be used for comparison of text similarity. The experimental results showed that the fuzzy metrics had a higher correlation with human ratings in comparison with other traditional metrics. An unsupervised fuzzy approach was used in [38] towards achieving gender based classification of Twitter users. A three-layer sentiment propagation model was proposed in [39] for determining fuzzy membership degrees for sentiment classification, and the experimental results show that the proposed approach led to reduction of mean squared error (MSE) on seven data sets for sentiment rating prediction, comparing with SVM and other methods of fuzzy membership determination.

A fuzzy rule based approach was proposed in [9] for addressing the issue of model interpretability, and the results showed that the fuzzy approach could lead to a reduction in computational complexity while maintaining a similar performance to other well-known machine learning approaches, such as DT and NB. Furthermore, the use of the above fuzzy rule based approach was investigated in [6] for multi-sentiment analysis (with more than two classes of sentiments), and the results showed that the fuzzy approach could provide more refined outputs by reflecting different intensities of sentiment.

In addition, Dragoni et al proposed fuzzy approaches for exploiting opinion mining in computational advertising [40], undertaking concept-level sentiment analysis [41] and achieving multi-domain sentiment analysis [42]. Crockett et al evaluated the suitability of fuzzy semantic similarity measures for detection of potential future events and the results show that the detection can be achieved by using a group of prototypical event tweets [43]. An automatic approach based on a semantic similarity measure was introduced in [44] for recognizing emotion context from online social networks in the setting of fuzzy classification.

To the best of our knowledge, fuzzy approaches have not been used for cyberhate classification. However, cyberhate data sets are naturally in the form of text, which have similar issues to other sentiment data and thus need the capability of fuzzy approaches in dealing with fuzziness, imprecision, uncertainty for opinion mining. Our proposed fuzzy approach for hate speech classification will be presented in Section III.

III. Fuzzy Rule Based Classification of Cyberhate

In this section, we describe the proposed fuzzy approach for cyberhate classification. In particular, we briefly introduce the theoretical preliminaries of fuzzy logic and rule based systems. The procedure of the proposed fuzzy approach is then illustrated using examples.

A. Theoretical Preliminaries

Fuzzy logic is an extension of deterministic logic, i.e. it employs continuous truth values ranging from 0 to 1, rather than binary truth value (0 or 1). In general, fuzzy logic is involved in a variety of fuzzy theories, such as fuzzy sets and fuzzy rule based systems.

In the context of fuzzy sets, each of the elements $e_1, e_2, ..., e_n$ has a certain degree of membership to the set A, and the membership degree value depends on the membership function μ_A defined for the fuzzy set A, where $\mu_A(e_i)$ indicates the degree of membership of the element e_i in the fuzzy set A, $\mu_A(e_i) \in [0, 1]$ and $1 \leq i \leq n$.

In the context of fuzzy rule based systems, the main operation in the training stage is to fuzzify continuous (numerical) attributes. This can be done by transforming each numerical attribute into several qualitative attributes. For example, 'Age', which is defined as a numerical attribute, can be transformed into three qualitative attributes, namely, 'Young', 'Middle-aged' and 'Old'. Each qualitative attribute is treated as a fuzzy set and is defined with a fuzzy membership function, so the domain of each qualitative attribute must be $[0, 1]$.

Membership functions could be of various shapes, which include trapezoidal, triangular and Gaussian ones. The essence of defining a membership function is to estimate its parameters. For example, a trapezoidal membership function involves four parameters (a, b, c, d) as illustrated in Fig. 1, and the four parameters can determine the membership degree of

a numerical value in a qualitative attribute A as shown below.

$$f_A(x) = \begin{cases} 0, & \text{when } x \leq a \text{ or } x \geq d; \\ (x-a)/(b-a), & \text{when } a < x < b; \\ 1, & \text{when } b \leq x \leq c; \\ (d-x)/(d-c), & \text{when } c < x < d; \end{cases}$$

A trapezoidal membership function can be seen as a generalization of triangular and rectangular membership functions. According to Fig. 1, if $b = c$, then the shape of the membership function would be triangle. Similarly, if $a = b$ and $c = d$, then the shape of the membership function would be rectangle.

According to [45], the fuzzy interval [a, d] is referred to as support region, which indicates a soft boundary for an element to have membership to a set, and the fuzzy interval [b, c] is referred to as core region, which indicates a hard boundary for an element to fully belong to a set.

In real applications, the parameters of membership functions can be estimated by experts [46], or through statistical learning from data [47], [48]. For high dimensional data, the latter way is more needed, which will be described in Section III-B.

In general, a fuzzy rule based system can be represented in the following form:

- Rule 1: if x_1 is A_{11} and x_2 is A_{12} and ... and x_n is A_{1n} then class = C_1;
- Rule 2: if x_1 is A_{21} and x_2 is A_{22} and ... and x_n is A_{2n} then class = C_2;
- $\vdots$
- $\vdots$
- Rule m: if x_1 is A_{m1} and x_2 is A_{m2} and ... and x_n is A_{mn} then class = C_q;

In order to classify a new instance v_i using the above fuzzy rules (r_1, r_2, ... , r_m), it is essential to identify the membership degree of v_i to the fuzzy set A_{tj} in each dimension x_j. Then the firing strength $fs(r_t) \in [0, 1]$ of each rule r_t is computed by combining the membership degrees of v_i to all the fuzzy sets (A_{t1}, A_{t2}, ... , A_{tn}) involved in the antecedents of rule r_t, using a T-norm $T(\cdot) \in [0, 1]$ such as min (Eq. (5)). Furthermore, the overall membership degree for each class C_k is computed by combining the firing strengths of all rules of C_k, using a T-conorm $S(\cdot) \in [0, 1]$ such as max (Eq. (6)). Finally, the new instance is classified by assigning the class with the maximum membership degree.

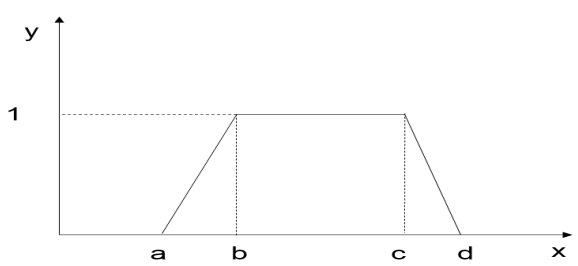

Fig. 1. Trapezoidal Membership Function [9]

$$T(\mu_{A_{t1}}(x_1), \mu_{A_{t2}}(x_2), ..., \mu_{A_{tn}}(x_n)) = \min_{j=1}^{n}\{\mu_{A_{tj}}(x_j)\} \quad (5)$$

$$S(\mu_{A_{t1}}(x_1), \mu_{A_{t2}}(x_2), ..., \mu_{A_{tn}}(x_n)) = \max_{j=1}^{n}\{\mu_{A_{tj}}(x_j)\} \quad (6)$$

$$T_\omega(\mu_{A_{t1}}(x_1), \mu_{A_{t2}}(x_2), ..., \mu_{A_{tn}}(x_n)) = 1 - \min\{1, [\sum_{j=1}^{n}(1 - \mu_{A_{tj}}(x_j))^\omega]^{\frac{1}{\omega}}\} \quad (7)$$

$$S_\omega(\mu_{A_{t1}}(x_1), \mu_{A_{t2}}(x_2), ..., \mu_{A_{tn}}(x_n)) = \min\{1, [\sum_{j=1}^{n}(\mu_{A_{tj}}(x_j))^\omega]^{\frac{1}{\omega}}\} \quad (8)$$

T-norms and T-conorms are jointly referred to as fuzzy norms and some popularly used ones include Min/Max norm [34], Product norm [49], Lukasiewicz norm [50] and Yager norm [51], which will be jointly incorporated into our proposed fuzzy approach. For the Yager Norm, there is a power parameter ω (see Eqs. (7) and (8)). In this paper, the value of the parameter is set to 2 according to the empirical investigation in [49].

