N-Gram-Based Recognition of Threatening Tweets

Nelleke Oostdijk and Hans van Halteren

Radboud University Nijmegen, CLS – Dept. of Linguistics / CLST
{N.Oostdijk,hvh}@let.ru.nl

Abstract. In this paper, we investigate to what degree it is possible to recognize threats in Dutch tweets. We attempt threat recognition on the basis of only the single tweet (without further context) and using only very simple recognition features, namely n-grams. We present two different methods of n-gram-based recognition, one based on manually constructed n-gram patterns and the other on machine learned patterns. Our evaluation is not restricted to precision and recall scores, but also looks into the difference in yield of the two methods, considering either combination or means that may help refine both methods individually.

Keywords: social media, text mining, text classification, manually constructed rules, machine learning.

1 Introduction

In recent years the microblogging service Twitter has gained immense popularity. Estimates are that in the Netherlands alone each day over 3 million tweets are posted. The very short 140-character messages are primarily used for sharing information on what is going on right there and then. However, as journalists, policy makers, businesses, marketing agencies etc. have been quick to discover, the collective information has also great potential when it comes to finding out about things that are about to happen or that have only just taken place, and what the prevailing sentiments are. For searching and retrieving information and for sentiment mining, existing NLP techniques are being deployed rather successfully.

However, there is also a dark side of the internet as in the perceived anonymity of the medium people are being bullied, harassed, and even threatened with violence. As acts of intimidation, harassment, and other forms of threatening are criminal offences punishable by law, law enforcement agencies are under pressure to develop a policy for dealing with these phenomena.[1] A possible course of action could be to monitor the internet so that immediate action can be taken when a threat is made. Such a task becomes only feasible when tools are available that will support it.

In the present study we investigate, for Dutch tweets, whether on the basis of the content of a single tweet (without further context) we can detect automatically whether

[1] See also http://www.rcmp-grc.gc.ca/qc/pub/cybercrime/cybercrime-eng.htm

A. Gelbukh (Ed.): CICLing 2013, Part II, LNCS 7817, pp. 183–196, 2013.

it contains a threat. This task is quite hard, as threats cannot be detected simply by means of a set of keywords or phrases.

For the present study we adopt the following working definition of what constitutes a threat:

> A threat is a declaration of an intention to cause death or bodily harm to a person or persons, to damage or destroy their personal property, or to kill or injure an animal that is the property of a person.[2]

Under this definition tweets that are intended to annoy, alarm or otherwise cause emotional distress to another person are not considered to hold a threat. Also verbal abuse of another person or persons does not by itself constitute a threat.

The recognition of what constitutes a real threat is especially difficult as there are numerous tweets containing riddles or jokes, or where people are being sarcastic or ironic (so that it would immediately be clear to someone that what was being said was not to be taken seriously). Other tweets where a threat is not normally taken seriously is where a tweet clearly refers to for example a game setting, a movie, or a soap series.

Recognizing a threat is all the more difficult as in a language like Dutch there are numerous expressions which hint at harm or violence, but which are generally understood as figures of speech (e.g. *je kunt doodvallen* ('drop dead'), *op sterven na dood* ('almost dead'), *rijp voor de sloop* ('ready to be demolished': 'written off')). Moreover, many words are ambiguous and only point towards a threat in particular contexts. For example, a word like *maken* ('to make') is mostly neutral, also when it occurs as part of a separable verb (e.g. *opmaken* ('to format'), *doormaken* ('to go through')). However, when it occurs as part of the verb *afmaken* it may be neutral (as in *huiswerk afmaken* ('to finish homework')) or threatening (as in *jou afmaken* ('to finish you off')).

In the present paper we investigate two approaches that might be employed for the task of automatically detecting threats in Dutch tweets. In the first approach we attempt to manually construct a set of n-grams that should detect threats. In the second approach, we use machine learning to discover which (surface) features characterize threats. The task is defined as a classification task in which the two approaches each attempt to classify tweets as either threatening or non-threatening, depending on whether or not they contain a threat. The approaches are evaluated and compared for efficacy but also so as to see how one approach might advance the other.