B. Fuzzy Approach Methodology

Popular fuzzy rule based systems include Mamdani, Sugeno and Tsukamoto [52]. In general, the design of these traditional fuzzy rule based systems typically depend on predefined membership functions for defuzzification of each of the numerical attributes, i.e. a predefined partitioning of all the numeric attributes, towards discriminating between different classes [53], [45]. This kind of fuzzification would suffer from high dimensional data, such as text. Also, hate speech data is usually very imbalanced, where the minority class is of high importance. In this context, a pre-defined partitioning of each numeric attribute for defining membership functions (fuzzy intervals) is likely to result in the bias of a trained fuzzy classifier on the majority class. Since a predefined partitioning for fuzzifying a numeric attribute typically aims to increase the ability of the fuzzified attribute to discriminate between different classes, in this case, most fuzzified attributes used in a fuzzy classifier would show the tendency to negate rather than identify the minority (target) class, while the training of this classifier is based on selection of the most discriminative attributes. From this point of view, a generative approach of rule learning is needed to more effectively define membership functions for better identifying the target class. Moreover, textual instances are very diverse in terms of their language characteristics, so it is necessary to adopt an instance based approach of fuzzy rule learning, i.e. each instance is checked and a fuzzy rule is added or modified by adjusting the membership functions in some (not necessarily all) dimensions.

Our proposed fuzzy approach for cyberhate detection involves two steps in the training stage, as illustrated in Fig. 2. In this figure, i, c and n represent the index of an instance, a class, and a classifier, respectively. In the first step, a set of

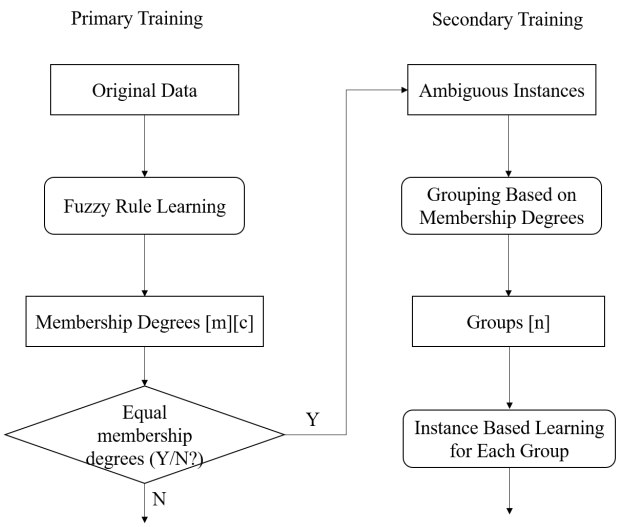

Fig. 2. Learning Framework for Ambiguous Text Classification

fuzzy rules is trained using the mixed fuzzy rule formation algorithm [45], [49].

The procedure of the mixed fuzzy rule formation algorithm involves a sequential and constructive generation of new rules and modification of existing rules in an instance-by-instance manner, i.e. each instance is checked, and a new rule is added into the rule set or some existing rules are modified. In the whole procedure, each rule r_t involves n membership functions (for n dimensions in the rule antecedent part) and two additional parameters w and λ to be defined, where w represents the number of instances covered by rule r_t and λ is a so-called anchor that remembers the original instance triggering the generation of this rule r_t.

After each instance $x_i \in C_k$ is checked, there are three possible cases, namely, covered, committed and shrink. In particular, if the instance x_i lies in the support region covered by an existing rule r_t, i.e. the value v_{ij} of the instance x_i in each dimension j has a non-zero membership degree to the corresponding membership function $\mu_{A_{tj}}$, then it will be judged that the instance x_i is covered by rule r_t. The core region $[b_{tj}, c_{tj}]$ of this rule r_t in each dimension j needs to be adjusted to let the instance x_i lie in the core region in case it does not already. Following the above adjustment, the instance x_i will have a full membership to the rule r_t.

If the instance $x_i \in C_k$ is not covered by any of the existing rules of class C_k in the rule set, then it is judged that the committed case is reached. In this case, a new rule needs to be generated and added into the rule set and the instance x_i is remembered as the anchor λ of this rule. The core region of the new rule in each dimension j is initialized according to the value v_{ij} of the instance x_i in each dimension j, e.g. if $v_{i1} = 2$, then the core region would be initialized as $[2, 2]$. The support region of this new rule in each dimension j is simply initialized to cover the full domain in dimension j, e.g. if the minimum and maximum of the first dimension in the data set are 1 and 5, respectively, the support region of the new rule in this dimension would be $(1, 2)$ or $(2, 5)$.

Once a new rule is added (in the committed case) or an existing rule is modified (in the covered case) after checking an instance $x_i \in C_k$, it is necessary to trigger the third case (shrink) to avoid conflict of classification, i.e. it is to check if there are any existing rules of class $C_l \neq C_k$ that cover the instance x_i (with a non-zero firing strength). If it is the case of conflict, the rules of class $C_l \neq C_k$ need to be adjusted to let the instance x_i have no membership to the rules.

In order to modify a rule r_t of class $C_l \neq C_k$ for avoiding conflict of classification, it is needed to identify whether the instance x_i only lies in the support region (but outside the core region). If so, the rule r_t can be modified without loss of the covered instances that belong to class C_l. In particular, only the dimensions in which the instance x_i falls into the support region of the rule r_t need to be considered for adjusting the membership functions. From these considered dimensions, only the one that results in a minimum loss of volume (Eq. (9)) is chosen for adjusting its membership functions.

The loss of volume (Eq. (10)) can be measured by using some shrink heuristics [49], e.g. rule based shrink (Eqs. (11) and (12)), anchor based shrink (Eqs. (11) and (13)) and border based shrink (Eqs. (11) and (14)).

$$j_{min} = \arg\min_{j=1}^{n}\{V_j\} \tag{9}$$

$$V_j = d_j^*(x_i, r_t) \cdot \prod_{h=1, h\neq j}^{n} d_h^{\times}(x_i, r_t) \tag{10}$$

$$d_j^*(x_i, r_t) = \begin{cases} v_{ij} - a_{tj}, & v_{ij} \leq \lambda_{tj}; \\ d_{tj} - v_{ij}, & \text{otherwise.} \end{cases} \tag{11}$$

$$d_h^{\times}(x_i, r_t) = d_{tj} - a_{tj} \tag{12}$$

$$d_j^{\times}(x_i, r_t) = \begin{cases} \lambda_{tj} - a_{tj}, & v_{ij} \leq \lambda_{tj}; \\ d_{tj} - \lambda_{tj}, & \text{otherwise.} \end{cases} \tag{13}$$

$$d_j^{\times}(x_i, r_t) = \begin{cases} b_{tj} - a_{tj}, & v_{ij} \leq \lambda_{tj}; \\ d_{tj} - c_{tj}, & \text{otherwise.} \end{cases} \tag{14}$$

In the above equations, j_{min} is the index of the dimension that leads to the minimum loss of volume; V_j is the actual amount of the volume loss; $d_j^*(\cdot)$ is the volume loss function and $d_j^{\times}(\cdot)$ is the function weighting the loss of volume; v_{ij} is the value of x_i in dimension j; λ_{tj} is the value of the anchor of rule r_t in dimension j; a_{tj} and d_{tj} are the left and right boundaries of the support region of rule r_t in dimension j, respectively; b_{tj} and c_{tj} are the left and right boundaries of the core region of rule r_t in dimension j.

When the instance $x_i \in C_k$ lies in the core region of a rule r_t of class $C_l \leq C_k$, it is not possible to modify the rule without loss of some covered instances that belong to C_l, since both the support and core regions of this rule need to be adjusted, such that the instance x_i has no membership to the rule r_t any more. In this case, the functions ($d_j^*(\cdot)$ and $d_j^{\times}(\cdot)$) for measuring and weighting the volume loss and the shrink heuristics need to be modified as shown in Eqs. (15) - (17).