The structure of the paper is as follows. In Section 2 we describe the data used for development and testing. A description of the manual construction of the n-grams is given in Section 3, while in Section 4 the machine learning approach is described. A quantitative analysis of the test results is given in Section 5. The two approaches are compared qualitatively in Section 6. Section 7 concludes this paper.

[2] Note that we are not looking for a legal definition, but rather for a definition that can be operationalized when attempting to identify what constitutes a threat when dealing with tweets. The definition is rather loosely based on that given in Black's Law Dictionary [1] and the Canadian Criminal Code [2].

2 Experimental Data

For our experiments we need data representing threats and also data representing non-threats. Although with Twitter large amounts of data are available, we do not know which tweets are threatening. Therefore we decided to use large random samples of data as background corpus for development and for measuring precision. For the positive examples (used for development and measuring coverage) specialized collections are needed.

2.1 Collection of Dutch Twitter Threats

Threatening tweets were obtained from the website `www.doodsbedreiging.nl`, a site which allegedly wants to raise a public debate on the phenomenon of threats made through Twitter.[3] Over the past two years or so the site has published over 5,000 threats that were posted on Twitter. We downloaded two data sets, viz. one that we used as development set and the other that we held apart and used as test set. As we found that not all downloaded tweets answered to our definition of what constitutes a threatening tweet, all data was checked manually and non-threats were removed. As a result in the (threat) development set (henceforth TDS) 4,564 tweets remain, while the (threat) test set (TTS) comprises 583 tweets. The TTS fully postdates and has no overlap with the TDS.

Data clean-up for both data sets involved the removal of collection artifacts such as the hash tag #doodsbedreiging, retweet markers (rt, RT etc.), time stamps and user names (@username). Moreover, in the development set proper names and URLs were anonymized so as to avoid recognizing regular targets (such as the controversial politician Geert Wilders) rather than the threat itself. Subsequently all data were tokenized: punctuation marks were separated from the word tokens and all upper case characters were converted to lower case. Complexes of punctuation marks and symbols, probably meant as emoticons, were not broken up into parts.

2.2 Samples of Dutch Twitter in General

For a large random sample of general tweets to be used as development set, we extracted some 2.3 million tweets, viz. the tweets from a single day in 2011, from a much larger set of Dutch Twitter data collected through the Dutch e-science centre [3]. As in the collection process a language filter was applied, the data contains virtually no dialect or street language, which we do find in the data from `www.doodsbedreiging.nl`. As test set, a random set of 1 million tweets was sampled from the same collection, with time stamps between October 2011 and September 2012. In what follows we refer to the general development set as the GDS and to the general test set as the GTS.

[3] Cf. the editorial on `www.doodsbedreiging.nl`

3 Manually Constructed Recognition Patterns

In the first of two approaches we want to compare, we use a set of manually constructed recognition patterns. Here we rely on our (linguistic) intuition as native speakers of Dutch. In the process, the development sets (TDS and GDS) are used for further inspiration and for obtaining more objective information as to how frequently certain patterns occur and with what senses.

The set of patterns consists of (token)[4] n-grams, more specifically positive and negative unigrams, bigrams, trigrams, and skipgrams (bigrams and trigrams). By definition, the tokens in bigrams and trigrams are adjacent while in skip bigrams they are non-adjacent. In a skip trigram, however, one of three situations may arise: (1) the first two tokens are adjacent, while the third is non-adjacent to the second, (2) the last two tokens are adjacent, while the first is non-adjacent to the second, or (3) the three tokens are all non-adjacent. There is no differentiation in pattern strength.

The total number of base n-grams[5] is 16,190. Of these 3,129 are positive and 13,061 negative. The distribution over the different n-gram types is given in Table 1.

Table 1. Characterization of the base n-gram set: distribution of n-gram types.The labels used are as follows: <NG1>=unigram, <NG2>=bigram, <NG3>=trigram, <SG2>=skip bigram, <SG3as>=skip trigram with only the first two tokens adjacent, <SG3sa>=skip trigram with only the last two token adjacent, <SG3ss>=skip trigram with only non-adjacent tokens.

n-gram type	positive	negative
<NG1>	304	--
<NG2>	831	1190
<NG3>	519	2875
<SG2>	709	201
<SG3as>	277	2944
<SG3sa>	299	2938
<SG3ss>	190	2913

3.1 N-Grams Expected in Threatening Tweets

The manual patterns focus on the recognition of phrasings that overtly express a threat. Therefore, most positive n-grams contain an action verb that is indicative of some violent action. Examples are *doden* (‘to kill’), *(neer)steken* (‘to stab’), *vermoorden* (‘to murder’) and *(neer/af/dood)schieten* (‘to shoot’). As threats typically refer to something happening in the near or not too distant future - such as that the sender of the tweet is going to inflict harm upon the receiver or, put differently, the receiver is

[4] Tokens are words, numbers, punctuation marks, or symbols.