In particular, rule based shrink and border based shrink are identical as shown in Eq. (16).

$$d_j^*(x_i, r_t) = \begin{cases} v_{ij} - b_{tj}, & v_{ij} \leq \lambda_{tj}; \\ c_{tj} - v_{ij}, & \text{otherwise.} \end{cases} \quad (15)$$

$$d_h^{\times}(x_i, r_t) = c_{tj} - b_{tj} \quad (16)$$

$$d_j^{\times}(x_i, r_t) = \begin{cases} \lambda_{tj} - b_{tj}, & v_{ij} \leq \lambda_{tj}; \\ c_{tj} - \lambda_{tj}, & \text{otherwise.} \end{cases} \quad (17)$$

TABLE I
EXAMPLE OF FUZZY RULES

Rule	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5	Feature 6	Feature 7	Class
1	[0,0,0,2.9]	[0,0,2.2,2.2]	[0,0,0,4.0]	[0,0,0,4.1]	[0,0,0,3.4]	[0,0,0,4.8]	[0,0,0,3.1]	No
2	[0,2.9,2.9,2.9]	[0,0,0,0]	[0,0,0,0]	[0,0,0,0]	[0,0,0,0]	[0,0,0,0]	[0,3.1,3.1,3.1]	Yes
3	[0,0,0,2.9]	[0,0,0,2.2]	[0,0,0,4.0]	[0,0,0,4.1]	[0,0,0,3.4]	[0,0,0,4.8]	[0,0,0,0]	Yes
4	[0,0,0,2.9]	[0,0,0,0]	[0,0,0,0]	[0,0,0,0]	[0,3.4,3.4,3.4]	[0,0,0,0]	[0,0,0,0]	No
5	[0,2.9,2.9,2.9]	[0,0,0,0]	[0,0,0,4.0]	[0,0,0,0]	[0,0,0,4.0]	[0,0,0,0]	[0,0,0,3.1]	No
6	[0,0,0,2.9]	[0,0,0,0]	[0,4.0,4.0,4.0]	[0,0,0,0]	[0,0,0,0]	[0,0,0,0]	[0,0,0,0]	No
7	[0,0,0,2.9]	[0,0,0,0]	[0,0,0,0]	[0,4.1,4.1,4.1]	[0,0,0,0]	[0,0,0,0]	[0,0,0,0]	No
8	[0,0,2.9,2.9]	[0,0,0,0]	[0,4.0,4.0,4.0]	[0,0,0,0]	[0,0,0,0]	[0,0,0,0]	[0,0,0,0]	Yes
9	[0,0,2.9,2.9]	[0,0,0,0]	[0,0,0,0]	[0,0,0,0]	[0,3.4,3.4,3.4]	[0,0,0,0]	[0,0,0,0]	Yes
10	[0,2.9,2.9,2.9]	[0,0,0,0]	[0,0,0,0]	[0,0,0,0]	[0,0,0,0]	[0,0,0,0]	[0,0,0,0]	Yes
11	[0,0,0,0]	[0,0,0,0]	[0,0,0,0]	[0,4.1,4.1,4.1]	[0,0,0,0]	[0,0,0,0]	[0,0,0,0]	Yes

Following the above procedure of the mixed fuzzy rule formation algorithm, a set of fuzzy rules are trained and an example of the fuzzy rules for cyberhate detection is shown in Table I. In particular, there are seven terms used as features for identifying if a tweet is hate speech, and the seven terms are implicitly represented in this table using the seven headers 'Feature 1', 'Feature 2',..., and 'Feature 7', due to ethical reasons. For each rule, a membership function $[a, b, c, d]$ is defined for each dimension (feature) involved in the rule antecedent part, where a, b, c and d represent the lower bound of the support region, the lower bound of the core region, the upper bound of the core region, and the upper bound of the support region, respectively.

In the testing stage, a new instance is classified through fuzzification, inference and defuzzification. The fuzzification operation is simply aimed at mapping the numeric value of each feature of the new instance into a membership degree to each rule in each dimension. The inference operation is adopted to compute the firing strength of each rule by using a T-norm and to derive the overall membership degree for each class by using a T-conorm. The defuzziciation operation is to finally classify the new instance by assigning it the class with the maximum membership degree.

Due to the text ambiguity, it is possible that an instance could obtain equal membership degrees for the hate and non-hate classes, which leads to a likely increase in error when classifying unseen instances through a fixed rule (choosing the class with the maximum membership degree). In this case, we propose to train multiple fuzzy classifiers using different fuzzy norms to encourage diversity between these fuzzy classifiers, i.e. these fuzzy classifiers lead to different sets of ambiguous instances, due to the fact that different fuzzy norms have different impacts on training fuzzy classifiers through the mixed fuzzy rule formation algorithm [49]. Therefore, the fusion of these fuzzy classifiers is likely to reduce the number of ambiguous instances in the training stage and has the chance to disambiguate an unseen instance that obtains equal membership degrees to both classes in the testing stage. The fusion can be achieved through averaging the overall membership degrees obtained from these fuzzy classifiers for each class. In Fig. 2, the membership degree of each instance i to each class c is checked to identify if the instance is still ambiguous. If so, the instance will be sent to the next stage for instance based learning or instance based reasoning, depending on whether it is a training instance or a test instance. In this case, all the training instances sent to the instance based learning stage are collated to form the second training set.

However, the fusion of fuzzy classifiers can not guarantee that all the ambiguous instances are disambiguated, so the second step is required to collate all the remaining ambiguous instances and produce a new training set. Using the second training set, a complement rule is trained for classifying ambiguous unseen instances.

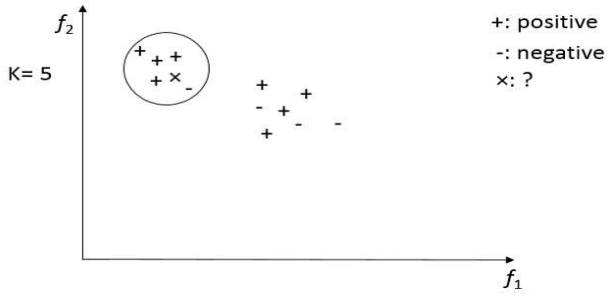

Fig. 3. Example of Instance Based Reasoning [54]

The second training set, in this instance, included all the cases from the training phase to which equal membership to both classes was assigned. We therefore undertook an instance based learning approach based on cosine similarity (see Eq. (18)) and K nearest neighbours (KNN). This works by gathering all ambiguous instances in multi-dimensional vector space (i.e. based on the features derived from the text) and plotting a new ambiguous instance in the same feature space. The class label of the 'nearest neighbours', i.e. the features that are most closely matched to those of the unseen instance, is assigned to the unseen instance. For example, as shown in Fig 3, there are 5 previously labelled instances selected as the nearest neighbours to the unseen instance - so sharing the most similar features - and the majority (4 of them) belong to the positive class vs. 1 negative, so the unseen instance is assigned the positive class. In this context, it is necessary to determine how many nearest neighbours (usually an odd number to avoid ties) will be used for classifying unseen instances and whether class imbalance handling is needed. These are dependent on the data sets being used and will be discussed in Section IV.

$$Score = cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \quad (18)$$

Cosine similarity is a measure of similarity between two non-zero vectors, which is efficient especially for evaluation of sparse (but non-zero) vectors, since only the dimensions with non-zero values for both vectors need to be evaluated.

In practice, cosine similarity has been popularly used in information retrieval and text mining [55]. In Eq. (18), A and B represent two non-zero vectors and i is expressed as the index of a dimension (feature).

Overall, the proposed fuzzy approach involves two main stages. In the first stage, multiple fuzzy classifiers are trained using the mixed fuzzy rule formation algorithm alongside different fuzzy norms, and the fuzzy classifiers are then fused to identify ambiguous instances. In the second stage, the ambiguous instances are collated to produce the second training set for using KNN to classify new instances that are ambiguous. The performance of the proposed fuzzy approach and impacts of fuzzy classifiers fusion and instance based reasoning are evaluated in Section IV.

IV. Experimental Setup, Results and Discussion

In this section, we conduct an experimental study on cyber-hate classification. In particular, we use four data sets collected from Twitter, regarding four types of hate speech (religion, race, disability and sexual orientation). From each data set, four different sets of features are prepared by using BOW and Doc2vec, respectively, for feature extraction and using the correlation based feature subset selection method [56] for feature selection. Also, the fuzzy approach proposed in Section III is used for training classifiers, and its performance is compared with DT, NB, SVM, GBT, DNN as well as the existing fuzzy approaches, in terms of precision, recall and F-measure for the 'hate' class.

A. Data

The data sets used for the study of online hate speech were collected from Twitter for a period immediately following selected 'trigger' events, which were: for religion, the attack on Lee Rigby in Woolwich, London on 22 May 2013 by Islamist Extremists; for race, the presidential re-election of Barack Obama starting November 6th 2012; for sexual orientation, the public announcement by Jason Collins on 30th April 2013 - the first active athlete in an American professional sports team to come out as gay; and for disability, the opening ceremony of the Paralympic games in London, UK on 29th August 2012.