[5] Base n-grams are expressed using conventional spelling, with the exception of spelling variants involving different spacing in words (cf. note 5). See also Section 4.3 which describes how spelling variation is handled.

going to experience something bad happening to him - the verb form commonly is first or second person present tense or future.[6] Examples can be found in the unigrams <snijd> ('cut'), <schiet> (shoot) and <djoek> ('kill') and the bigrams <ik vermoord> ('I kill') and <gaat sterven> ('are going to die').

As the n-grams are token-based and no part-of-speech information can be brought to bear to disambiguate between homographs of, for example, a noun and a verb (*dood*, 'death'/'kill'), or a present tense verb form and a past participle (*vermoord*, 'kill'/'killed'), the unigrams are likely to overgenerate. Therefore, in many such cases we have opted to use a (skip) bigram rather than a unigram (<ik dood> ('I kill') and <ik vermoord> ('I murder')).[7]

The large proportion of n-grams that are not unigrams can further be explained by the fact that in Dutch there are many separable verbs (e.g. *doodsteken* ('to stab to death'), for which the first person present tense is *steek dood*) and there is a frequent use of subject-verb inversion (so that apart from the bigram <ik vermoord> we also need to specify the inverse <vermoord ik>).

3.2 N-Grams Inhibiting Erroneous Recognition

Negative n-grams are brought into play in order to delimit the extent to which the positive n-grams are overgenerating. Thus where the unigram <aanval> ('attack') will yield a great many false accepts including *hart aanval* ('heart attack'), *paniek aanval* ('panic attack'), *schijn aanval* ('mock attack'), the inclusion of such instances as negative n-grams effectively cancels them out.[8]

While there are quite a few cases where it suffices to identify an adjacent item that 'disarms' the otherwise threatening wording, there are also many cases where it is only clear from the wider semantic context that there is actually no threat. When we look once more at the word *aanval* we find that it is more commonly used in non-threatening contexts, for example in a sports context (soccer, basketball, tennis, etc.) or when talking about politics (politicians 'attacking' each other in a polical debate). Negative skip bigrams in which we include domain-specific words (for example, in the case of *aanval* words from the sports context like *doelpunt* ('goal'), *middenveld* ('centre field'), *rechterflank* ('right wing'), *wedstrijd* ('match'), *bal* ('ball'), *beker* ('cup'), and *finale* ('final')) cancel out positive matches in non-threatening contexts and contribute to reducing the proportion of false accepts.

Virtually all negative skip trigrams are directed at canceling out positive matches that are the result of skip bigrams applying across clause boundaries. For example, the

[6] The expression of future time in Dutch requires the use of an auxiliary such as *gaan* ('go') or *zullen* ('shall') with the infinitive form of the verb.

[7] The proportion of unigrams is still fairly substantial. This is due to the fact that they also include some proclitic forms (such as *kschiet* ('I shoot') and *ksteek* ('I stab')), and contracted forms such as *ikwurg* ('I strangle') and *iksla* ('I hit') where there is no space between the word tokens where there normally would be.

[8] All of these are compounds which normally in Dutch are written as single words. However, in tweets we find that they are frequently written as separate words.

skip bigram <maak af> (from the separable verb *afmaken* ('to finish off')) finds a match in the tweet

> *maak jij nog 3 screenshots met 3 zinnen er onder? moet maandag af x*
> [Eng: will you make 3 screenshots with 3 sentences below them? must be ready by Monday x]

where the tokens *maak* and *af* occur in different clauses and therefore are completely unrelated items. The negative skip trigram <maak ? af> identifies the match as a false accept and cancels it. We included the following tokens as clause boundary markers: *. , : ; ? ! en of* (punctuation, 'and' and 'or').

3.3 Spelling Variation

As there is a great deal of spelling variation in tweets, we can expect to miss out on many threatening tweets if we employ the n-grams in their base form, i.e. using essentially conventional spelling. We therefore automatically expanded the set of n-grams by including possible spelling variants of the word tokens.[9] To this end we used data from previous work on spelling variation [3], where spelling variants were clustered and represented by means of a normal form. The spelling suggestions were manually checked and where necessary removed.[10] Where on the basis of the development set we were aware of variants that did not occur among the suggestions, such variants were added. This was the case for some word tokens that are typical of Dutch street language (e.g. *deade* for Dutch *dood* ('dead') and *joeke* for *djoeke*, i.e. Dutch *doden* ('to kill')). After expansion the n-gram set comprised some 11.3 million n-grams (see also Section 6.3).

3.4 Limitations of the Present n-Grams

With the present n-grams there are clearly limitations to what can be expressed and the amount of control one may have over a pattern:

- The n-grams are (on occasion too) limited in size: max n=3;
- The length of the skip cannot be defined;
- Negative n-grams are applied independently of the positive n-gram they have been designed to cancel out;
- As the base n-grams are expanded, spelling variants are introduced for individual word tokens in isolation, i.e. not in the context of the n-gram.

[9] We refrained from expanding the negative bigrams.

[10] Items that were removed include items that had inadvertently been associated with a particular cluster (as for example *bloedband* ('blood tie'), one of the suggested variants for *bloedbad* ('blood-bath')), but also items that were at odds with what the pattern is attempting to match such as third person verb forms where the pattern is directed at first person: in Dutch the morpheme *–t* marks the third person singular form (cf. *snijdt* (3rd person singular of *snijden* ('to cut')) vs *snijd* (1st person singular)); while we do want to include *snij* as variant for *snijd*, we want to exclude *snijdt*.

4 Machine Learning of Recognition Patterns

The second approach we test for recognizing threatening tweets is machine learning. Now, a machine learning system rather than a human expert attempts to identify those n-grams that are indicative of threats. Because of computational complexity, it cannot make use of skip trigrams, but unigrams, bigrams, trigrams en skip bigrams are all available. As training material, the machine learner has access to the development sets also used in manually constructing patterns (TDS and GDS). In order to maintain optimal comparability with the first approach, we will set the acceptance threshold for the machine learning system in such a way that, on the GDS, it will accept the same amount of the tweets, about 0.8%.

4.1 Machine Learning System

Our machine learning system will have to decide whether or not a tweet is threatening or not, purely on the basis of the text in the tweet. This task is very similar to other text classification tasks, but differs in the amount of text that is available. We have decided to base our system on the Linguistic Profiling (LP) system [5]. However, it is necessary to change this system because of the shortness of tweets. Where LP bases its judgements on both overuse and underuse of n-grams, underuse cannot be used here. In the on average ten words present in tweets, practically all n-grams will be underused. Overuse will also have to be treated differently. In a text of about a thousand words, an n-gram may be overused more or less, but in a tweet one can only sensibly use presence or absence and LP's weighting based on the frequency in the test text should therefore not be used. On the other hand, the degree of overuse in the training material can still be used fruitfully.

Therefore, we use the following procedure. During training we determine which n-grams occur more frequently in the set of tweets known to be threatening (TDS) than in a background corpus of tweets (GDS), and to which degree. To determine this degree we split the TDS and GDS into blocks of 100 tweets (comparable to the texts of about one thousand words that LP has been used for in other tasks). On the GDS, we calculate the means and the standard deviations for the frequencies per block of the various n-grams. Then, on the TDS, we calculate for each block how many standard deviations the occurring n-grams are overused. The average of this value over the blocks is taken to be the degree of overuse. During testing, every presence of an overused n-gram yields a contribution to the recognition score equal to the degree of overuse, raised to the power determined by a hyperparameter P_O. The hyperparameter is set automatically during the training process.

However, when we simply add the scores for all n-grams, longer tweets can be expected to get higher scores than shorter tweets. We need to introduce some kind of correction for the text length. We have chosen to divide the score by the number of tokens in the tweet, raised to the power determined by a second hyperparameter P_L, again set during training. Finally, the corrected score is compared to a threshold to determine acceptance.