Data collection used search terms based on named entities that were the focus of the events, i.e. 'woolwich', 'obama', 'paralympic', 'jason collins'. These terms would include many references to the events and the main hashtags surrounding the event e.g. '# paralympics'. Each event produced datasets between 300,000 and 1.2 million, from which we randomly sampled 2,000 to be human coded. Coders were provided with each tweet and the question: 'is this text offensive or antagonistic in terms of religion/race/sexual orientation/disability?' They were presented with a ternary set of classes - yes, no, undecided. We required at least four human annotations per tweet as per the convention in related research [57]. Based on the annotation results, we can determine the agreement rate of human coders on each tweet. In particular, we removed all tweets with less than 75 percent agreement and also those upon which the coders could reach an absolute decision (i.e., the 'undecided' class), as suggested in [58].

The results of the annotation exercise produced four 'gold standard' data sets as follows: Religion - 1,901 tweets, with 222 instances of offensive or antagonistic content (11.68% of the annotated sample); Race - 1,876 tweets, with 70 instances of offensive or antagonistic content (3.73% of the annotated sample); Disability - 1,914 tweets, with 51 instances of offensive or antagonistic content (2.66% of the annotated sample); and Sexual Orientation - 1,803 tweets, with 183 instances of offensive or antagonistic content (10.15% of the annotated sample). The proportion of instances of offensive or antagonistic content, which we refer to after this point as cyber hate, is small relative to the size of the sample. However, these are random samples of the full datasets for each event and are therefore representative of the overall levels of cyber hate within the corpus of tweets.

B. Experimental Design and Results

The experimental study is divided into two parts. The first part is aimed at investigating the impacts of fusion of fuzzy classifiers trained using different fuzzy norms and instance based reasoning through KNN on text disambiguation, i.e. it is to evaluate the effectiveness of reducing the number of ambiguous instances and advancing the classification performance. The second part is aimed at further evaluation of the proposed fuzzy approach in terms of its classification performance in comparison with the state-of-the-art probabilistic approaches. For both parts of the study, the experimental design involves text preprocessing, feature extraction, feature selection and classifiers training.

In terms of text processing, each word was converted to lowercase, and stop words, punctuation, numbers were removed. Then the remaining words were stemmed using the Snowball stemmer. Furthermore, the frequencies (TF and IDF) of each word was calculated prior to extraction of a bag of words as the features. Only the features that meet the predefined minimum absolute term frequency (Eq. (1)) are selected for training classifiers, and TF-IDF is used as the value of each feature. The minimum term frequency is generally set to 2, but the frequency is set to 5 for the religion data set to avoid over high dimensionality (with a large number of non-hateful terms), since the data set contains a relatively larger number of hateful instances and terms than the other data sets. Following the above procedure, a set of n feature vectors is extracted from each data set of n tweets.

NG and TD are not adopted for feature extraction in this study, since the use of these methods is likely to result in a large number of zero vectors extracted from very short text in a small corpus (as pointed out in Section II-A), which would affect the effectiveness of direct comparison of the impacts of different feature extraction methods. In other words, zero vectors are generally not useful for training classifiers, and the inclusion of a large number of such vectors in the feature set would really lead to negative impacts on the training of classifiers. In this case, the obtained classification performance can not fairly reflect the effectiveness of the used feature extraction method, while the text does not present this kind of features resulting from the method.

For Doc2vec feature extraction, the text preprocessing is operated in the same procedure as the one involved in BOW feature extraction, but the word stemming step is excluded to avoid loss of the semantic information for context identification. Furthermore, distributed bag of words (DBOW) was used to learn representations of words and documents (tweet text) by predicting the context words given each target word, and the learning rate was set to 0.025 with the context window size of 2 (2 left context words and 2 right context words around the target word) to deal more effectively with short text in a small corpus. All words (to be transformed into vectors) needed to meet the minimum absolute term frequency (Eq. (1)) of 2. Following the training of embeddings with the batch size (number of words used for each batch) of 10000 over 50 epochs, all the n tweets in each data set were finally transformed into n document vectors with 100 dimensions (features).

TABLE II
DIMENSIONALITY OF EACH FEATURE SET

Feature set	Religion	Race	Disability	Sexual Orientation
BOW(full)	501	1237	1368	1226
BOW(sub)	17	13	7	35
Doc2vec(full)	100	100	100	100
Doc2vec(sub)	23	15	7	21

For each of the two feature sets extracted from each data set using BOW and Doc2vec, respectively, the correlation based feature subset selection method is adopted to reduce the dimensionality leading to two smaller feature sets. Therefore, there are totally four feature sets prepared for each data set and the details on the dimensionality of these feature sets are shown in Table II. In this table, BOW(full) represents that a full set of features extracted using BOW is used for training classifiers, whereas BOW(sub) represents that a subset of selected BOW features is used. The same also applies to the embeddings extracted through Doc2vec to distinguish the case of using a full feature set or a subset of selected features.

In the machine learning stage, DT classifiers were trained using the C4.5 algorithm alongside the Reduced Error Pruning (REP) method. SVM classifiers were trained using the linear kernel. For GBT training, all the attributes were used and attribute selection for each node of a tree was done using the same set of attributes. For DNN training, in order to suit better smaller data sets, we chose a network architecture that consists of two fully connected layers and 100 units in each layer. The classifiers were trained using the mean squared error (MSE) loss function and the rectified linear unit (ReLU) activation function with the learning rate of 0.01, and Stochastic Gradient Descent was used for optimizing the parameters of DNNs over 20 epochs with the batch size of 500.

In terms of fuzzy rule based classification, we selected the Min/Max norm, Product norm, Lukasiewicz norm and Yager norm, respectively, alongside the border based shrink heuristic as the parameters (based on the empirical investigation in [49] of their influences on training classifiers), for defining trapezoid membership functions in all dimensions (at the same level of granularity) and training four fuzzy classifiers. The trained fuzzy classifiers are then fused by averaging the membership degrees (obtained from these classifiers) of each instance for each class. However, as mentioned in Section III, our proposed approach involves a trained rule to complement the fixed rule for the defuzzification step, so any instances that obtain equal membership degrees to both classes would need to be classified by adopting the trained rule (instance based reasoning), instead of the fixed rule. In addition, an instance may also obtain no membership to either one of the two classes, due to the diversity of textual instances. However, this case rarely occurs from our data sets so our experiment is set to classify such instances simply to the 'no' class.

In terms of instance based reasoning through KNN, the SMOTE oversampling (based on the five closest instances) is adopted to deal with the class imbalance issue [59] without loss of information (which can result from undersampling that randomly delete some instances of the majority class). The value of K is generally set to 5, unless the instance based reasoning stage is not necessary due to the case that no ambiguous instance appears after fuzzy classifiers learning and fusion. In this case, the K value is set to 0, which indicates that KNN is not used in the whole procedure.