4.2 The Training Process

During training, the system learns the degree of overuse of all n-grams and the optimal settings of the two hyperparameters and the threshold. To find the optimal settings, we go through a full training-test sequence, applying ten-fold cross-validation on the TDS, as it is rather small. In this process, we try various settings for the hyperparameters, using a rough grid in a first cycle and a finer grid in a second cycle. The best values found after the second cycle are used when the system is actually applied. We determine the best values by measuring how many tweets from the background corpus are accepted when the threshold is set in such a way that the false reject rate on the TDS is kept under a specified percentage (here 5%) and choosing the values where this accept rate is lowest.

Rather than a single recognizer, using the full GDS as its background corpus, we built three recognizers which each filter out non-threats.[11] The first is trained using the full GDS as background corpus, the second using only those GDS tweets accepted by the first recognizer and the third using only those accepted by the second recognizer. For each of the three training processes, we allowed the system to falsely reject 5% of the full TDS. As we wanted the system to accept the same amount of tweets as the manual patterns, we needed to reduce the final number slightly, which we did by adjusting the threshold for the third recognizer. The eventual three filters will reduce the GTS from 1M to 47,684 (-95.2%), to 17,001 (-64.4%) and finally to 9,188 (-46.0%).

4.3 Types of N-Grams Playing a Role

Where, in the manual construction of patterns, n-grams are chosen on semantic grounds, the machine learner has no notion of meaning and works purely with statistics. It selects those n-grams which systematically occur more often in threatening tweets than in randomly selected tweets. On the basis of the approximately 80,000 tokens in the TDS, the machine learner selects 337,084 n-grams (7,674 unigrams, 34,080 bigrams, 51,361 trigrams and 243,969 skip bigrams).

If we examine these n-grams, we can identify a number of clear groups. First of all, there are the references to the planned violence that were also targeted in the manual construction of patterns. These include action words like *vermoorden* (‘to murder’) and *aanslag* (‘attack’), but also weapons like *bom* (‘bomb’) or *kraspen* (‘scratching pen’), and targeted body parts like *kop* (‘head’) or *strot* (‘throat’). Secondly, there are the intended targets themselves, which can be people (individual persons, groups of people, institutions/organizations) and/or their possessions, but also parts of the infrastructure, buildings, etc. For example, *jeugdzorg* (‘child welfare organization’), *politiebureau* (‘police station’), and *school* (‘school’). With individual persons particularly there is lot of name calling (e.g. *hoer* (‘whore’) and *mongool* (‘Downie’, i.e. person suffering from Down syndrome)) and frequent use of abusive forms of address. Examples of the latter frequently involve the use of adjectives like *vuile*, *vieze*, *gore* or *smerige* (all various degrees of ‘dirty’). Next we find interjections, such as *wollah* (street language ‘I swear’, ‘truly’) or *kanker* (originally ‘cancer’). Then

[11] On the development sets, this sequential set-up outscored the single recognizer by 2%.

there are words expressing that we are talking about a future event (*morgen* ('tomorrow')), possibly containing a warning (*wacht maar* ('just wait')) The next group are the pronouns one might expect to be more prevalent in threats, such as *ik* ('I'), *je* ('you'). Finally, we also see very general words which we cannot link directly to threats, such as *en* ('and') and *de* ('the'). As these even occur as unigrams, this may well just be caused by statistical coincidence.

The coincidence hypothesis is possibly confirmed by the observation that n-grams are not used in all three recognizers. For example, the unigram *de* is only used in the third one. On the other hand, the differences between recognizers sometimes also have a reason. The bigram *ik ga* ('I go', 'I will'), for instance, is active in the first two recognizers, but no longer in the third one. Apparently, the fact that something is announced appears to be handled at the start of the filtering process and is no longer significant in the third phase.

Of the 3,125 positive n-grams in the manually constructed patterns (before spelling expansion), 477 (15%) are also selected by the machine learner. Interestingly, even though the training set is not that large, a further 210 overlapping n-grams are found containing spelling variation.[12] As could be expected, most of the overlapping n-grams (641 out of 667) are active in all three recognizers.