TABLE III
FUZZY CLASSIFICATION PERFORMANCE ON HATE SPEECH DATA

Method		Religion			Race			Disability			Sexual Orientation		
Feature	Learning	P	R	F	P	R	F	P	R	F	P	R	F
BOW(full)	Fuzzy1.1	0.779	0.491	0.602	0.644	0.543	0.589	0.536	0.588	**0.561**	0.655	0.311	0.422
	Fuzzy1.2	0.743	0.495	0.595	0.694	0.486	0.571	0.491	0.549	0.519	0.581	0.333	0.424
	Fuzzy1.3	0.772	0.505	0.61	0.654	0.486	0.557	0.5	0.569	0.532	0.626	0.339	0.44
	Fuzzy1.4	0.767	0.518	0.618	0.694	0.486	0.571	0.517	0.588	0.55	0.642	0.333	0.439
	Fuzzy2	0.771	0.532	0.629	0.649	0.529	0.583	0.527	0.569	0.547	0.597	0.404	0.482
	Fuzzy3	0.679	0.667	**0.673**	0.627	0.671	**0.648**	0.517	0.608	0.559	0.519	0.525	**0.522**
BOW(sub)	Fuzzy1.1	0.986	0.324	0.488	0.857	0.343	0.49	1	0.569	0.725	0.89	0.355	0.508
	Fuzzy1.2	0.987	0.333	0.498	0.931	0.386	0.545	1	0.569	0.725	0.929	0.355	0.514
	Fuzzy1.3	0.987	0.342	0.508	0.923	0.343	0.5	1	0.549	0.709	0.915	0.355	0.512
	Fuzzy1.4	0.987	0.342	0.508	0.929	0.371	0.531	1	0.569	0.725	0.926	0.344	0.502
	Fuzzy2	0.988	0.383	0.552	0.889	0.457	0.604	1	0.569	0.725	0.878	0.393	0.543
	Fuzzy3	0.878	0.617	**0.725**	0.877	0.714	**0.787**	0.912	0.608	**0.729**	0.875	0.536	**0.664**
Doc2vec(full)	Fuzzy1.1	0.694	0.387	0.497	0.88	0.314	0.463	0.906	0.569	0.699	0.451	0.404	0.427
	Fuzzy1.2	0.712	0.356	0.474	0.8	0.343	0.48	0.933	0.549	0.691	0.435	0.399	0.416
	Fuzzy1.3	1	0.185	0.312	0.913	0.3	0.452	0.933	0.549	0.691	0.433	0.23	0.3
	Fuzzy1.4	0.931	0.243	0.386	0.917	0.314	0.468	0.933	0.549	0.691	0.456	0.284	0.35
	Fuzzy2	0.778	0.378	0.509	0.926	0.357	0.515	0.967	0.569	0.716	0.744	0.333	0.46
	Fuzzy3	0.748	0.401	**0.522**	0.933	0.4	**0.56**	0.969	0.608	**0.747**	0.688	0.35	**0.464**
Doc2vec(sub)	Fuzzy1.1	0.577	0.356	0.44	0.756	0.443	0.559	0.935	0.569	0.707	0.449	0.454	0.451
	Fuzzy1.2	0.656	0.369	0.473	0.778	0.4	0.528	0.853	0.569	0.682	0.432	0.432	0.432
	Fuzzy1.3	0.982	0.243	0.39	0.893	0.357	0.51	0.935	0.569	0.707	0.442	0.29	0.35
	Fuzzy1.4	0.804	0.333	0.471	0.737	0.4	0.519	0.879	0.569	0.69	0.468	0.404	0.434
	Fuzzy2	0.839	0.351	0.495	0.861	0.443	**0.585**	0.935	0.569	0.707	0.848	0.366	0.511
	Fuzzy3	0.835	0.365	**0.508**	0.861	0.443	**0.585**	0.939	0.608	**0.738**	0.789	0.388	**0.52**

For the first part of the experimental study, the results on fuzzy classification are shown in Table III. In this table, Fuzzy1.1, Fuzzy1.2, Fuzzy1.3 and Fuzzy1.4 represents that the four fuzzy norms (Min/Max norm, Product norm, Lukasiewicz norm and Yager norm) are used, respectively, for training fuzzy classifiers. Fuzzy2 represent the case of fusion of the four fuzzy classifiers trained using the above four fuzzy norms. Fuzzy3 represents our proposed fuzzy approach that involves fuzzy2 for classifiers fusion and KNN for instance based reasoning through the use of cosine similarity.

The results shown in Table III indicate that the fusion of fuzzy classifiers trained using different fuzzy norms generally leads to effective advances in the classification performance and the adoption of KNN for instance based reasoning even leads to further advances in the performance following the fusion of fuzzy classifiers in most cases. For the selected subset of features extracted from the Race data set, Fuzzy3 performs the same as Fuzzy2, since there is no ambiguous instance remaining after the fusion of the fuzzy classifiers, as shown in Table IV. In other words, Fuzzy2 has successfully led to reducing the number of ambiguous instances to 0, so there is no space for Fuzzy3 to lead to further improvements.

TABLE IV
NUMBER OF AMBIGUOUS INSTANCES

Feature	Learning	Religion		Race		Disability		Sexual Orientation	
		#Total	#Hate	#Total	#Hate	#Total	#Hate	#Total	#Hate
BOW(full)	Fuzzy1.1	212	53	414	10	73	4	393	69
	Fuzzy1.2	204	55	403	11	64	1	361	62
	Fuzzy1.3	201	52	406	11	65	5	357	60
	Fuzzy1.4	194	46	403	11	68	4	349	60
	Fuzzy2	169	40	311	10	56	3	249	40
BOW(sub)	Fuzzy1.1	1263	150	1661	46	1805	20	1031	113
	Fuzzy1.2	1261	148	1660	43	1805	20	1039	115
	Fuzzy1.3	1259	146	1663	46	1805	21	1039	114
	Fuzzy1.4	1259	146	1661	44	1805	20	1041	117
	Fuzzy2	1250	137	1653	38	1804	20	1023	107
Doc2vec(full)	Fuzzy1.1	25	11	5	5	2	2	83	15
	Fuzzy1.2	12	4	3	3	3	3	80	12
	Fuzzy1.3	5	2	4	4	3	3	71	9
	Fuzzy1.4	5	2	4	3	3	3	72	11
	Fuzzy2	19	8	3	3	2	2	83	15
Doc2vec(sub)	Fuzzy1.1	16	5	5	3	3	3	84	15
	Fuzzy1.2	21	10	5	3	3	3	81	12
	Fuzzy1.3	6	2	3	3	2	2	73	11
	Fuzzy1.4	8	4	3	3	2	2	77	12
	Fuzzy2	11	6	0	0	2	2	67	14

The results shown in Table IV indicates that the fusion of fuzzy classifiers generally leads to reduction of the number of ambiguous instances. In the worst case, the number of ambiguous instances is not higher than the largest number of ambiguous instances resulting from a single fuzzy classifier.

When BOW is used for feature extraction, the further feature selection leads to a significant increase of the number of ambiguous instances, which provides more space for the instance based reasoning part of Fuzzy3 to lead to a larger improvement of the performance, in comparison with the use of a full feature set. The increase of the number of ambiguous instances is likely due to the case that the dimentionality reduction results in the increase of the firing strength of each rule. For example, when the number of dimensions is largely reduced, the minimum membership degree obtained in a dimension in the rule antecedent part is likely to become larger using the Min function of T-norm (Eq. 5).

Since the adopted correlation based feature subset selection is essentially aimed to reduce the feature-to-feature correlation but increase the feature-to-class correlation, the selected features are considered as highly relevant to the classes and the removed features are considered as redundant or irrelevant. In this context, if an instance obtains a high membership degree in one feature dimension, then it would be likely to obtain a high membership degree in the other correlated feature dimensions. Through using T-norm, the firing strength of a rule is no greater than the minimum of the membership degrees in the feature dimensions in the rule antecedent part. Also, the inclusion of the irrelevant features in some dimensions in the rule antecedent part would lead to a further decrease of the the firing strength of this rule. In this case, while the redundant and irrelevant features are removed, the firing strength of a rule would be likely to get increased. However, textual instances of different classes may have some common words as relevant features, which leads to the case that an instance become ambiguous when the instance contains such common words.

When Doc2vec is used for feature extraction, the adoption of feature selection leads to a similar number of ambiguous instances in comparison with the use of a full set of features. In this case, the performance of Fuzzy3 is generally similar when using the full feature set or a subset of selected features, except for the sexual orientation data set.

For evaluation of the fuzzy classification performance, we conduct statistical analysis through the Wilcoxon rank tests [60] to identify whether the difference between our proposed fuzzy approach and the other ones is statistically significant. In particular, the comparison of different fuzzy approaches is made on the basis of each feature set prepared for each data set, i.e. there is totally 16 (4×4) feature sets. The results shown in Table V indicate that the adoption of Fuzzy3 leads to significant advances in the classification performance.

TABLE V
RANK TESTS FOR PERFORMANCE OF FUZZY CLASSIFICATION

Compared methods	p-value	Null Hypothesis
Fuzzy1.1 vs Fuzzy3	0	Reject
Fuzzy1.2 vs Fuzzy3	0	Reject
Fuzzy1.3 vs Fuzzy3	0	Reject
Fuzzy1.4 vs Fuzzy3	0	Reject
Fuzzy2 vs Fuzzy3	0	Reject

We also apply the Wilcoxon rank tests to identify the impacts of different feature sets on the performance of Fuzzy3. The results are shown in Table VI, which indicate that applying feature selection to the set of BOW features leads to significantly better performance of Fuzzy3, in comparison with the use of the full set of BOW features. Furthermore, the use of the subset of selected BOW features leads to better performance of Fuzzy3 in comparison with the use of the other two feature sets, but the performance difference is somewhat less than statistically significant (p-value=0.065).

TABLE VI
RANK TESTS FOR IMPACTS OF FEATURE SETS ON FUZZY CLASSIFICATION

Compared methods	p-value	Null Hypothesis
BOW(full) vs BOW(sub)	0.029	Reject
Doc2vec(full) vs Doc2vec(sub)	0.225	Accept
BOW(full) vs Doc2vec(full)	0.353	Accept
BOW(sub) vs Doc2vec(sub)	0.065	Accept
BOW(full) vs Doc2vec(sub)	0.353	Accept
BOW(sub) vs Doc2vec(full)	0.065	Accept

In addition, we conduct complexity analysis (Table VII) in terms of the number of fuzzy rules generated using different fuzzy norms for classifiers training. We also identify the impacts of different feature sets on the number of fuzzy rules.

TABLE VII
NUMBER OF FUZZY RULES

Feature	Learning	Religion	Race	Disability	Sexual Orientation
BOW(full)	Fuzzy1.1	173	107	107	261
	Fuzzy1.2	188	107	101	236
	Fuzzy1.3	184	122	85	267
	Fuzzy1.4	189	109	104	246
BOW(sub)	Fuzzy1.1	11	18	11	35
	Fuzzy1.2	11	18	11	33
	Fuzzy1.3	11	18	11	32
	Fuzzy1.4	11	18	11	33
Doc2vec(full)	Fuzzy1.1	68	41	24	68
	Fuzzy1.2	72	38	28	67
	Fuzzy1.3	847	212	66	515
	Fuzzy1.4	429	74	45	272
Doc2vec(sub)	Fuzzy1.1	112	54	39	103
	Fuzzy1.2	115	50	38	109
	Fuzzy1.3	413	112	62	290
	Fuzzy1.4	250	66	45	157

When BOW is used for feature extraction, the adoption of feature selection consistently leads to significant reduction of the number of fuzzy rules when using different fuzzy norms. For both the full set of BOW features and a subset of selected ones, the use of different fuzzy norms lead to similar complexity of the trained fuzzy classifiers, i.e. the numbers of rules generated using different fuzzy norms are similar.

When Doc2vec is used for feature extraction, the adoption of feature selection consistently leads to an increase of the number of rules when using the Min/Max norm and the Product norm, whereas the number of rules is consistently reduced when using the other two fuzzy norms. Moreover, the number of rules generated using the Min/Max norm or the product norm is consistently much lower than the number of rules generated using any one of the other two fuzzy norms. This phenomenon indicates that the use of embedding features is likely to result in diverse impacts of using different fuzzy norms on the complexity of the trained fuzzy classifiers.

Overall, the adoption of feature selection would generally lead to the increase of the interpretability of fuzzy classifiers, due to the reduction of the dimensionality, especially for the fuzzy classifiers trained on the subsets of selected BOW features, where the number of rules is also significantly reduced.

For the second part of the experimental study, the results on fuzzy classification are shown in Table VIII. In this table, 'N/A' indicates that the corresponding classifier never outputs the hate class, i.e. it is fully biased on the non-hate class.

TABLE VIII
COMPARISON WITH PROBABILISTIC APPROACHES ON CLASSIFICATION PERFORMANCE

Method		Religion			Race			Disability			Sexual Orientation		
Feature	Learning	P	R	F	P	R	F	P	R	F	P	R	F
BOW(full)	DT	0.802	0.658	0.723	0.741	0.571	0.645	0.833	0.49	**0.617**	0.792	0.415	0.545
	NB	0.444	0.626	0.52	0.336	0.529	0.411	0.596	0.549	0.571	0.448	0.475	0.462
	SVM	0.808	0.532	0.641	0.849	0.643	**0.732**	0.682	0.294	0.411	0.635	0.219	0.325
	GBT	0.863	0.653	**0.744**	0.78	0.657	0.713	0.596	0.608	0.602	0.884	0.415	**0.565**
	DNN	N/A	0	N/A	N/A	0	N/A	N/A	0	N/A	N/A	0	N/A
	Fuzzy3	0.679	0.667	0.673	0.627	0.671	0.648	0.517	0.608	0.559	0.519	0.525	0.522
BOW(sub)	DT	0.855	0.613	0.714	0.812	0.557	0.661	0.931	0.529	0.675	0.871	0.443	0.587
	NB	0.85	0.662	**0.744**	0.592	0.871	0.705	0.654	0.667	0.66	0.739	0.557	0.636
	SVM	0.854	0.608	0.711	0.857	0.6	0.706	0.882	0.588	0.706	0.868	0.503	0.637
	GBT	0.871	0.608	0.716	0.849	0.643	0.732	0.933	0.549	0.691	0.903	0.459	0.609
	DNN	0.847	0.5	0.629	0.833	0.571	0.678	0.903	0.549	0.683	0.811	0.164	0.273
	Fuzzy3	0.878	0.617	0.725	0.877	0.714	**0.787**	0.912	0.608	**0.729**	0.875	0.536	**0.664**
Doc2vec(full)	DT	0.455	0.383	0.416	0.365	0.271	0.311	0.806	0.569	0.667	0.385	0.377	0.381
	NB	0.516	0.374	0.433	0.109	0.814	0.193	0.192	0.647	0.296	0.507	0.186	0.272
	SVM	0.705	0.333	0.473	0.706	0.171	0.276	1	0.451	0.622	0.704	0.311	0.432
	GBT	0.821	0.351	0.492	0.711	0.386	0.5	0.763	0.569	0.652	0.701	0.333	0.452
	DNN	N/A	0	N/A	N/A	0	N/A	N/A	0	N/A	N/A	0	N/A
	Fuzzy3	0.748	0.401	**0.522**	0.933	0.4	**0.56**	0.969	0.608	**0.747**	0.688	0.35	**0.464**
Doc2vec(sub)	DT	0.481	0.41	0.443	0.349	0.214	0.265	0.659	0.529	0.587	0.438	0.383	0.408
	NB	0.424	0.563	0.484	0.103	0.714	0.18	0.157	0.627	0.251	0.417	0.437	0.427
	SVM	N/A	0	N/A	N/A	0	N/A	1.00	0.451	0.622	0.636	0.077	0.137
	GBT	0.721	0.36	0.48	0.722	0.371	0.491	0.784	0.569	0.659	0.595	0.361	0.449
	DNN	0.667	0.045	0.084	N/A	0	N/A	N/A	0	N/A	0.611	0.12	0.201
	Fuzzy3	0.835	0.365	**0.508**	0.861	0.443	**0.585**	0.939	0.608	**0.738**	0.789	0.388	**0.52**

The results shown in Table VIII indicates that Fuzzy3 outperforms the other approaches in most cases. In particular, when the full set of BOW features is used, the performance of Fuzzy3 is generally worse than the other approaches, but the adoption of feature selection leads to significant advances in the performance of Fuzzy3, which is better than the performance of other approaches, except for the Religion data set, where Fuzzy3 performs slightly worse than NB. When Doc2vec is used for feature extraction, the use of Fuzzy3 consistently shows better performance than the use of other approaches on both the full feature set or a subset of selected embedding features prepared for each data set.

Furthermore, we apply the Wilcoxon rank tests again to identify the significance level of the difference between the performance of Fuzzy3 and the one of the other approaches.

TABLE IX
RANK TESTS FOR PERFORMANCE OF LEARNING METHODS

Compared methods	p-value	Null Hypothesis
DT vs Fuzzy3	0.002	Reject
NB vs Fuzzy3	0	Reject
SVM vs Fuzzy3	0.001	Reject
GBT vs Fuzzy3	0.049	Reject
DNN vs Fuzzy3	0	Reject

The comparison of different learning approaches is made on the same basis as the statistical analysis conducted for comparing the performance of different fuzzy approaches, i.e. there is totally 16 (4×4) feature sets. The results (Table IX) indicate that the use of Fuzzy3 leads to significant better performance of classification than the use of the other approaches.