5 Test Results: Quantitative Evaluation

We tested the two systems by applying them to the general and threat test sets (i.e. the GTS and the TTS resp.). We then examined all tweets from the GTS that were accepted by either system (15,312 tweets) and marked those which we deemed to be threats as described above (1,134 tweets).[13] The resulting data was used in the subsequent evaluation.

5.1 Overall Recall and Precision

The recall and precision scores of the various systems on the test sets are summarized in Table 2.

The manually constructed patterns recognize 84.8% (3871/4564) of the TDS, 84.7% (494/583) of the TTS and 79.9% (906/1134) of the threats we found in the GTS. The machine learner, with a threshold accepting the same amount of tweets on the GDS, recognizes 90.0% (4108/4564), 90.1% (525/583) and 55.8% (633/1134) respectively. However, for the machine learner we can vary the threshold, which leads to the recall scores shown in Figure 1. We see that, for both systems, there is hardly any difference between the recall on the TDS and TTS. Recall on the randomly selected tweets (from the GTS) is lower, though, for the machine learner scores quite a lot lower.

[12] These are not just idiosyncratic n-grams from the training data as 62 of the 210 (30%) are also found in the 1M tweets of the GTS, versus 271 of the 477 (57%).

[13] As also described above, this task is a difficult one and we have to assume that we missed some threats. Furthermore, there will of course also be threats that were not caught by the systems. As a result, the recall figures below can be taken to be (reasonably accurate) overestimates, but the precision figures will be underestimates.

Table 2. Recall and precision scores of various systems on various data sets.MP represents the manually constructed patterns. MP- = MP without spelling variation, MP+ = MP with spelling variation. ML represents the machine learner. The last two columns show (simple) combinations, in which MP is used with the spelling variation active.

	MP-	MP+	ML	ML or MP+	ML and MP+
Recall TDS	81.8%	84.8%	90.0%	95.5%	79.3%
Recall TTS	82.5%	84.7%	90.1%	95.5%	79.2%
Recall threats in GTS	75.5%	79.9%	55.8%	100.0%[14]	35.7%
Precision threats in GTS	12.2%	12.1%	6.9%	7.4%	30.1%

Table 3. Number of recognition patterns used by the various systems.MP represents the manually constructed patterns. MP- = MP without spelling variation, MP+ = MP with spelling variation. ML represents the machine learner. POS refers to positive n-grams and NEG to negative n-grams.

	MP- (POS/NEG)	MP+ (POS/NEG)	ML
# patterns in total	3125/13056	~7.09M/~4.25M	337,084
# patterns used on GTS	589/795	918/917	162,071
# patterns used for accepted tweets	578/83	876/102	83,917
# patterns used for correctly accepted tweets	268/13	357/15	20,141

If we examine the various threat sets (TDS, TTS, and threats in GTS) more closely, we observe that the tweets extracted from www.doodsbedreiging.nl form a rather biased sample. These are the threats that someone apparently found to be of particular interest, e.g. when they target well-known people or institutions such as schools. They also have a certain level of seriousness. The bulk of threats in the random sample, however, concern potentially violent disagreements between individuals, and are often likely to be bluster rather than real intent. We also have the impression that the language use in the two sets differs. The manually constructed patterns suffer somewhat from the differences between the data sets, but not very much. The machine learner, however, suffers greatly from the shift in data type. In order to reach the same kind of recall as seen on the threat sets, we would need to collect a training set at least as large as our TDS.

[14] Remember that we only checked tweets accepted by one of the two systems. There are probably more threatening tweets among the one million in the GTS.