Overall, the results indicate that the proposed fuzzy approach can effectively dealing with the text ambiguity issue, which overcomes the limitations of using a single fuzzy classifier, leading to considerable advances in the classification performance through fusion of multiple fuzzy classifiers trained using different fuzzy norms and instance based reasoning by KNN based on cosine similarity. Also, the proposed fuzzy approach involves diverse ways of dealing with fuzziness in text, which overcomes the limitations of probabilistic approaches assuming that each instance is clear-cut.

V. CONCLUSION

In this paper, we proposed a modified fuzzy approach for cyberhate classification. In particular, we argued that fuzzy approaches are more suitable than previously used non-fuzzy approaches that are known to perform well on hate speech data, due to the advantages of fuzzy approaches in dealing with fuzziness, imprecision and uncertainty of text. For example, fuzzy approaches are capable of providing more refined outputs by reflecting different intensities of sentiments, which is effective for detecting any ambiguous instances (i.e. checking if they obtain the same membership degrees for the two classes), so that people can be aware that further analysis of such text in more depth is necessary through fusion of multiple fuzzy classifiers and instance based reasoning.

We conducted experiments using four data sets on four types of hate speech, namely: religion, race, disability and sexual orientation. In particular, we compared the performance of the proposed fuzzy approach with the one of leading discriminative approaches to cyber hate classification (i.e. DT, NB, SVM, GBT and DNN) as well as the traditional fuzzy approaches with a fixed rule of defuzzification through a single fuzzy classifier. Also, we prepared various feature sets using two feature extraction methods alongside a feature selection method, for the purpose of evaluating the impacts of different ways of feature preparation on training fuzzy classifiers. The experimental results show that the proposed fuzzy approach outperforms all other methods in most cases and leads to a considerable improvement on the classification performance. We discussed the likelihood that the improvement is likely due to the ability of fuzzy classifiers fusion combined with KNN in dealing with fuzziness and ambiguity of text.

In future, we will aim to develop larger data sets towards increasing the text diversity, so it will be more likely to detect various cases of text ambiguity and the proposed fuzzy approach with two stage training will be investigated more broadly by exploring how to get the ambiguous instances into different groups towards in-depth disambiguation in the instance based learning step. Also, we will investigate the impact of the combination of different types of hate speech sample on the performance of training fuzzy classifiers. In this context, there is intersectionality between different types of hate speech so we will aim to explore whether the intersectionality can result in extraction of more diverse features for hate speech detection, leading to better performance of fuzzy classification. In addition, we will investigate the use of fuzzy approaches for identifying the context and topic of hate speech to better understand its use and motivations.

REFERENCES

[1] R. Zhao, A. Zhou, and K. Mao, "Automatic detection of cyberbullying on social networks based on bullying features," in *17th International Conference on Distributed Computing and Networking*, 4-7 January 2016.

[2] K. Reynolds, A. Kontostathis, and L. Edwards, "Using machine learning to detect cyberbullying," in *10th International Conference on Machine Learning and Applications*, Honolulu, HI, USA, 18-21 December 2011, pp. 241–244.

[3] J. H. Park and P. Fung, "One-step and two-step classification for abusive language detection on twitter," in *1st Workshop on Abusive Language Online*, Vancouver, Canada, 4 August 2017, pp. 41–45.

[4] H. Mubarak, K. Darwish, and W. Magdy, "Abusive language detection on arabic social media," in *1st Workshop on Abusive Language Online*, Vancouver, Canada, 4 August 2017, pp. 52–56.

[5] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? sentiment classification using machine learning techniques," in *ACL-02 Conference on Empirical Methods in Natural Language Processing*, 2002, pp. 79–86.

[6] C. Jefferson, H. Liu, and M. Cocea, "Fuzzy approach for sentiment analysis," in *IEEE International Conference on Fuzzy Systems*, Naples, Italy, 9-12 July 2017, pp. 1–6.

[7] P. Burnap and M. Williams, "Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making," *Policy and Internet*, vol. 7, no. 2, pp. 223–242, 2015.

[8] ——, "Us and them: identifying cyber hate on twitter across multiple protected characteristics," *EPJ Data Science*, vol. 5, no. 11, 2016.

[9] H. Liu and M. Cocea, "Fuzzy rule based systems for interpretable sentiment analysis," in *International Conference on Advanced Computational Intelligence*, Doha, Qatar, 4-6 February 2017, pp. 129–136.

[10] ——, "Fuzzy information granulation towards interpretable sentiment analysis," *Granular Computing*, vol. 2, no. 4, pp. 289–302, 2017.

[11] J. Sivic, "Efficient visual search of videos cast as text retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 4, pp. 591–605, 2009.

[12] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau, "Sentiment analysis of twitter data," in *Workshop on Languages in Social Media*, Stroudsburg, PA, USA, 23 June 2011, pp. 30–38.

[13] N. Pendar, "Toward spotting the pedophile telling victim from predator in text chats," in *International Conference on Semantic Computing*, Irvine, CA, USA, 17-19 September 2007, pp. 235–241.

[14] K. Thiel and M. Berthold, "The knime text processing feature: An introduction," KNIME, Tech. Rep., 2012.

[15] Y. Chen, Y. Zhou, S. Zhu, and H. Xu, "Detecting offensive language in social media to protect adolescent online safety," in *2012 ASE/IEEE International Conference on Social Computing*, Amsterdam, Netherland, 3-5 September 2012, pp. 71–80.

[16] M.-C. de Marneffe, B. MacCartney, and C. D. Manning, "Generating typed dependency parses from phrase structure parses," in *5th International Conference on Language Resources and Evaluation*, Genoa, Italy, 27-28 May 2006, pp. 71–80.

[17] V. Tursi and R. Silipo, *From Words to Wisdom. An Introduction to Text Mining with KNIME*. Zurich, Switzerland: KNIME Press, 2018.

[18] N. Djuric, J. Zhou, R. Morris, M. Grbovic, V. Radosavljevic, and N. Bhamidipati, "Hate speech detection with comment embeddings," in *24th International Conference on World Wide Web*. ACM, 18-22 May 2015, pp. 29–30.

[19] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *31st International Conference on Machine Learning*. Beijing, China: Springer, 21-26 June 2014, pp. II–1188–II–1196.

[20] I. Nemes, "Regulating hate speech in cyberspace: Issues of desirability and efficacy," *Journal of Information and Communications Technology Law*, vol. 11, no. 3, pp. 193–220, 2002.

[21] J. Banks, "Regulating hate speech online," *International Review of Law, Computers and Technology*, vol. 24, no. 3, pp. 233–239, 2010.

[22] I. Kwok and Y. Wang, "Locate the hate: Detecting tweets against blacks," in *27th AAAI Conference on Artificial Intelligence*, Bellevue, Washington, 14-18 July 2013, pp. 1621–1622.

[23] A. Mahmud, K. Z. Ahmed, and M. Khan, "Detecting flames and insults in text," in *6th International Conference on Natural Language Processing*, Gothenburg, Sweden, 25-27 August 2008.

[24] W. Warner and J. Hirschberg, "Detecting hate speech on the world wide web," in *2nd Workshop on Language in Social Media*, Montreal, Canada, 07 June 2012, pp. 19–26.

[25] A. H. Razavi, D. Inkpen, S. Uritsky, and S. Matwin, "Offensive language detection using multi-level classification," in *23rd Canadian conference on Advances in Artificial Intelligence*, Ottawa, Canada, 31 May-2 June 2010, pp. 16–27.

[26] G. Xiang, B. Fan, L. Wang, J. Hong, and C. Rose, "Detecting offensive tweets via topical feature discovery over a large scale twitter corpus," in *21st ACM international conference on Information and knowledge management*. Maui, Hawaii, USA: Springer, 29 October-2 November 2012, pp. 1980–1984.

[27] Z. Waseem and D. Hovy, "Hateful symbols or hateful people? predictive features for hate speech detection on twitter," in *Proceedings of NAACL-HLT 2016*, San Diego, California, USA, 12-17 June 2016, pp. 88–93.

[28] H. Watanabe, M. Bouazizi, and T. Ohtsuki, "Hate speech on twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection," *IEEE Access*, vol. PP, no. 99, pp. 1–11, 2018.

[29] C. Nobata, J. Tetreault, and A. Thomas, "Abusive language detection in online user content," in *25th International Conference on World Wide Web*, Montral, Qubec, Canada, 11-15 April 2016, pp. 145–153.

[30] K. Sebastian, R. D. M, H. Steffen, and B. Jrg, "Discussing the value of automatic hate speech detection in online debates," in *Multikonferenz Wirtschaftsinformatik*, Leuphana, Germany, 6-9 March 2018, pp. 83–94.

[31] B. Gambck and U. K. Sikdar, "Using convolutional neural networks to classify hate-speech," in *1st Workshop on Abusive Language Online*, Vancouver, Canada, 4 August 2017, pp. 85–90.

[32] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, "Deep learning for hate speech detection in tweets," in *26th International Conference on World Wide Web Companion*, Perth, Australia, 3-7 April 2017, pp. 759–760.

[33] Z. Zhang, D. Robinson, and J. Tepper, "Detecting hate speech on twitter using a convolution-gru based deep neural network," in *European Semantic Web Conference*, Heraklion, Crete, Greece, 3-7 June 2018, pp. 745–760.

[34] L. Zadeh, "Fuzzy logic: A personal perspective," *Fuzzy Sets and Systems*, vol. 281, pp. 4–20, 2015.

[35] J. P. Carvalho, F. Batista, and L. Coheur, "A critical survey on the use of fuzzy sets in speech and natural language processing," in *IEEE International Conference on Fuzzy Systems*, Brisbane, QLD, Australia, 10-15 June 2012.

[36] F. Batista and J. P. Carvalho, "Text based classification of companies in crunchbase," in *IEEE International Conference on Fuzzy Systems*, Istanbul, Turkey, 2-5 August 2015.

[37] D. Chandran, K. A. Crockett, D. Mclean, and A. Crispin, "An automatic corpus based method for a building multiple fuzzy word dataset," in *IEEE International Conference on Fuzzy Systems*, Istanbul, Turkey, 2-5 August 2015.

[38] M. Vicente, F. Batista, and J. P. Carvalho, "Twitter gender classification using user unstructured information," in *IEEE International Conference on Fuzzy Systems*, Istanbul, Turkey, 2-5 August 2015.

[39] C. Zhao, S. Wang, and D. Li, "Determining fuzzy membership for sentiment classification: A three-layer sentiment propagation model," *PLoS ONE*, vol. 11, no. 11, 2016.

[40] M. Dragoni, "A three-phase approach for exploiting opinion mining in computational advertising," *IEEE Intelligent Systems*, vol. 32, no. 3, pp. 21–27, 2017.

[41] M. Dragoni, A. G. B. Tettamanzi, and C. da Costa Pereira, "Propagating and aggregating fuzzy polarities for concept-level sentiment analysis," *Cognitive Computation*, vol. 7, no. 2, pp. 186–197, 2015.
[42] M. Dragoni and G. Petruccia, "A fuzzy-based strategy for multi-domain sentiment analysis," *International Journal of Approximate Reasoning*, vol. 93, pp. 59–73, 2018.
[43] K. Crockett, N. Adel, J. O'Shea, A. Crispin, D. Chandran, and J. P. Carvalho, "Application of fuzzy semantic similarity measures to event detection within tweets," in *IEEE International Conference on Fuzzy Systems*, Naples, Italy, 9-12 July 2017, pp. 1–7.
[44] I. B. Sassi, S. B. Yahia, and S. Mellouli, "Fuzzy classification-based emotional context recognition from online social networks messages," in *IEEE International Conference on Fuzzy Systems*, Naples, Italy, 9-12 July 2017, pp. 1–6.
[45] M. R. Berthold, "Mixed fuzzy rule formation," *International Journal of Approximate Reasoning*, vol. 32, no. 2-3, pp. 67–84, 2003.
[46] E. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Human-Computer Studies*, vol. 51, no. 2, pp. 135–147, 1999.
[47] S.-M. Chen, "A fuzzy reasoning approach for rule-based systems based on fuzzy logics," *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics*, vol. 26, no. 5, pp. 769–778, 1996.
[48] F. Bergadano and V. Cutello, "Learning membership functions," in *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, Granada, Spain, 8-10 November 1993, pp. 25–32.
[49] T. R. Gabriel and M. R. Berthold, "Influence of fuzzy norms and other heuristics on mixed fuzzy rule formation," *International Journal of Approximate Reasoning*, vol. 35, no. 2, pp. 195–202, 2004.
[50] J. Lukasiewicz, *Selected Works-Studies in Logic and the Foundations of Mathematics*. Amsterdam: North-Holland Publishing, 1970.
[51] R. R. Yager, S. Ovchinnikov, R. M. Tong, and H. T. Ngugen, *Fuzzy Sets and Applications*. New York: Wiley, 1987.
[52] T. Ross, *Fuzzy Logic with Engineering Applications*. West Sussex: Wiley, 2010.
[53] L.-X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 22, no. 6, pp. 1414–1427, 1992.
[54] H. Liu and M. Cocea, *Granular Computing Based Machine Learning: A Big Data Processing Approach*. Berlin: Springer, 2018.
[55] T. Korenius, J. Laurikkala, and M. Juhola, "On principal component analysis, cosine and euclidean measures in information retrieval," *Information Sciences*, vol. 177, no. 22, pp. 4893–4905, 2007.
[56] M. A. Hall and L. A. Smith, "Feature subset selection: a correlation based filter approach," in *1997 International Conference on Neural Information Processing and Intelligent Information Systems*. Berlin, Germany: Springer, 1997, pp. 855–858.
[57] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas, "Sentiment strength detection in short informal text," *Journal of the Association for Information Science and Technology*, vol. 61, no. 12, pp. 2544–2558, 2010.
[58] M. Thelwall, D. Wilkinson, and S. Uppal, "Data mining emotion in social network communication: Gender differences in myspace," *Journal of the Association for Information Science and Technology*, vol. 61, no. 1, pp. 190–199, 2010.
[59] A. Ali, S. M. Shamsuddin, and A. L. Ralescu, "Classification with class imbalance problem: A review," *International Journal of Advanced Soft Computing Applications*, vol. 7, no. 3, pp. 176–204, 2015.
[60] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine learning research*, vol. 7, pp. 1–30, 2006.

Han Liu (S'15-M'16) received his BSc in Computing from University of Portsmouth in 2011, an MSc in Software Engineering from University of Southampton in 2012, and a PhD in Machine Learning from University of Portsmouth in 2015. His research interests include data mining, machine learning, rule based systems, intelligent systems, fuzzy systems, pattern recognition, big data, granular computing and computational intelligence. He has published two research monographs in Springer and over 50 papers in the areas such as data mining, machine learning and intelligent systems. One of his papers was identified as a key scientific article contributing to scientific and engineering research excellence by the selection team at Advances in Engineering and the selection rate is less than 0.1%. He also has two papers selected, respectively, as finalists of Lotfi Zadeh Best Paper Award in the 16th and 17th International Conference on Machine Learning and Cybernetics (ICMLC 2017 & 2018). He is currently a Research Associate in Data Science and a member of the HateLab (hatelab.net) and the Social Data Science Lab (socialdatalab.net) in the School of Computer Science and Informatics at the Cardiff University. He has previously been a Research Associate in Computational Intelligence in the School of Computing at the University of Portsmouth. He is a member of the Institution of Engineering and Technology (IET).

Pete Burnap is Professor of Data Science & Cybersecurity at Cardiff University. He is Director of the Social Data Science Lab and Deputy-director of HateLab at Cardiff University - both funded by the UK Economic and Social Research Council. He has published extensively on the development of machine learning approaches to classifying and understanding the production and propagation of crime and cybersecurity-related content on social media

Wafa Alorainy is a PhD student in the School of Computer Science and Informatics at the Cardiff university. She is a member of the Social Data Science Lab (socialdatalab.net), a part of the ESRC Big Data Network. Her research interests include text mining, social media analysis and machine learning applications.

Matthew Williams is Professor of Criminology at the School of Social Sciences, Cardiff University. He is the Director of HateLab (hatelab.net) and the Social Data Science Lab (socialdatalab.net), both part of the ESRC Big Data Network. He has published extensively on the use of social media data in crime and security research.