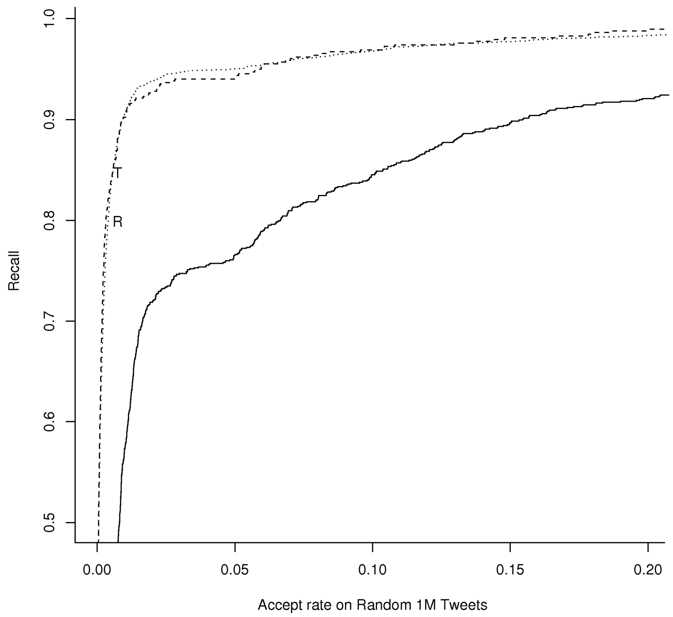

Fig. 1. The recall on the various known threat sets as a function of the accept rate on the GDS. The lines represent the machine learning recalls for the TDS (dotted), the TTS (dashed) and marked threats from the GTS (full). The markers T and R represent the manual pattern recalls for the two threat sets (TDS and TTS, represented by T) and GTS (R).

5.2 Effectiveness of Negative N-Grams

As we saw in Section 4,[15] the number of negative n-grams in the manually constructed set was far larger than the number of positive n-grams, while the machine learner could only use positive n-grams. When we look into the effectiveness of the negative n-grams we find that on the GTS they boost the precision of the manual patterns from 10.1% to 12.1% (+19%) as they prevent 1,569 tweets from being falsely accepted. There is very little loss of recall: on the TDS 30 threats are missed (-0.8%), on TTS 4 (-0.8%) and on GTS 5 (-0.5%).

5.3 Effectiveness of Modeling Spelling Variation

Modeling spelling variation increases the recall measured on all sets (Table 2). Where the gain for the threat sets, TDS and TTS with 3.7 and 2.7% respectively, is already worthwhile, the gain for the GTS is as much as 5.8%. Precision, on the other hand, is decreased much less, about 0.8%. Apart from variants where letters are repeated any number of times as for example in the various variants for *gaat dood* ('will die', which include *gaaaat dood , gaaat dooood, gaat doood*), a very frequent but more systematic type of spelling variant involves leaving out the final *–n* with infinitive

[15] See also Table 3.

forms (e.g. *aanvalle(n)* ('to attack')*, afschiete(n)* ('to shoot')*, djoeke(n)* ('to kill'), *murdere(n)* ('to murder')*, gooie(n)* ('to throw')*, neerknalle(n)* ('to shoot down')).

6 Qualitative Comparison of the Two Approaches

Apart from presenting a general evaluation, we can now also compare the two approaches that we used for our recognition task.

First of all, we can observe that both approaches are viable. The machine learner appears to score a bit better for the already available threat sets (TDS and TTS) and the human expert's patterns do better on the random selection of tweets, but both produce quite acceptable results. However, both also need a substantial amount of work, be it manual construction of patterns or manual selection of examples for the learner. We see that an often used reason for using machine learning, the reduction of labour by reusing apparently compatible data sets and annotations, is an illusion here as the recognition quality greatly degrades when we move to differently sampled data.

The two systems operate in a quite different manner as can also be deduced from Table 3. Where, for the manually constructed patterns, only a few n-grams activate and almost always lead to recognition, the machine learner uses a large amount of n-grams which each contribute a bit to the recognition. This difference leads to a relatively small overlap in recognized tweets (Table 2) and may suggest some manner of combination. However, union or intersection do not appear to be very useful, as we can see in Table 2, unless we are dealing with a task where either precision or recall is less important. And a voting technique is useless since the patterns provide only a yes/no decision (barring the rather low number of tweets where more than one pattern is present). This means that we should rather examine whether and how one approach can help improve the other.

6.1 Lessons for the Machine Learner

In order to see how the machine learner might be improved, we took the threatening tweets in the GTS which were recognized by the manually constructed patterns, but not by the machine learner, and examined which n-grams were apparently missed by the machine learner. For these 501 tweets, there were 249 different patterns active (in total 558 matches). 142 of these (403 matches) were also known to and used by the machine learner, but the threshold was not reached. 9 n-grams (25 matches) were used in some but not all the three recognizers (1 only in the first filter, 8 in the first two). In only 9 of the 25 matches, the tweet was rejected by the filter missing the pattern, but it is not clear if the presence of the n-gram would have helped. More interesting is the set of 99 n-grams (133 matches) which the machine learner missed altogether. 11 of these (12 matches) concern skip trigrams, an n-gram type which the machine learner does not use at all. The number does not appear high enough to introduce skip trigrams, given the concomitant computational cost. For 44 n-grams (46 matches), some also skip trigrams, there is some kind of non-standard spelling. This would imply that we should look into the possibility of handling spelling variation for the machine learner too. We fear that the method used here for the manually constructed patterns is far too liberal and that we should rather attempt to normalize training and test material in some way [3]. The remaining n-grams (48, of which 22 unigrams, 6 bigrams, 5 trigrams and 15 skip bigrams) have simply not been seen in the training material. They sometimes concern more rare types of violence, like *stenigen* ('to stone', but equally present are far more normal types, like *afknallen* ('to shoot down') and *vechten met* ('to fight with'). If we want to hold on to a pure machine learning approach, the solution here is to collect more training data, probably also more geared towards the type of threatening tweets that we want to find. The manually constructed patterns can of course be useful here in filtering tweets for this collection process.

6.2 Lessons for the Manual Construction of Patterns

Conversely, in order to see how the manually constructed patterns might be improved, we took the threatening tweets in the GTS which were only recognized by the machine learner. Again we listed the n-grams, this time those which were active in the machine learner's recognition. In this case, we might consider adding new patterns to our collection, copied directly from this list. However, as we have already seen, the machine learner uses very large amounts of n-grams, also ones that are innocent by themselves but correlated with threats. As a result, the 228 tweets in question yield a list of 9,630 n-grams so far unrepresented in the patterns. Most of these have no place in our patterns as they seem to have no direct bearing on threats. All in all it is doubtful whether examining the list is more fruitful than simply examining the set of additionally accepted tweets. However, we should keep in mind that this set was only constructed through a large amount of work, viz. the inspection of more than 15,000 tweets.

7 Conclusion

We have attempted to recognize threatening tweets, on the one hand using manually constructed recognition patterns and on the other hand machine learning. Both methods used token n-grams as a handle on the meaning of the tweets and both had access to the same development data and the same test data.

An evaluation on unseen data showed that both methods led to good results (85% or more recall when accepting less than 1% of the input data) when tested on unseen data that has been collected in the same way as the training data, with the machine learner having a slight edge. However, when testing on data collected in a different way, the recall of the manually constructed patterns dropped slightly, but that of the machine learner significantly.

We conclude that, for this kind of data and task, both methods require a substantial investment of labour before they can reach an acceptable level of quality, be it the construction of patterns or the collection of training material. For machine learning, there is the possibility of the shortcut of reusing existing data sets, but this shortcut proves effective only if the existing data set and annotation are very close to the target data set and task.

As for recognizing threats, we deem that both methods do provide a good start but also show room for improvement. Each method can help to some degree in improving the other, but the current precision levels are still rather low and significant amounts of manual intervention will probably be needed. We expect that progress can be made faster by investing in more information-rich methods instead of approximating meaning by way of surface features like n-grams.

References

1. The Law Dictionary. Featuring Black's Law Dictionary Free Online Legal Dictionary, 2nd edn., http://thelawdictionary.org/search2/?cx=partner-pub-4620319056007131%3A7293005414&cof=FORID%3A11&ie=UTF-&q=threat&x=6&y=6
2. Canadian Criminal Code, http://www.rcmp-grc.gc.ca/qc/pub/cybercrime/cybercrime-eng.htm
3. Tjong Kim Sang, E.: Het Gebruik van Twitter voor Taalkundig Onderzoek. TABU: Bulletin Voor Taalwetenschap 39(1/2), 62–72 (2011)
4. van Halteren, H., Oostdijk, N.: Towards Identifying Normal Forms for Various Word Form Spellings on Twitter. CLIN Journal 2, 2–22 (2012), http://www.clinjournal.org/sites/default/files/1VanHalteren2012_0.pdf
5. van Halteren, H.: Linguistic Profiling for Author Recognition and Verification. In: Scott, D., Daelemans, W., Walker, M.A. (eds.) Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, Barcelona, Spain, July 21-26. ACL, Barcelona (2004)